A comprehensive flow-based anomaly detection architecture using entropy calculation and machine learning classification

Juma Ibrahim*, Valentina Timčenko**, Slavko Gajin*
*University of Belgrade - School of Electrical Engineering, Belgrade, Serbia
**Institute Mihajlo Pupin, Belgrade, Serbia
valentina.timcenko@pupin.rs, jumaibrahim04@yahoo.com, slavko.gajin@rcub.bg.ac.rs

***Abstract*—The network behavior analysis relies on the understanding of normal or acceptable behavior characteristics in the network communication, in order to efficiently detect the anomalous traffic patterns and deviations that could cause performance issues or indicate a breach, thus allowing near real-time alerting and visibility of the potential network security threats. In contrast to the signature based intrusion detection systems, this approach is extremely beneficial not only for identifying unknown threats, zero-day attacks, and suspicious behavior regardless the used cryptographic methodology, but also to identify and allow the performance optimization opportunities. We propose a comprehensive architecture for practical implementation of the flow based anomaly detection solution for real life use cases, which is based on the combination of the entropy calculation and machine learning techniques, with the ability to model the attacks and generate representative labelled training data set.**

I. INTRODUCTION

With the continuous advances in network technologies, there is a constant change of the security threat landscape. A number of different threats and attacks, combined with the suspicious network behavior, have imposed to the network and security specialists an increased pressure in the process of security threats detection and minimization of the attack impacts, all in a timely manner. With the increasing complexity of the modern networks, the use of the bare manual security analysis with no automation leaves many blind spots. On the other hand, the signature-based intrusion detection methods are inefficient in detecting cryptographic traffic and zero-day attacks, while the intelligence put on the firewall does not provide protection from internal network usage. Therefore, the network anomaly detection that is based on the traffic pattern behavior analysis, is now recognized as the mandatory part of modern security protection solutions. This subject is addressed in a large number of scientific papers, where the authors report that both entropy and machine learning approaches provide promising solutions to this problem.

In this paper we have proposed the architecture of the solution that combines entropy and volumetric based network behavior profiling, which is supported by machine learning decision making in order to accurately recognize and classify security threats and other performance issues and easily distinguish them from normal traffic.

The following section provides some related work information. The third section discusses the details of the proposed solution architecture, while in section IV we provide a brief explanation of the applied entropy calculation methodology. The V. section is dedicated to the explanation of the experimental environment and test result analysis. Finally, we conclude this paper by summarizing our main contributions and results, and defining directions of further work.

II. RELATED WORK

A number of studies show that there is a high aspiration of the full implementation of the entropy based techniques for intrusion and anomaly detection. The main stumbling stone is that these techniques alone can be efficient, but the high sensitivity to the traffic pattern changes and need to adapt parameters to these changes, make them inaccurate sometimes. Needless to say, the industrial vendors are keeping their solutions as corporate secrets. In spite of number of researches in this area, they mostly rