Computer Science and Information Systems 19(1):87–116 https://doi.org/10.2298/CSIS201229045I

Entropy-based Network Traffic Anomaly Classification Method Resilient to Deception

Juma Ibrahim and Slavko Gajin

University of Belgrade – School of Electrical Engineering,
Bul. kralja Aleksandra 73, 11000 Belgrade, Serbia
jumaibrahim04@yahoo.com
slavko.gajin@rcub.bg.ac.rs

Abstract. Entropy-based network traffic anomaly detection techniques are attractive due to their simplicity and applicability in a real-time network environment. Even though flow data provide only a basic set of information about network communications, they are suitable for efficient entropy-based anomaly detection techniques. However, a recent work reported a serious weakness of the general entropy-based anomaly detection related to its susceptibility to deception by adding spoofed data that camouflage the anomaly. Moreover, techniques for further classification of the anomalies mostly rely on machine learning, which involves additional complexity. We address these issues by providing two novel approaches. Firstly, we propose an efficient protection mechanism against entropy deception, which is based on the analysis of changes in different entropy types, namely Shannon, Rényi, and Tsallis entropies, and monitoring the number of distinct elements in a feature distribution as a new detection metric. The proposed approach makes the entropy techniques more reliable. Secondly, we have extended the existing entropy-based anomaly detection approach with the anomaly classification method. Based on a multivariate analysis of the entropy changes of multiple features as well as aggregation by complex feature combinations, entropy-based anomaly classification rules were proposed and successfully verified through experiments. Experimental results are provided to validate the feasibility of the proposed approach for practical implementation of efficient anomaly detection and classification method in the general real-life network environment.

Keywords: anomaly classification, anomaly detection, entropy, entropy deception, network behaviour analysis.

1. Introduction

The increasing complexity of modern networks is accompanied by constant changes in the security threat landscape. Signature-based intrusion detection methods are inefficient in detecting cryptographic traffic and zero-day attacks, while the intelligence put on the firewall does not protect from internal network usage. Therefore, network anomaly detection based on traffic pattern behaviour analysis is now recognized as a mandatory part of modern security analytics and protection solutions.

Several studies show that there is a significant interest in implementing entropy-based techniques for network behaviour analysis and anomaly detection 1. Their

efficiency is often demonstrated by using examples with heavily loaded anomalous traffic, such as intensive botnet or DDoS attacks. For attacks with less intensive traffic, such as SYN Flood, Port Scan or Dictionary attacks, the volumetric features do not provide sufficient information. Additional features must be used, such as the flow count and the degree of communication with other peers, the so-called behaviour features 2.

In contrast to widely presented entropy-based anomaly detection methods, significantly fewer efforts have been done on entropy-based anomaly classification. Most authors dealing with anomaly classification propose supervised machine learning techniques, even if detection is based on entropy 34. With such an approach, training with the labelled dataset is required, while the simplicity for practical implementation as one of the main benefits of entropy-based approaches, is significantly diminished.

One of the biggest weaknesses of entropy-based approaches is highlighted in 5, where the authors have shown the method to deceive flow-based detection systems by injecting additional spoofed network traffic during a DDoS attack. To the best of our knowledge, the proper solution to this problem has not been presented yet.

The motivation behind our research was to fill the above-mentioned gaps in this research problem, namely the classification of the detected anomalies which is resilient to entropy deception. The research method was based on conducting a detailed behaviour analysis of various types of anomalies caused by security attacks and investigating how they affect the entropy of the observed features, using various entropy types. The main research goal is to propose the anomaly classification method as an extension to the existing entropy detection systems, which is improved with the protection mechanism against entropy deception.

An important objective for the proposed solution is the feasibility for practical implementation in the general network environment. For this reason, only basic flow features have been chosen because they can be easily collected from network routers using NetFlow protocol 6 or similar industrial standards. The aggregation process is based on combinations of the basic flow attributes and additional so-called behaviour features which are calculated using the aggregation of the second degree. Accordingly, the presented research does not focus on specific attacks and particular use cases forcing the efficiency as high as possible by fine-tuning the parameters, but on providing a robust entropy-based method for both anomaly detection and classification that can be easily implemented in any type of the real-life network traffic.

The rest of the paper is organized as follows: the second section discusses the most relevant scientific publications related to this research. The third section outlines the proposed methodology, while section four presents and discusses the experimental results. Finally, the paper is concluded by summarizing the main contributions and results, and by defining directions for further research.

2. Related Work

Due to the relative simplicity and application in real networks, entropy-based anomaly detection still attracts great interest in the research community 789, along with more complex methods such as classification, clustering, deep learning or statistical-based approaches 10. It often relies on the flow feature distributions, based on data taken from

offline datasets for research purposes, or on data collected from real networks in practical implementation 1112.

A classical approach leverages the well-known Shannon entropy in the context of information theory 13. Feature selection and aggregation are used to generate distributions of all distinct elements and their aggregated metrics 14. A straightforward approach for DDoS attack detection is based on the volumetric feature, either using total byte and packet counts 15161718 1920 or using additionally derived features, such as average packets and bytes per flow 2122. However, volume-based metrics are insufficient for sophisticated attacks and less intensive anomalies.

Lakhina et al. in 23 used entropy measurements to analyse the real traffic aggregated inside the research networks Internet2 in the US and Geant in Europe. Using additionally injected synthetic flows, they found significant advantages of using entropy-based features over the traditional volume-based approach. The authors in 24 extended Lakhina's work using unidirectional flows and host-level granularity, modelling the behaviour for outgoing and incoming traffic.

The authors in 2 further contributed to better understand anomalous behaviour in a real network. They suggested the utilization of bidirectional data flows to avoid the biases arising from unidirectional flow analysis. Then, they analyzed the entropy of volumetric data, flow count, packet size distribution and host in/out-degree of communications with other hosts and reported a strong correlation of address and port features, emphasizing better detection abilities of behaviour features.

In 3, the authors proposed the utilization of parametrized Tsallis entropy 25 to capture separately the regions with high and low activity in the feature distribution. They modelled 20 anomaly types and injecting artificial flows into real background traffic they trained a support vector machine (SVM) to classify the anomalies.

In 26, entropy was used for profiling per-host behaviour in Internet traffic. Each of the source and destination IP addresses and ports was aggregated and the entropies of the three remaining features gave a three-dimensional entropy space with a total of 27 behaviour clusters. It was shown that different anomalies fit into particular clusters with high accuracy.

Bereziński et al. analysed realistic, synthetically generated botnet traffic injected into real flow data 4. They concluded that the parametrised Tsallis and Rényi entropy 27 provide better entropy change detection, depending on the applied parameter. They also confirmed the poor performance of volume-based approaches.

Giotis et al. in 28 presented an effective and scalable anomaly detection mechanism based on OpenFlow and sFlow data sources. The proposed architecture is modular and can accept any detection methods, such as statistical, data mining or machine learning anomaly detection. They validated the concept by adopting an entropy-based approach, using basic attributes from flow tuples, namely source and destination IP addresses and port numbers. Based on entropy changes of these four features, the proposed mechanism can identify the three most prominent network attacks, namely DDoS, port scan, and worm propagation. Using this classification method and taking advantages of SDN environments, they further contributed with an attack mitigation mechanism that identifies the attackers and protects the victim by blocking the attack.

The usability of entropy-based techniques was put under question when the authors in 5 demonstrated a method to deceive entropy-based detection by injecting additional traffic that camouflage the entropy change caused by the attack. The method exploits the simplicity of the entropy approach that transforms the whole data distribution into a

single metric. This work reveals the weakness of entropy-based techniques but has not been addressed well in the scientific literature so far.

Our previous work was focused on anomaly detection problem based on the entropy of flow features. In 29 we proposed architecture of network traffic anomaly detection system feasible for practical implementation, which includes data pre-processing, root-cause analysis, machine learning decision process, and control mechanisms for correcting, fine-tuning, and training the system. In 30 we investigated possibilities to improve the anomaly detection process by finding the outliers in time series data points using unsupervised machine learning techniques.

In this paper, we contribute to the above-mentioned research problems by providing a protection mechanism against deceiving the existing entropy-based anomaly detection techniques, further extended with a comprehensive network traffic anomaly classification method, which are the main novelties in our research.

Similar to most of the related previous work we also use bidirectional flows which are shown that provide more reliable information for the anomaly detection process, such as recognition of asymmetric traffic with no responses. However, our research is primarily based on the flow-count and behaviour features only, since the volumetric features have higher variation and generate more false positive alarms. We have analysed the characteristics of Shannon, Tsallis, and Rényi entropy types, pointing out their advantages and drawbacks. A developed method can accept any entropy types, but it is demonstrated and validated using Shannon entropy.

3. Proposed method

This section presents the problem analysis and the most relevant findings. The feature selection process is formalized and generalized defining the aggregation key features and calculated behaviour features and the feature annotation is proposed accordingly. All the entropy types are analysed in terms of changes in data distributions and their ability to detect anomalous behaviour. This analysis leads to our main contributions - the method for the protection against entropy deception and detection technique improved with the anomaly classification rules using a multivariate analysis of entropy results, which is based on the patterns in the way the features are affected by different anomalies in network communications. In contrast to similar works in this research area which are mostly based on supervised machine learning techniques, our approach is especially suitable for practical implementation in real-life network environments.

3.1. Flow feature selection

Original raw flow records, the so-called flows, are unidirectional, carrying the total packet and byte counts in the direction from the source to the destination. Combining two unidirectional flows from both directions into a single bidirectional flow offers more information about the communication pattern, and this is confirmed to be more useful in anomaly detection 24.

In the client-server communication model, which is considered in an ordinary network operation, the client initiates communication as a source in the bidirectional

flow, choosing a random source port to access the server on a fixed destination IP address and port number. The source and destination IP addresses and port numbers, as well as the protocol type, identify the flow, representing identification features, also known as a flow tuple. The packet and byte numbers in each direction are used as a metric for volume, representing volumetric features. In this paper, we use short labelling for the source and destination IP address with capital letters *S* and *D*, the source and destination ports with lowercase letters *s* and *d*, and the protocol with the letter *P*. The source and destination packet and byte counts are labelled as *sP*, *dP*, *sB* and *sB* respectively. More formally, we can introduce a set of identification features *I* and a set of volumetric features *V*:

$$I = \{S, D, P, s, d\} \tag{1}$$

$$V = \{sP, dP, sB, dB\} \tag{2}$$

Entropy calculation is based on data aggregation, which is the process of grouping flows based on the value of one or more flow features during a certain period, called epoch. For each distinct aggregated element, the so-called aggregation key, all related flows are counted into a flow number, labelled as *f*, while the volumetric features are summarized into total packets and bytes for both directions separately.

The flow identification features are the most meaningful to be used as the aggregation key. Having in mind that the protocol feature takes just a few distinct values, mostly TCP, UDP and ICMP, aggregation by this feature would not provide useful information. A set of aggregation features is therefore defined as follows:

$$\Phi = \{S, D, s, d\} \tag{3}$$

It should be noted that the aggregation can be done using more than one feature, altogether creating a complex aggregation key, or more formally, using features from any set of the power set of Φ, except an empty set. To annotate the complex aggregation key, we will use feature labels in the following order: *S*, *D*, *s*, and *d*, separated by the character '.'. Therefore, a total of 15 aggregation keys are available:

$$A = \{S, D, s, d, S.D, S.s, S.d, D.s, D.d, s.d, S.D.s, S.D.d, S.s.d, D.s.d, S.D.s.d\} \tag{4}$$

The straightforward aggregation will result in the distribution of flow count and the sum of source/destination packets/bytes of each distinct aggregated element. We will label these distributions using the feature label followed by the aggregation key in squared brackets. For instance, the distribution of the flow count feature (f), aggregated by all distinct pairs of source IP addresses (S) and destination ports (d) is labelled as f [S.d]. The flow count feature is a useful metric not only because the attacks generate a lot of malicious flows but it also influences normal traffic and increases its flow numbers due to exhaustion of the internal memory (flow cache) of the flow probes 31.

At this point, we will generalize the concept of the in-degree and out-degree features, used in 2632, which is defined by a total number of distinct source hosts per each destination host and a total number of distinct destination hosts per each source host, labelled as S[D] and D[S], respectively. Taking into consideration any other identifying features that are not used in the aggregation key, such as source and destination ports, we can additionally count the distinct occurrence of these features per aggregated element. Since they represent the communication behaviour of the main aggregated elements, we will call these additional features *behaviour* features. More formally, for

the set of aggregation features Φ and the set of aggregation keys K, a set of available behaviour features is:

$$B = \Phi \setminus \{K\} \tag{5}$$

Robust anomaly detection with a novel classification method proposed in this paper heavily utilizes behaviour features as the main source for network behaviour analysis along with the flow count feature. To briefly illustrate the usability of this approach, let us consider a DDoS amplification attack, where many source IP addresses send packets to a single destination host, all using the same source port, such as the UDP port number 53 used by DNS amplification attacks 33. This unusual network behaviour can be detected by counting the number of distinct source IP addresses (S) for each element aggregated by the destination IP address and the source port (D.s), labelled as S[D.s]. The same stands for a port scanning scenario, which can be captured by counting the occurrence of distinct destination port (d) in aggregation with the source and the destination IP address (S.D), i.e. d[S.D].

3.2. Entropy calculation

In anomaly detection techniques entropy is used to present the level of randomness in a data distribution. The changes in a data structure in a distribution obtained from the aggregation process will change the entropy value. If the entropy change is significant, it is considered as unusual behaviour in network communication or an anomaly, which often indicates security threats.

In many researchers the well-known Shannon entropy 13 is used, which is defined by the following equation:

$$H_S(X) = \sum_{i=1}^{N} p(x_i)\ \log_b \frac{1}{p(x_i)} \tag{6}$$

In the general case, N is a total number of elements in the distribution of feature values, while $p(x_i)$ is an empirical probability, calculated by the relative contribution of element x_i with value m_i in the total sum of all values, M:

$$p(x_i) = \frac{m_i}{M},\ M = \sum_{i=1}^{N} m_i \tag{7}$$

Rényi 27 and Tsallis 25 entropies involves an additional parameter α, where positive values put more weight on the highest values in the distribution (*peak*), while negative values favourite elements with low values in the distribution (*tail*):

$$H_R(X) = \frac{1}{1-\alpha}\log_b\left(\sum_{i=1}^{N} p(x_i)^{\alpha}\right) \tag{8}$$

$$H_T(X) = \frac{1}{1-\alpha}\left(\sum_{i=1}^{N} p(x_i)^{\alpha} - 1\right) \tag{9}$$

In this paper, we use a scaling factor to normalize the entropy to a value of 1 for fully randomized distribution. The scaling factor for Shannon and Rényi entropy is $1/\log_b N$ and for Tsallis entropy it is $(1-\alpha)/(N^{1-\alpha}-1)$. With such a scaling, the Shannon entropy always provides values between 0 and 1, as well as Rényi and Tsallis entropies

with positive parameter α, while the negative parameter α results in entropy values above 1.

3.3. Entropy changes detection

Over time, the aggregation and entropy calculation process generates many time series of entropy values for each feature. With normal network traffic, the entropy values are stable with minor deviations, while in the presence of an anomaly, some features are dramatically affected with significant entropy change (drop or increase). To detect these changes in the time series entropy values, a margin of accepted entropy deviation needs to be calculated first. A commonly used approach is based on the Exponential Moving Average (EMA) technique for short trend prediction 34 or taking maximum and minimum values from the sliding time window of some recent epochs. Both techniques can be used, but the rest of the presented research is based on the EMA prediction technique since it can be fined tuned to adapt more accurately and provides a baselining useful for data visualization and analysis.

With this approach, a predicted value in epoch *n*, denoted as $\hat{H}_n$, is calculated recursively, taking into account the previously predicted value $\hat{H}_{n-1}$ and the newly calculated entropy value H_{n-1} in epoch *n-1*:

$$\hat{H}_n = (1- \alpha_h)\, \hat{H}_{n-1} + \alpha_h H_{n-1} \tag{10}$$

The coefficient α_h represents the degree of weighting decrease, the so-called *smoothening factor*, which falls in the range between 0 and 1. A lower value for α_h gives a stronger influence of the previously predicted value $\hat{H}_{n-1}$, resulting in smoother baselining values, while at higher values for α_h the predicted values faster adopt and follow recent data H_{n-1} in the observed data sequence.

Some entropy time series can regularly vary their values more than the others. To identify significant entropy changes, we propose to analyse these relative variations in the context of the baselined standard deviation (*S*), using the same EMA approach as follows:

$$\hat{S}_n = (1- \alpha_s)\, \hat{S}_{n-1} + \alpha_s S_{n-1} \tag{11}$$

Finally, the range of acceptable entropy values considered as normal is defined by lower and upper thresholds, as measures of acceptable deviation from the baselined entropy value $\hat{H}_n$ as follows:

$$T_n = [\underline{T}_n , \overline{T}_n] \tag{12}$$

where

$$\underline{T}_n = \hat{H}_n - k_t \hat{S}_n \tag{13}$$

$$\overline{T}_n = \hat{H}_n + k_t \hat{S}_n \tag{14}$$

and k_t is the multiplication factor that makes the range wider, the so-called *threshold factor*. For any entropy value H_n that falls out of the threshold range T_n in epoch *n*, an alarm is triggered as an indication of an anomaly.

With proper tuning of parameters α_h, α_s and k_t, the above-mentioned technique efficiently detects significant changes in the observed time series values. We have empirically concluded that the optimal baselining trend is achieved by the following smoothing coefficient values of α_h=0.1 and α_s=0.05, while the threshold factor was set to k_t=4, which accurately captured the anomalies, while still eliminating most of the false positive alarms.

Some authors claim that parametrised Tsallis and Rényi entropy outperform the Shannon entropy in terms of the better detection of peaks or tails in the feature distributions 34. We believe that their conclusions are tightly related to the applied detection methods, data and features used in the experiments, and accordingly, this conclusion cannot be simply generalised. For that reason, in this paper, we analyse and compare the Shannon, Rényi, and Tsallis entropies from two main aspects: the ability to detect anomalies and sensitivity to deception. For Rényi and Tsallis entropies, we will use a fixed value of the parameter α (+2 and -2), which is shown to provide optimal performances 4.

To better understand the behaviour of each entropy type, we will consider a reciprocal distribution of 100 elements, given by the function 1/x, where the distribution starts with values 100, 50, 33, 25, and ends with *a long tail* of value 1. According to our experiments, this distribution roughly approximates a deviation of flow feature values in real network traffic, which is also reported in 3. Gradually increasing the peak of the distribution, from the value of 100 to 1000, the entropy is changed in the way presented in Fig. 1. The Shannon entropy, as well as parametrised entropies with the positive parameter α, results in decreased values, while the negative parameter α leads to an entropy increase. On the other hand, increasing the tail of the distribution up to 1000 new elements with value 1 involves more similarities in the data, and consequently, the entropies approach to value 1, which is shown in Fig. **2**. In all cases, the Rényi entropy with positive parameter gives the lowest entropy values, while Tsallis entropy gives much higher values (in a range from 1.7 to 106), which are not shown since they are out of the scale used in the chart. It is worth highlighting that the entropy with lower values leaves less space to detect a drop, especially when the standard deviation is higher. This is the case with the Rényi entropy with a positive parameter, which is more sensitive to the regular variation of data (highest slope in Fig. 1) and also provides the lowest values.

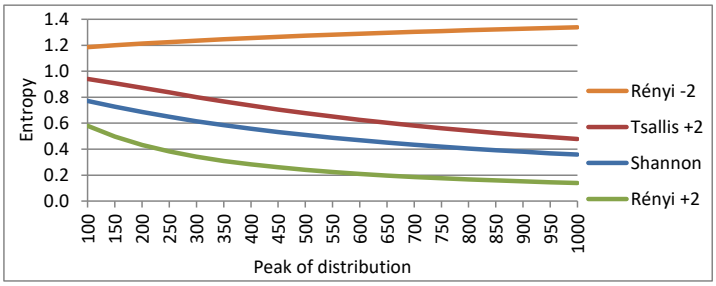

Fig. 1. The entropy change given by the increase of the distribution peak.

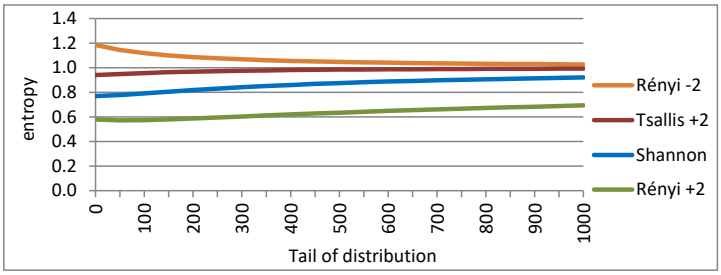

Fig. 2. The entropy change given by the increase of the distribution tail.

It should be highlighted that features with smaller standard deviation generally provide more distinguished changes, which gives better detection ability. Also, more randomized distribution and the entropy values near 1 generally leave more space for entropy drop and its detection. From the figure presented above, it should be concluded that the Rényi +2 entropy type gives the lowest values, and in case of higher standard deviation, there will be not enough space to detect changes.

3.4. Protection against entropy deception

In entropy-based approaches, anomalies are usually detected by features that generate a peak in the data distribution. This peak will make the entropy drop or increase with regards to entropy type and parameter α. Anyhow, the authors in 5 have shown that every entropy change caused by a peak in a distribution can be suppressed by adding more elements of the average value in the distribution to make data more even. The same effect can be also achieved using a value equal to 1 for each added element, but much more elements are needed in this case. With this method, attackers can camouflage the attack by generating spoofed traffic in parallel to the attack, and effectively deceive the entropy-based detection systems.

To provide a protection mechanism to this entropy deception, we analyzed the effect of entropy suppression on different entropy types, as well as on different features. The previously mentioned reciprocal distribution with peak values of 200, 500, and 1000, gives the average data values equal to 5, 9, and 13, respectively. The number of elements needed to suppress these peaks using these average values according to 5, as well as reference value equal to 1 for each entropy type, is given in Table 1. The Rényi entropy with positive parameter α (‘Rényi +2’) and Tsallis entropy with negative parameter α (‘Tsallis -2’) require the highest number of injected elements using the average value. However, this number is much higher for ‘Rényi +2’ entropy when adding elements at the end of distribution using a value equal to 1.

Table 1. The number of elements needed to deceive entropy.

Entropy type	Peak / average					
	200/5	200/1	500/9	500/1	1000/13	1000/1
Shannon	34	275	97	935	143	2135
Tsallis +2	53	280	130	1185	206	2750
Rényi +2	82	1135	207	5275	365	15200
Tsallis -2	38	45	265	166	694	348
Rényi -2	24	28	125	98	273	195

The results from Table 1 lead to the conclusion that the deception of one entropy type does not necessarily mean that the other entropy types are deceived too. This expectation is confirmed in Table 2 which shows the ratio of entropies before and after a deception in our base reciprocal distribution with a peak of 1000 and adding elements with average values 13. When nulling one entropy type (in rows), the other entropies (in columns) are below or above the initial values.

Table 2. Relative differences in deceiving different entropy types.

Entropy type	Peak=1000, average=13				
	Shannon	Tsallis +2	Rényi +2	Tsallis -2	Rényi -2
Shannon	0%	-4%	-27%	220%	3%
Tsallis +2	7%	0%	-17%	153%	2%
Rényi +2	16%	4%	0%	67%	-2%
Tsallis -2	22%	5%	18%	0%	-5%
Rényi -2	12%	2%	-8%	108%	0%

The entropy deception method proposed in 5 addresses only one feature distribution, while other features are not considered. Like the analysis of different entropy types, we can generally expect that different features are differently affected by spoofed traffic. This disbalance especially holds when injecting new elements in a behaviour feature distribution using average value, since the spoofed flows with aggregation attributes must be repeated using distinct values of behaviour feature. The easiest approach is to use full randomization of all attributes in the spoofed traffic, which would produce the elements with a value of 1 at the end of the feature distributions. However, this would significantly increase the number of distinct elements in a feature distribution, which is the case with the "Rényi +2" entropy in Table 1 with 15.200 new elements. It is also noteworthy that it is relatively easy to generate spoofed traffic to the targeted victim network, but this traffic will be highly asymmetric, mostly with no reply in opposite direction.

According to the previous analysis, we propose a protection method against entropy deception attempts, which relies on the detection of spoofed injected traffic that camouflage the attacks, based on the following principles:

- Prefer the entropy type which requires more injected elements to deceive the entropy (such as 'Rényi +2')
- Use the number of distinct elements in a feature distribution as a new detection metric, named as a *distribution length*. To the best of our knowledge, this metric has not been used in the scientific literature so far.

– Monitor the flow count of asymmetric traffic (traffic with no reply) as an indication of spoofed traffic.

The experimental results that validate the proposed protection method are presented in Section IV.

3.5. Multivariate analysis – a taxonomy of communication patterns

To identify the class of the anomaly as an indication of a particular type of security threats in addition to its detection, we propose a multivariate analysis of entropy values, which involves the observation and mutual analysis of many features. To better investigate the behaviour of different anomalies in terms of aggregation keys and the corresponding features, we have analysed the normal network behaviour and the communication characteristics of the most prominent network security attacks. Based on this analysis, we have defined flow-based taxonomy of communication patterns, which is further used for anomaly classification.

Security threats usually follow the client-server model, but the magnitude of some communication characteristics is much higher. DDoS amplification attacks utilize services such as DNS or NTP on servers that are not properly configured, the so-called *open servers* 33. The attacker sends a large number of small queries with a spoofed source IP address of the targeted host, and all servers reply to it, generating traffic of a much higher magnitude. In October and November 2016, two websites within the network of the University of Belgrade were attacked by NTP and DNS amplification attacks respectively. A single UDP source port number was used as a source of the attack (123 for NTP and 53 for DNS), but the destination port for the DNS attack was fixed to HTTP, while the NTP used a random destination port. In both cases, more than 1000 open servers generated up to 4Gbps traffic for 20 to 30 minutes, bringing down not only the attacked web servers but also disrupting other services due to the overload of the uplink of the entire national research and education network AMRES. The intensity of the attacks was easily detected and mitigated by the NetFlow Analyser tool using volumetric statistics only (bytes, packets, and flows) 35. However, to detect less intensive attacks that may remain under the radar, the communication pattern with other features must be analysed.

On the other hand, many security threats start much earlier, before real damage is caused. Network scan is looking for an open service on the network, generating flows from a single source IP address and usually an arbitrary source port toward a fixed destination port on many hosts over an enterprise network 36. Port scan is a method for determining which ports on the single host are open, producing many flows with a different destination port and a fixed destination IP address 36.

Once a host is located with the open TCP port requiring authentication, such as port 22 for SSH or 3389 for Microsoft Remote Desktop, the attacker can perform brute-force password-guessing activities, trying commonly used phrases by a dictionary attack 37. The footprint of this traffic structure is characterized by too many short flows with one or two packets transferred between two fixed IP addresses, using multiple source ports and a single destination port.

All the above-mentioned network behaviours have a very specific communication pattern marked by single or multiple sources and destination IP addresses and port numbers involved. These characteristics can be simply described using label ‘1’ for

single or ‘N’ for multiple occurrences of identification features in the order from the source IP address (S) and source port (s) to the destination IP address (D) and destination port (d), in form of ‘Ss-Dd’. In this way, previously analysed anomalies can be categorized as follows:

- DNS amplification DDoS N1-11
- NTP amplification DDoS N1-1N
- Port scan 1N-1N
- Network scan, worm propagation 1N-N1
- Dictionary attack 1N-11

This classification and labelling can be further generalized to cover all 16 permutations of labels of ‘1’ and ‘N’ for source and destination IP addresses and ports. With this generalized approach, network scan with a fixed source port is related to the communication pattern of class 11-N1, while a distributed SYN flood attack falls into the class NN-11 since the attack is performed from many source IP addresses and port numbers to a single destination IP address and TCP port number.

Regular network traffic can be also described with the introduced labelling of the communication patterns. A client can initiate many connections to a certain server, which falls into the 1N-11 class, while public servers, which are used by many clients, fall into the class NN-11. Additionally, DNS, SMTP, and HTTP proxy services follow the 1N-N1 pattern, acting as a client establishing communications with many other servers.

Along with the protection against entropy deception, another main goal in this research has been to extend the existing entropy-based anomaly detection approaches with a classification method using a multivariate analysis of different flow count and behaviour features, which is easy to implement in real-life networks. It is achieved by identifying a unique signature of the anomalous behaviour by analysing entropy changes of many observed features and developing rules for their classification. These classification rules are developed based on the analysis of the experimental results, and therefore they are analysed and presented in the next section, along with the validation of our findings.

3.6. System complexity

Calculating entropy values, with EMA prediction and standard deviations, is not a complex process once the distributions are generated by the aggregation process. However, many aggregation tasks need to process a great number of flow records, which are both CPU- and memory-intensive processes. Each flow record needs to be matched with all aggregation keys separately, counting or summarizing the corresponding values of the remaining features. Additionally, calculating behaviour features requires second-degree aggregation to count all distinct data-point occurrences.

The complexity of the algorithm highly depends on its implementation. The most efficient solution is achieved by using an unordered associative array (hash-map in Java), which in most cases has O(1) complexity in time, while the worst-case complexity is O(log n) when using balanced search trees, which are created only for a small number of entries sharing the same hash-map key. However, the complexity in the memory space is O(n) in all cases. With such implementation in Java programming

language, processing a dataset of one million flow records on a desktop computer and generating a total of 208 data distributions for all possible aggregation keys and feature combinations (including volumetric features), we have achieved a high processing speed of 30.000 flows per second consuming a total of 8 GB of RAM.

The root cause analysis requires keeping raw flow records data for at least one (previous) epoch, while a long history is always beneficial depending on the available storage space. An efficient method can be achieved by using the compressed text of binary files, while the more flexible solution for practical usage can be based on a NoSQL database, such as Elasticsearch.

Another concern relating to the real-time aggregation process and data storage is a high rate of incoming flows, such as tens of thousands per second. A solution to this is to use a flow sampling technique, processing only a statistical fraction of the flow data stream while rejecting the rest. Some information will be lost in that case, but a sufficient amount of data (up to the processing limit) is taken into account, resulting in a fairly good statistical approximation.

4. Evaluation

4.1. Datasets used

To validate the proposed approach we have chosen two labelled datasets, namely the CICIDS2017 dataset 38 and the CTU-13 dataset 39, each consisting of several flow data traces taken from real network communications and a controlled laboratory environment.

The CICIDS2017 dataset is one of the latest and most complete publicly available flow-based labelled datasets. It includes the most common attack scenarios, covering the profiles of Web-based, Brute force, DoS, DDoS, Infiltration, Heartbleed, Bot, and Scan attacks, each in a different file named according to the weekday when the dataset was created. A total of 80 flow-based features related to network communications were generated by processing real traffic with simulated attacks.

The CTU-13 dataset consists of 13 independent parts, each with internally controlled legitimate traffic, traffic generated by a real botnet network, and a large portion of the so-called background traffic, taken from the Czech Technical University network, in which minor ‘noise’ anomalies were intentionally retained. We have used the CTU-13 dataset and model communication patterns using synthetically generated anomalies to analyse their effect on each feature. Then the classification rules are validated by investigating minor ‘noise’ anomalies in the real background traffic from another trace in the same dataset.

Since our research is based on pure flow data that can be easily collected from the routers and used in real-life network environments, both datasets were slightly modified. At first, only the basic flow features were kept. Then, following the usual practice of flow configuration on network devices to avoid burst traffic load, long-lasting flows were proportionally fragmented into short equivalent flows, with the maximum duration of 60 seconds, which was set as the default epoch period.

4.2. Validation of the protection against entropy deception

Validation of the protection method against entropy deception is demonstrated on the CICIDS2017 dataset, using the trace named 'Friday afternoon'. It contains a PortScan attack when an attacker is trying to establish connections to many destination ports on a remote victim system to find vulnerabilities. In this case, a single source port is used during this process, which is described by the 11-1N communication pattern. Using the source and destination IP addresses as the aggregation key, the destination port behaviour feature d[S.D] generates a quite random distribution with the entropy value close to value 1 and a small standard deviation for all entropy types, shown in Fig. **3**. However, during the attack, in three series from epoch 112 till epoch 145, significant entropy drops are noticeable for all entropy types. The margin of acceptable deviation calculated by the EMA technique is too narrow and not shown.

We will demonstrate a deception mechanism on the second attack only, which occurs from epoch 129 till 131 with an average value in data distribution equal to 4, while the first and last attacks are left unchanged for comparison purpose. To deceive the Shannon entropy, an entropy value of 0.84 during the attack needs to be increased above the threshold value of 0.98, which requires a total of 3,000 new elements with an average value of 4. This is achieved by generating 3,000 series of 4 synthetic flows, where those 4 flows have unique source and destination IP addresses and distinct destination port number. For that reason, a total of 12,000 synthetic flows was generated and added to the dataset to deceive the Shannon entropy during this attack. Fig. 4 demonstrates that the Shannon, Rényi -2, and Tsallis +2 are successfully deceived, while Rényi +2 and Tsallis -2 are also affected, but still not sufficient to avoid detection. To camouflage the Rényi +2 entropy in this case, a total of 22,500 series of 4 synthetic flows need to be generated, which requires a total of 90,000 new spoofed flows during each of these epochs.

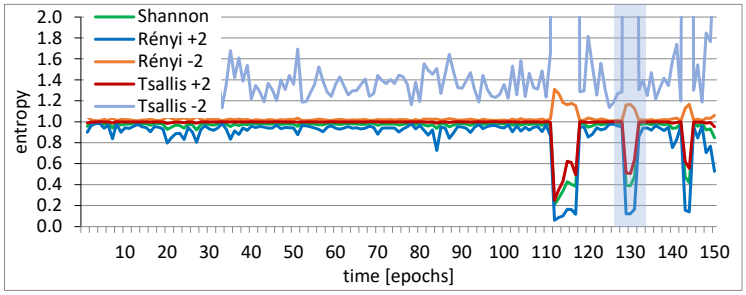

Fig. 3. The entropy of the d[S.D] feature – the original dataset.

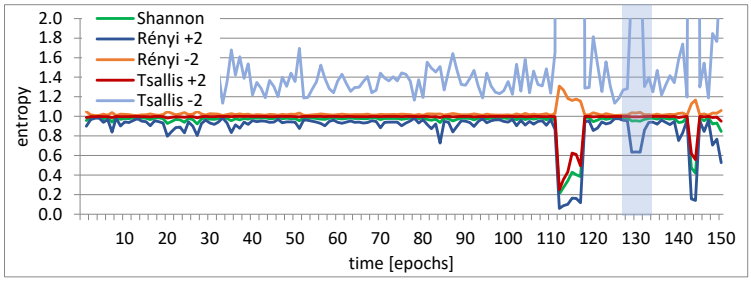

Fig. 4. The entropy of the d[S.D] feature – deceiving the Shannon entropy in epochs 129-131.

In parallel to entropy calculation, as a control mechanism to detect this deception attempt, we propose to monitor a total number of elements in feature distributions. Fig. 5 presents a total number of elements that need to be added to the d[S.D] feature distribution to deceive all entropy types. It is obvious that the injected traffic significantly exceeds the regular values and the, especially for Rényi +2 entropy type, which is far above the presented scale. The threshold can be based on a fixed value or dynamically applied using the EMA technique. It noteworthy that this metric is not affected by the other two anomalies which are not deceived.

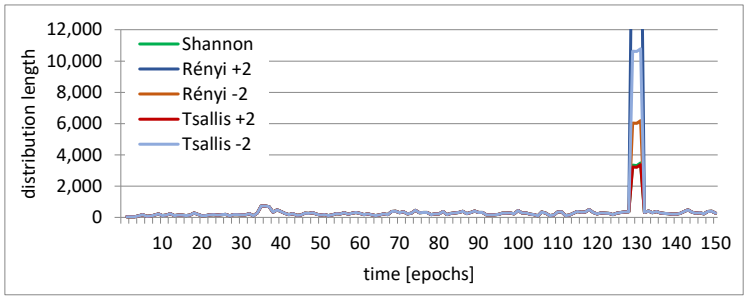

Fig. 5. The length of the d[S.D] feature distribution with spoofed traffic.

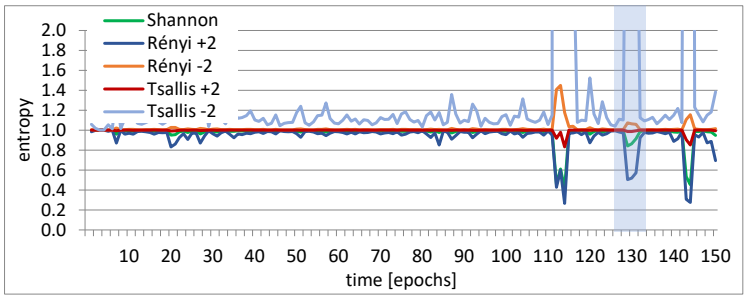

Fig. 6. The entropy of the d[S.s] feature, deceiving the feature d[S.D].

Even such a huge number of added elements for deceiving the d[S.D] feature is not enough to completely deceive the d[S.s] feature for all entropy types, which is shown in Fig. 6. To deceive the d[S.s] feature many more elements need to be injected, which is even easier to detect with the proposed metric.

4.3. Anomaly modelling

Entropy-based network traffic anomalies detection is efficient if the anomalous traffic is intensive enough to cause a significant spike in data distribution and change the entropy values for the observed feature. However, our attention is attracted by the fact that different types of anomalies leave different footprints in the entropy of the corresponding features. To better analyse this behaviour and provide a key instrument for the anomaly classification based on the proposed multivariate analysis, we have modelled characteristic anomalies for each of the 16 communication pattern classes (from 11-11 to NN-NN).

For each anomaly model, we have generated a modified dataset, combining flows of normal network traffic with the synthetically generated flows representing modelled anomalous behaviour. Normal traffic was extracted from the CTU-13 dataset, the trace named '51', with around one million flow records collected during four hours. We additionally removed smaller 'noise' anomalies and obtained a stable traffic structure with no significant deviations over time. This traffic is not used to test anomaly detection accuracy but rather as ground truth for the analysis of which features are affected by different anomalies, even those of small intensity.

Anomalies have been modelled by synthetic traffic using a flow generator software, developed by Bereziński 4 and slightly modified following our dataset format. Starting very modestly with only 25 anomalous flows per epoch, the intensity gradually increased producing a total of 50, 100, 200, 500 and 5000 flows per epoch. Small random variations were involved to present a stochastic traffic nature more realistically. It should be mentioned that the last anomaly burst was extremely huge to check whether the entropy of some features was completely immune to the anomaly. Moreover, this burst was repeated twice. The first traffic burst had 5000 purely random and mostly unique values of the aggregation feature (labelled in the model with 'N'). The second burst had the same amount of flows, but containing 10 times fewer distinct elements, each of them repeated 10 times on average. With this method, having a DDOS attack as an example described by the N1-1N model, the source port and destination IP address were fixed in the corresponding synthetic flows, while the source IP address and destination port numbers were randomized. The generated synthetic flows for each modelled anomaly class were injected separately into the dataset with the normal traffic, starting from epoch 80 in short series of three epochs, increasing the intensity every 20 epoch.

4.4. The entropy of anomaly models

The experiments were conducted for each of the 16 anomaly models separately, starting from 11-11 up to NN-NN, aggregating by all aggregation keys defined by Equation 4.

Calculating the total flow count and all behaviour features, a total of 103 feature distributions were generated for each model. As the result, a total of 1648 series of each entropy type were calculated. Volumetric features, such as the source and destination byte and packet counts, have not been used since they are efficient only for DDoS and similar volume-intensive attacks, but useless for many other attacks, such as Port Scan, Network Scan or Dictionary attack.

It is already highlighted that the entropy is changed due to a spike or *a long tail* in feature distribution. Having as an example the N1-1N model, which relates to DDoS NTP amplification attacks 35, both the destination IP address and the source port number are unique during the attack and they are good candidates for the aggregation key to capturing a spike in distribution. On the other hand, the source IP address and the destination port, which relates to the label 'N' in the N1-1N model, can be used in the aggregation key to detecting *a long tail* of the distribution. Using the Shannon entropy these two characteristic cases are demonstrated in Fig. 7 and Fig. 8 for the feature f[s] and S[d] respectively.

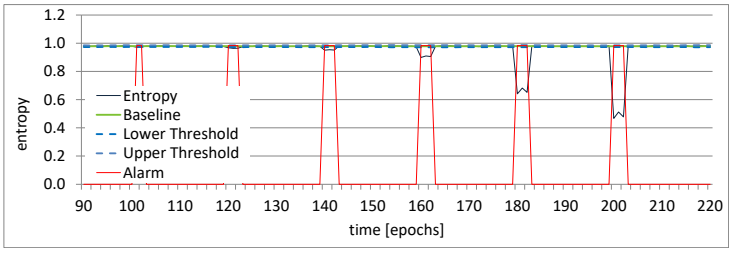

Fig. 7. The Shannon entropy and the N1-1N model - the flow count feature aggregated by the source port (f[s]).

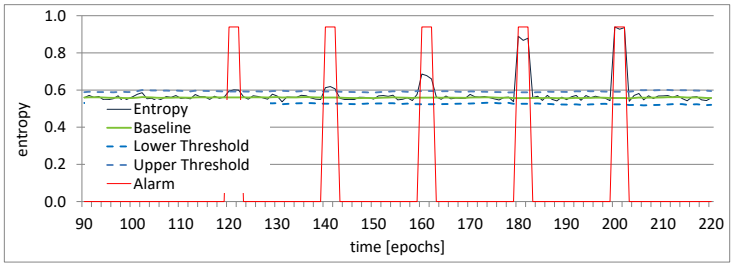

Fig. 8. The Shannon entropy and the N1-1N model - the source IP address feature aggregated by the destination port (S[d]).

Due to the natural randomness of the source port number in network communications, which is used in the aggregation key in the feature f[s], the entropy of regular traffic reaches its maximum value of 1, with a small standard deviation. With DDoS NTP amplification traffic, distinct pair of the victim IP address and the source port number used in many flows of the attack (UDP port number 123) makes a significant spike in the flow count distribution, resulting, in turn, in significantly decreased entropy. The opposite stands for the feature S[d] - the entropy values of

regular traffic are lower (around 0.55) and using a highly randomized destination port number as an aggregation key produces many elements with only one occurrence in the feature distribution.

The high sensitivity of the feature f[s] on the observed anomaly, demonstrated in Fig. 7, is used to further validate the feasibility of the protection method against entropy deception on low-rate attacks. The increase of the distribution length with the spoofed traffic to deceive all entropy types of the f[s] feature, presented in Fig. 9, can be easily detected. Only a deception of the smallest and barely noticeable anomaly around epoch 100 (with 25 synthetic flows only) can not be detected due to a high number of data elements in the distribution of the feature f[s] since the source port is highly randomized in regular network communications.

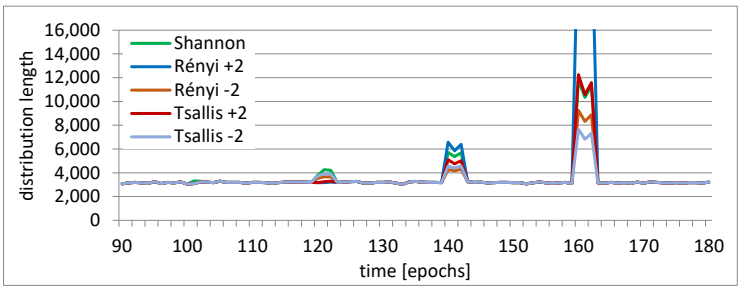

Fig. 9. Protection against entropy deception - the length of the f[s] feature distribution with spoofed traffic applied on the N1-1N model.

An extensive analysis of all entropy types has been done for all anomaly models and complete behaviour features set. Even though that some authors reported better detection ability of the Rényi and Tsallis entropy over Shannon entropy, our experiments confirmed that this is not the general rule. For the previously introduced base ground dataset with synthetically generated 7 anomaly series and the N1-1N anomaly model as an example, Table 3 presents the number of anomalies detected by different entropy types for the most characteristic features. The Shannon entropy outperforms some other entropy types for the feature S[s], there are other cases and features when other entropy types perform better.

In all cases, the difference in detection ability is related only to the smallest anomalies, while all entropy types successfully detected all anomalies of modest and especially high intensity. For that reason, we believe that the right selection of the entropy type is not a straightforward task and should consider many aspects, such as specific network traffic and its variety and deviations, a technique used for entropy change detection, as well as previously analysed resilient to deception.

Table 3. Number of detected anomalies in N11N model by different entropy types

N11N	Shannon	Renyi +2	Renyi -2	Tsallis +2	Tsallis -2
f[S]	3	2	4	3	3
S[D]	3	2	4	2	4
d[D]	6	6	6	6	6
S[s]	6	6	5	5	6
d[s]	5	4	5	4	6
S[d]	5	3	4	3	4
f[S.d]	3	2	3	3	3
S[D.s]	6	6	5	5	6

The thorough analysis of all experimental results is summarized in Table 4, which describes entropy changes for flow count and behaviour features using each aggregation key (shown in rows) and each anomaly model (shown in columns). The label 'X' in the table denotes the entropy change caused by a peak in the feature distribution, while the label 'o' denotes the entropy changes due to a tail of the distribution. The results are equal for all entropy types which confirms already demonstrated similar detection ability of all entropy types. Therefore, in the rest of the paper, we will consider the results of the Shannon only, where the labels 'X' and 'o' related to entropy drop and increase respectively. The length of the labels expresses the detection efficiency of the observed features, where more characters in the label reflect a higher efficiency, while only one character indicates a low detection ability useful only in case of extremely intensive anomalies. More precisely, one character is used when the feature is affected only by two of the most intensive synthetic anomalies in our reference dataset (the last two anomalies in Fig. 7), two characters for anomalies of moderate intensity, while the label with three characters is used for the most sensitive features able to detect even the low-rate anomalies (two left most anomalies in Fig. 7). For example, the previously mentioned features f[s] and d[S] for the N1-1N model can detect most of the generated anomalies and, therefore, they can be considered as very sensitive and are labelled with 'XXX' and 'ooo' respectively.

Even a brief look at the table reveals that the entropies of different features behave differently for different anomaly models, while some of them are not affected by a particular anomaly at all (the empty cells in the table). More importantly, how the entropies are affected by the modelled anomalies follows a very specific periodic pattern. It can be observed that the entropy drop (marked with 'X') occurs only when all identification features in the aggregation key have a single occurrence in the anomaly model (marked with '1' in the anomaly model label). In this case, entropy is always affected for the flow count feature (such as f[S.s] in the first four columns), while the behaviour features are affected only when it corresponds to mark 'N' in the anomaly model label (such as D[S.s] in 11-N1 and 11-NN models, and d[S.s] in 11-1N and 11-NN models). An increase in entropy values (marked with 'o') occurs when at least one identification feature in the aggregation key has many occurrences in the anomaly model (marked with 'N' in the anomaly model label) since this element will produce many new elements in the distribution tail. It should be noted that when the source port feature is used in the aggregation key, it can result only in an entropy drop, and not in an increase. The reason for this lies in the behaviour of the regular network, where a source host as a client initiates connections using a random source port number so that the corresponding distribution is already randomized.

Table 4. Entropy changes of the flow count and behaviour features affected by the anomaly models.

Feature		Anomaly model															
Behaviour	Flow count	11-11	11-1N	11-N1	11-NN	1N-11	1N-1N	1N-N1	1N-NN	N1-11	N1-1N	N1-N1	N1-NN	NN-11	NN-1N	NN-N1	NN-NN
D[S]				XX	XX			XX	XX	oo	oo	oo	oo	oo	oo	oo	oo
s[S]						X	X	X	X	oo	oo	oo	oo	oo	oo	oo	oo
d[S]			XXX		XXX		XXX		XXX	o	o	o	o	o	o	o	o
	f[S]	XX	XX	XX	XX	XX	XX	XX	XX	oo	oo	oo	oo	oo	oo	oo	oo
S[D]				oo	oo			oo	oo	XX	XX	oo	oo	XX	XX	oo	oo
s[D]				oo	oo	X	X	oo	oo			oo	oo	X	X	oo	oo
d[D]			XXX				XXX				XXX				XXX		
	f[D]	X	X	oo	oo	X	X	oo	oo	X	X	oo	oo	X	X	oo	oo
S[s]										XXX	XXX	XXX	XXX				
D[s]				XXX	XXX							XXX	XXX				
d[s]			XXX		XXX						XXX		XXX				
	f[s]	XXX	XXX	XXX	XXX					XXX	XXX	XXX	XXX				
S[d]			oo		oo		ooo		oo	X	ooo	X	oo	X	oo	X	oo
D[d]			ooo	XX	oo		ooo	XX	oo		ooo	XX	oo		ooo	XX	oo
s[d]			ooo		oo	X	oo	X	oo		ooo		oo	X	ooo	X	oo
	f[d]	X	ooo	X	oo	X	oo	X	oo	X	ooo	X	oo	X	oo	X	oo
s[S.D]				oo	oo	X	X	oo	oo	oo	oo	oo	oo	oo	oo	oo	oo
d[S.D]			XXX				XXX										
	f[S.D]	XX	XX	oo	oo	XX	XX	oo	oo	oo	oo	oo	oo	oo	oo	oo	oo
D[S.s]				XXX	XXX												
d[S.s]			XXX		XXX												
	f[S.s]	XXX	XXX	XXX	XXX												
D[S.d]			o	XX	o		o	XX	o	o	o	o	o	o	o	o	o
s[S.d]			oo		oo	X	oo	X	oo	oo	oo	oo	oo	oo	oo	oo	oo
	f[S.d]	XX	oo	XX	oo	XX	oo	XX	oo	oo	oo	oo	oo	oo	oo	oo	oo
S[D.s]										XXX	XXX						
d[D.s]			XXX								XXX						
	f[D.s]	XXX	XXX							XXX	XXX						
S[D.d]			oo	oo	oo		oo	oo	oo	X	oo	oo	oo	X	oo	oo	oo
s[D.d]			oo	oo	oo	X	oo	oo	oo		oo	oo	oo	X	oo	oo	oo
	f[D.d]	X	oo	oo	oo	X	oo	oo	oo	X	oo	oo	oo	X	oo	oo	oo
S[s.d]										XXX		XXX					
D[s.d]				XXX								XXX					
	f[s.d]	XXX		XXX						XXX		XXX					
d[S.D.s]			XXX														
	f[S.D.s]	XXX	XXX														
s[S.D.d]			oo	oo	o	X	o	o	oo	oo	o	oo	oo	o	oo	oo	oo
	f[S.D.d]	X	oo	oo	oo	X	oo	oo	oo	oo	oo	oo	oo	oo	oo	oo	oo
D[S.s.d]				XXX													
	f[S.s.d]	XXX		XXX													
S[D.s.d]										XXX							
	f[D.s.d]	XXX								XXX							
	f[S.D.d.s]	XXX															

The described behaviour, i.e. how the entropy of different features is affected by different anomaly models, is the key reason for the clear periodic pattern noticeable from the table. This is also the explanation of why some features are very effective in detecting some anomalies while being completely useless for others. This finding is consistent with the previous research 32628 but covers the full feature set and all the anomaly models. Moreover, it reveals that behaviour features or complex aggregation key for some anomaly models outperform commonly used flow count feature of basic

flow attributes. For instance, the authors in 28 classify a DDoS attack by detecting entropy decrease in the flow count of destination IP address and port number. This corresponds to our NN-11 anomaly model and features f[D] and f[d], which are sensitive only to the most intensive anomalies, while behaviour feature S[D] can detect less intensive anomalies (up to 10 times in our experiment). The difference in the metric performances is more obvious in the case of port scan attack, defined by the 11-1N model (using fixed source port) and the 1N-1N model (using random source port). In both cases the best performance is achieved using behaviour features d[S], d[D] and d[S.D], marked with 'XXX' label in Table 4.

This periodic pattern in the feature sensitivity to different anomalies leads to an important conclusion that each anomaly model has a unique footprint of triggered entropies, which is used in the development of the classification rules, as explained further in the text.

4.5. Classification rules

The aim of the proposed multivariate analysis is, firstly, to select the right features to ensure the most efficient detection of anomalies, and secondly, to accurately classify a detected anomaly to identify more precisely a potential security threat. Several methods for feature selection, including feature correlation, are proposed in the literature 232. In our approach, Table 4 reveals which feature is the most appropriate for which anomaly type. More importantly, a unique pattern of how the features are triggered by different anomalies can be recognised and used for defining the rules that classify an anomaly into an appropriate model.

Due to feature correlation, it is possible to minimize the set of features while keeping the ability in anomaly detection and classification. This could be done in several different ways so that the following principles are used to select the optimal rules for anomaly recognition and classification, based on the results from Table 4:

- Prefer the most efficient features ('XXX' or 'ooo').
- Prefer features affected by the minimal number of models.
- Prefer features with a simpler aggregation key.
- Use the 'Not affected' rule to differentiate feature behaviour from another model (empty cells in the table).
- Use the 'Not decrease/increase' rule to differentiate feature behaviour from another model (make a difference between 'X' and 'o' cells in the table)
- Select model identified by the smallest number of unique features first, then proceed with others.

By applying these principles to the results from Table 4, the following classification rules are jointly defined in the columns of Table 5, where all conditions must be satisfied to match the anomaly model given by the rows.

Table 5. Anomaly models identification and classification rules.

	Anomaly model	Affected (decrease)	Affected (increase)	Not affected	Not decreased
1	**11-11**	f[D.s]		S[D.s], d[D.s]	
2	**11-1N**	f[s], d[S.D]			
3	**11-N1**	D[s.d]		S[s.d]	
4	**11-NN**	d[s], D[S.s]			
5	**1N-11**	f[S.D]		d[S.D]	
6	**1N-1N**	d[S.D]		f[s]	
7	**1N-N1**	D[S.d]		D[s.d]	
8	**1N-NN**	D[S], d[S]		d[s]	
9	**N1-11**	S[D.s]		d[D.s]	
10	**N1-1N**	S[D.s], d[D.s]			
11	**N1-N1**	S[s.d], D[s.d]			
12	**N1-NN**	S[s], D[s], d[s]			
13	**NN-11**	s[D]		d[D]	d[S]
14	**NN-1N**	d[D]		S[s]	d[S]
15	**NN-N1**	S[d], D[d]		S[s]	
16	**NN-NN**		d[S], S[d]	S[s]	

For example, the 1N-N1 model, related to the horizontal port scan attack using many source port numbers, is affected by the feature D[S.d] but not by the feature D[s.d], which primarily identifies the similar 11-N1 model.

Defined classification rules include a minimal set of features, which is important for performance optimization since aggregation is a CPU and memory consuming process. However, in an anomaly detection process, it is useful to keep more features, even if they are correlated and redundant, to minimize false alarms.

4.6. Classification rules validation

The classification ability and the usefulness of the methodology presented in this paper are demonstrated on real network data taken from the dataset CTU-13. More precisely, data trace named ‘43’ was used, where intensive botnet traffic was excluded from the dataset, keeping a large portion of real-life background traffic with several anomalies of smaller intensity. Using only the flow count and behaviour features, the most characteristic results are presented below.

The entropy of the flow count feature aggregated by the source IP addresses (f[S]), shown in Fig. 10, reveals several smaller anomalies, including some minor deviations which generate false positive alarms.

The entropy of other features, such as destination port behaviour feature with the same aggregation key, namely d[S], illustrated in Fig. 11, shows that a part of the anomalies has disappeared, indicating the presence of different anomaly types in the traffic over time.

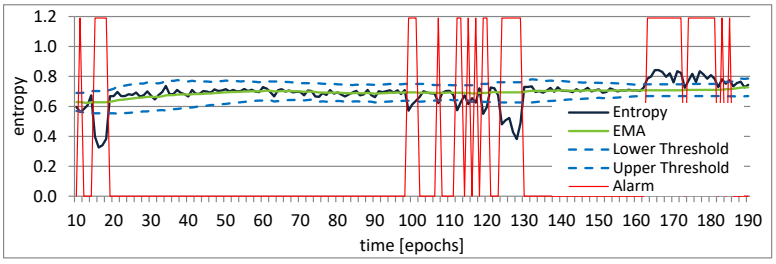

Fig. 10. The CTU-13 dataset, trace 43, regular traffic, feature f[S].

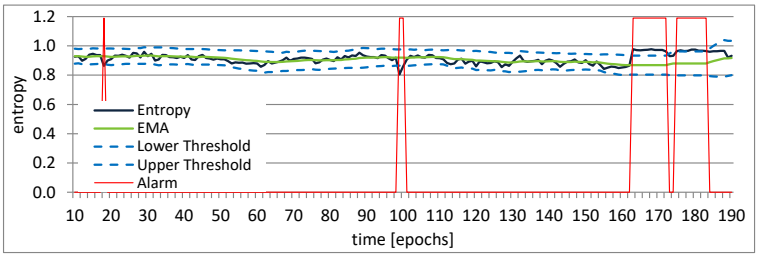

Fig. 11. The CTU-13 dataset, trace 43, regular traffic, feature d[S].

On the other hand, the entropy applied to the partition of the traffic, filtered by the protocol field, reveals new anomalies in different epochs. For TCP traffic only, the entropy of the destination port behaviour feature aggregated by the source IP addresses is shown in Fig. 12, while the entropy for the ICMP traffic using only the flow count feature aggregated by the source and the destination IP addresses are shown in Fig. 13. These less intensive anomalies were masked by the total traffic, but taking only a smaller portion of the traffic into account, the entropy changes become obvious and relevant. A small entropy deviation around epochs 100 in the total traffic had been barely noticeable in Fig. 11 and subject to suspicion as a false positive alarm until it was analysed for TCP traffic only (Fig. 12). These cases clearly demonstrate that data filtering into smaller parts is a simple method to achieve better detection sensitivity and efficiency.

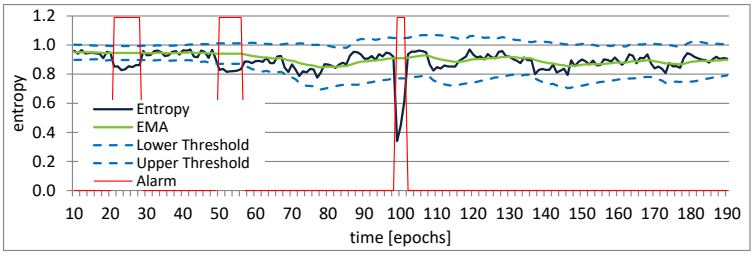

Fig. 12. The CTU-13 dataset, trace 43, TCP traffic only, feature d[S].

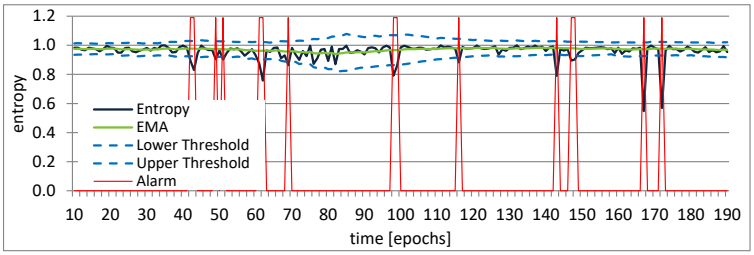

Fig. 13. The CTU-13 dataset, trace 43, ICMP traffic only, feature f[S.D].

The results of entropy analysis of the CTU-13 data trace named '43' using the proposed classification rules in different epochs for the most severe anomalies are presented in Table 6.

Table 6. Verification of anomaly classification rules using real network traffic.

Traffic	All	All	All	TCP	TCP	TCP	TCP	ICMP	ICMP
Epohs	15-19	124-130	163-180	21-29	51-57	99-102	170-171	167-168	172-173
Anomaly model	1N-11	1N-11	NN-11	1N-1N	1N-11 (1N-1N)	1N-1N	11N1	1N-11	1N-1N
D[S]							XXX		
s[S]	XXX	XX	o		XXX	X			
d[S]			o	XX	XX	XXX			XXX
f[S]	XXX	XXX	o		XXX	XX			
S[D]			XX						
s[D]	X	X	X	X	XXX	XXX			
d[D]				XXX	XXX	XXX			XXX
f[D]	XXX	X	X	X	XXX	XXX		XX	XXX
f[s]							XXX		
S[d]			XX			oo			
D[d]						oo	XX		
s[S.D]	XXX	XX			XXX	XXX			
d[S.D]				XXX	XXX	XXX			XXX
f[S.D]	XXX	XXX		X	XXX	XXX		XXX	XXX
D[S.s]							XXX		
D[S.d]							XXX		
s[S.d]	XXX	XXX			XXX				
f[S.d]	XXX	XXX			XXX		X	XXX	
S[D.d]			XX						
S[s.d]									
D[s.d]							XXX		

All detected anomalies follow the unique signature presented in Table 4 and can be properly classified by the developed rules. Only the TCP anomaly in epochs 51–57 presents a combination of two similar anomaly models: 1N-1N and 1N-11. A drill-down analysis of raw data has confirmed that the anomaly consists of flows with a larger

number of distinct destination ports according to the 1N-1N model, while one of them occurs more frequently, following the 1N-11 model. This is the case that demonstrates the feasibility of the proposed method to identify multi-vector attacks, combining two or more anomaly models. Raw data forensic is still needed for root cause analysis to mitigate the attack and proactively protect the victim.

4.7. Comparison with machine learning approaches

Entropy-based network traffic anomaly detection in many aspects completely differs from the machine learning methods, which makes it difficult and even impossible to directly compare their performances. For that reason, we rather discuss their general characteristics and leave a decision on which one is better for specific use-cases.

The main difference lays in the fact that entropy-based detection operates on the time interval level, detecting anomalies in epochs, while machine learning detection methods provide detection granularity on the data level, classifying each data point as normal or anomalous. This fundamental difference implies the following consequences:

- Anomalies detected using the entropy-based approach require further root-cause analysis to extract the information about the attackers, victims and services used.
- The entropy-based approach does not require training with a labelled dataset, as opposed to supervised machine learning, which makes it attractive for general purpose application in real-life networks with any kind of traffic unknown in advance.
- The entropy-based approach requires less processing power than most of the other techniques, which makes it attractive for real-time application.
- Performance metrics used in machine learning (Accuracy, Precision, Recall, ROC curve etc.) take into account individual labelled data and, therefore, they are inconvenient for application in the entropy-based approach.

As previously stated, the motivation behind our research has been to extend the anomaly detection technique with the classification method, for practical use in a general network environment. The entropy-based approach was chosen having in mind the above-mentioned characteristics. Anomaly detection is based on the data obtained by NetFlow or similar protocols, which are industry standards and the most convenient way to collect information about the network traffic structure. Flow data collected from network routers provide only basic information about communication peers (IP addresses, protocol, and port numbers), duration and total bytes and packets transferred. Enriched with flow count and behaviour features obtained in the aggregation process, this basic information appears to be sufficient for entropy calculation. Our experimental results confirm that this approach is efficient when the traffic structure is significantly changed during the attack, while it is useless for other attacks whose communication characteristics cannot be distinguished from regular traffic. In this work, we have solved the entropy deception problem, as a main weakness of the existing entropy detection methods.

On the other hand, machine learning approaches to network behaviour analysis rely on other communication details, such as TCP flags and window size, packets length, packet inter arrival time, jitters and their statistical parameters (average, min, max, standard deviation). Obtaining these data is based on processing raw traffic on the

packet level, which requires direct access to network traffic and demanding data processing, especially for real-time application.

The CICIDS2017 dataset was generated in this way and the authors originally used it for anomaly detection based on supervised machine learning 38. For each simulated attack, they achieved very high detection performances using various features and the following metrics: Precision (Pr - the ratio of the correctly detected attacks to all triggered alarms), Recall (Rc - the ratio of the correctly detected attacks to all attacks), and F-Measure (F1 - the harmonic mean of the Precision and Recall).

We have reproduced their experiment with the same dataset named "Thursday morning", which consists of Brute Force, Cross Site Scripting (XSS) and SQL Injection web attacks. We have also used Random Forest (RF), Multilayer Perceptron (MLP), and Naive-Bayes (NB) machine learning algorithms, and the same features used by the authors in 38, namely the initial TCP window size in both directions and the total bytes transferred from the source to destination.

In our reproduced experiments in the Weka software, using 70% of training and 30% of testing data randomly chosen from the dataset, we have generally confirmed their results, especially in terms of the Recall performance metrics.

Furthermore, we have performed a deeper investigation of raw data, which has revealed that most attack flows used the initial TCP window size of 29,200 and 28,960 bytes from the source and destination directions, respectively. Since the TCP window can take an arbitrary value even in attack communications, we wanted to check the detection capability of the machine learning algorithms when these values were changed. For this reason, we manually increased the initial TCP windows of attack flows in the testing dataset by 3%, 10% and 30% and repeated the experiments. From Table 7, which summarises the results, it is obvious that the Random Forest algorithm dramatically lost the detection capability even with small changes of 3%, while the Multilayer Perceptron algorithm was not able to detect any attack at all. Only the Naive-Bayes algorithm was more resilient to the initial TCP window value changes, but its performance was the lowest.

Table 7. Supervised machine learning performance evaluation

Alg.	Dataset	Precision	Recall	F1
RF	**Original**	**0.850**	**0.981**	**0.911**
	Modified, 3%	0.176	0.037	0.061
	Modified, 10%	0.176	0.037	0.061
	Modified, 30%	0.176	0.037	0.061
MLP	**Original**	**0.771**	**0.840**	**0.804**
	Modified, 3%	0.000	0.000	N/A
	Modified, 10%	0.000	0.000	N/A
	Modified, 30%	0.000	0.000	N/A
NB	**Original**	**0.132**	**0.909**	**0.230**
	Modified, 3%	0.132	0.908	0.230
	Modified, 10%	0.123	0.842	0.215
	Modified, 30%	0.123	0.842	0.215

The above example demonstrates that some machine learning algorithms, which are based on such specific feature values, can be easily deceived with just a small variation in the attack scenario. Rather than just presenting a pure performance measurement,

which can be misleading, we suggest further analysis of raw data and the meaning of the features in the context of the applied machine learning algorithms.

5. Conclusions

In this paper, we have presented a comprehensive method for entropy-based network traffic anomaly classification empowered by a novel protection mechanism against the deception of entropy detection capabilities. We contribute to the research topics in several directions.

Firstly, we have compared the ways how the Shannon, Tsallis and Rényi entropies respond to changes in feature distribution caused by spoofed traffic injected to deceive the entropy detection systems. We have found that a total number of elements in distribution, so-called a distribution length, is an efficient metric to detect entropy deception attempts. If deception is applied, a distribution length is much longer, exceeding the threshold, either fixed or dynamically calculated. We have shown that the Rényi entropy with positive parameter α is the most resilient to deception since it requires the largest amount of spoofed traffic. However, this entropy type provides the lowest entropy values, and for some features with higher data variation, it is not suitable to detect entropy drops.

Secondly, we have formalized and generalized the concept of aggregation and behaviour features, which better represents the network traffic structure using only basic flow attributes. Based on these features, we have modelled 16 anomaly models, associate with a wide range of security attacks. Extensive experiments were conducted for all anomaly models using full features set, calculating the Shannon, Tsallis and Rényi entropies, with both positive and negative parameter. Contrary to the widely accepted belief that the parameterized Tsallis and Rényi entropies outperform the Shannon entropy, we have shown that there is no significant difference in anomaly detection capability between these entropy types. The right choice of entropy type rather depends on the specific network traffic, its variety and deviations, used features and other parameters and characteristics, including the resilience to deception.

Thirdly, the conducted experiments confirmed that each anomaly model leaves a unique signature in the behaviour, indicating how entropies of different features are affected. Based on the multivariate analysis of different features, the original anomaly classification rules have been developed, which is another novel contribution presented in this paper. The efficiency of the anomaly classification method is validated through the presented experimental results.

Finally, but not less important, we believe that our work contributes in many respects to a better understanding of the entropy-based network behaviour analysis and anomaly detection, despite many papers in this research field. Based on the comprehensive experimental results and the conducted analysis, we have also concluded that supervised machine learning methods used for network behaviour analysis involve significant limitations for efficient practical use in real-time. Consequently, the proposed method based on the entropy of the basic flow data seems to be more feasible for practical implementation and general use. In this context, unsupervised machine learning, with no training required, could be a promising alternative solution.

Therefore, our further work will be oriented towards unsupervised machine learning, along with testing the concept and performances in various real-time network environments. This includes a classical approach using external data collection and processing system, as well as data plane programmability techniques on modern software defined networking architecture.

Acknowledgement. This work was partially supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia under the EUREKA project "Network Traffic Anomaly Detection system based on NetFlow data analysis – TRADE" (grant number E! 13304).

References

1. J. Mazel, R. Fontugne, K. Fukuda, A taxonomy of anomalies in backbone network traffic, in: Proceedings of the 2014 International Wireless Communications and Mobile Computing Conference (IWCMC), Nicosia, Cyprus, 4-8 Aug 2014: 30-36. IEEE. doi: 10.1109/IWCMC.2014.6906328.
2. G. Nychis, V. Sekar, D.G. Andersen, H. Kim, H. Zhang, An Empirical Evaluation of Entropy-based Traffic Anomaly Detection, in: Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement (IMC "08), Vouliagmeni, Greece, 20–22 October 2008: 151–156. ACM New York, NY, USA. doi: 10.1145/1452520.1452539.
3. B. Tellenbach, M. Burkhart, D. Schatzmann, D. Gugelmanna, D. Sornette, Accurate network anomaly classification with generalized entropy metrics, Computer Networks 55 (11), (2011) 3485-3502, doi: 10.1016/j.comnet.2011.07.008.
4. P. Berezinski, B. Jasiul, M. Szpyrka, An entropy-based network anomaly detection method, Entropy 17 (4): 2367–2408, (2015) doi: doi.org/10.3390/e17042367.
5. I. Özçelik, R. R. Brooks, Deceiving entropy based DoS detection, Computers & Security 48, (2015) 234-245, doi: 10.1016/j.cose.2014.10.013.
6. B. Claise, Cisco Systems NetFlow Services Export Version 9, RFC 3954.
7. B. Li, J. Springer, G. Bebis, M.H. Gunes, A survey of network flow applications, Journal of Network and Computer Applications 36 (2), (2013) 567–581. doi: 10.1016/j.jnca.2012.12.020.
8. M. Ahmed, A.N. Mahmood, J. Hu, A survey of network anomaly detection techniques, Journal of Network and Computer Applications 60, (2016) 19–31. doi: 10.1016/j.jnca.2015.11.016.
9. V. Chandola, A. Banerjee, V. Kumar, Anomaly Detection: A Survey, ACM Computing Surveys 41 (3), (2009), doi: 10.1145/1541880.1541882.
10. N. Moustafa, J. Hu, J. Slay, A holistic review of Network Anomaly Detection Systems: A comprehensive survey, Journal of Network and Computer Applications 128, (2019) 33-55. doi: 10.1016/j.jnca.2018.12.006.
11. M.F. Umer, M. Fahad, M. Sher, Y. Bi, Flow-based intrusion detection: Techniques and challenges, Computers & Security 70, (2017) 238-254. doi: 10.1016/j.cose.2017.05.009.
12. A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, B. Stiller, An Overview of IP Flow-Based Intrusion Detection, IEEE Communications Surveys & Tutorials 12 (3), (2010) 343 – 356. doi: 10.1109/SURV.2010.032210.00054.
13. C.E. Shannon, A mathematical theory of communication, Bell system technical journal, 27(3), 1948, 379-423, doi: 10.1002/j.1538-7305.1948.tb01338.x.
14. N. Moustafa, G. Creech, J. Slay, Flow Aggregator Module for Analysing Network Traffic, Progress in Computing, Analytics and Networking, Advances in Intelligent Systems and Computing, vol. 710 (2018) 19-29. Springer, Singapore. doi: 10.1007/978-981-10-7871-2_3.

15. A. Lakhina, M. Crovella, C. Diot, Diagnosing Network-Wide Traffic Anomalies, ACM SIGCOMM Computer Communication Review 34 (4), (2004) 219-230. doi: 10.1145/1030194.1015492.
16. P.D. Bojovic, I. Basicevic, S. Ocovaj, M. Popovic, A practical approach to detection of distributed denial-of-service attacks using a hybrid detection method, Computers and Electrical Engineering 73, (2018) 84-96. doi: 10.1016/j.compeleceng.2018.11.004.
17. O. Joldzic, Z. Djuric, P. Vuletic, A transparent and scalable anomaly-based DoS detection method, Computer Networks 104, (2016) 27-42. doi: 10.1016/j.comnet.2016.05.004.
18. D. Roosi, S. Valenti, Fine-grained traffic classification with netflow data, in: Proceedings of the 6th International Wireless Communications and Mobile Computing Conference, Caen, France, June 28 - July 02 2010: 479-483. ACM New York, NY, USA. doi: 10.1145/1815396.1815507.
19. P. Barford, J. Kline, D. Plonka, A. Ron, A signal analysis of network traffic anomalies, in: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement, Marseille, France, November 06 – 08 2002: 71-82. doi: 10.1145/637201.637210.
20. H.A. Nguyen, T. Van Nguyen, D.I. Kim, D. Choi, Network Traffic Anomalies Detection and Identification with Flow Monitorin, 5th IFIP International Conference on Wireless and Optical Communications Networks (WOCN '08), Surabaya, Indonesia, 5-7 May 2008. IEEE. doi: 10.1109/WOCN.2008.4542524.
21. R. Braga, E. Mota, A. Passito, Lightweight DDoS flooding attack detection using NOX/OpenFlow, IEEE 35th Conference on Local Computer Networks, Denver, USA, 10-14 Oct. 2010. IEEE: 408–415. DOI: 10.1109/LCN.2010.5735752.
22. Y. Feng, R. Guo, D. Wang, B. Zhang, Research on the Active DDoS Filtering Algorithm Based on IP Flow, 2009 Fifth International Conference on Natural Computation. Tianjin, China, 14-16 Aug. 2009: 628–632. IEEE.
23. A. Lakhina, M. Crovella, C. Diot, Mining Anomalies Using Traffic Feature Distributions, Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications 35 (4), (2005) 217-228. doi: 10.1145/1080091.1080118.
24. T. Pevný, M. Rehák, M. Grill, Identifying suspicious users in corporate networks, Proceedings of workshop on information forensics and security: 1–6, (2012).
25. C. Tsallis, Possible generalization of Boltzmann–Gibbs statistics, Journal of Statistical Physics 52 (1-2), (1988) 479–487, doi: 10.1007/BF01016429.
26. K. Xu, Z.L. Zhang, S. Bhattacharyya, Internet traffic behaviour profiling for network security monitoring, IEEE/ACM Transactions on Networking 16 (6), (2008) 1241-1252, doi: 10.1109/TNET.2007.911438.
27. A. Rényi, On measures of entropy and information, in: Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability 1, (1961) 547–561.
28. K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, V.Maglaris, Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments, Computer Networks, Vol 62, (2014), 122-136, doi: 10.1016/j.bjp.2013.10.014
29. J. Ibrahim, V. Timčenko, S. Gajin, A comprehensive flow-based anomaly detection architecture using entropy calculation and machine learning classification, Proceedings of the 9th International Conference on Information Society and Technology, ISBN 978-86-85525-24-7, (2019) 138-143
30. V. Timčenko, S. Gajin, Time-series entropy data clustering for effective anomaly detection, Proceedings of the 10th International Conference on Information Society and Technology, Information Society of Serbia, ISBN 978-86-85525-24-7, (2020) 170-175.
31. R. Sadre, A. Sperotto, A. Pras, The effects of DDoS attacks on flow monitoring applications, 2012 IEEE Network Operations and Management Symposium, (2012) 269-277, doi:10.1109/NOMS.2012.6211908.

32. L. Ertöz, E. Eilertson, A. Lazarevic, P.N. Tan, V. Kumar, J. Srivastava, P. Dokas, Chapter 3: The MINDS - Minnesota Intrusion Detection System, Next Generation Data Mining, MIT Press, Boston.
33. C. Fachkha, E. Bou-Harb, M. Debbabi, Fingerprinting Internet DNS Amplification DDoS activities, in: 6th International Conference on New Technologies, Mobility and Security (NTMS), Dubai, United Arab Emirates, 30 March-2 April 2014, 1–5, doi: 10.1109/NTMS.2014.6814019.
34. A. J. Lawrance, P.A.W. Lewis, An exponential moving-average sequence and point process (EMA1), Journal of Applied Probability 14 (1), (1977) 98-113, doi: 10.2307/3213263.
35. NetVizura, "NetVizura Netflow Analyzer, Case study – DDoS Attack by NTP Amplification.", Accessed 22 July 2020. https://www.netvizura.com/files/products/netflow/resources/doc/DDoS-Attack-by-NTP-Amplification-NetVizura.pdf.
36. M. Allman, V. Paxson, J. Terrell, A brief history of scanning, in: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, San Diego, California, USA — October 24 - 26, (2007) 77-82. doi: 10.1145/1298306.1298316.
37. R. Hofstede, L. Hendriks, A. Sperotto, A. Pras, SSH compromise detection using NetFlow/IPFIX, ACM SIGCOMM Computer Communication Review 44 (5): 20–26. ACM New York, NY, USA, (2014) doi: 10.1145/2677046.2677050.
38. I. Sharafaldin, A.H. Lashkari, A. Ghorbani, Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterizatio, in: Proceedings of the 4th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP, ISBN 978-989-758-282-0, (2018) pages 108-116. doi: 10.5220/0006639801080116.
39. S. Garcia, M. Grill, J. Stiborek, A. Zunino, An empirical comparison of botnet detection methods, Computers and Security Journal 45, (2014) 100-123. doi: 10.1016/j.cose.2014.05.011.

Juma A. Ibrahim received his BSc from the University of Tripoli, Faculty of science, Computer science department, and MSc in Computer Engineering from the University of Belgrade, School of Electrical Engineering. He worked as a professor at the College of Computer Technology, Tripoli, and CCNA instructor at the Cisco Networking Academy, Tripoli and Injella, Libya. He is currently a PhD student at the University Of Belgrade, School Of Electrical Engineering, at the Department of Computer Science and Information Technology. His research interest includes computer networks and security, with the special attention to network intrusion detection and prevention systems.

Slavko Gajin received his BSc, MSc and PhD in Computer Engineering from the University of Belgrade, School of Electrical Engineering. He is currently working as a director of Computer Centre of the University of Belgrade and an associate professor at the University of Belgrade, School of Electrical Engineering at the Department of Computer Engineering and Information Theory, teaching topics in the field of computer networks. His research interests span from network performance monitoring and management to network security, network behaviour analysis and machine learning. He worked on the development of the Serbian Research and Education Network (AMRES).

Received: December 29, 2020; Accepted: July 12, 2021.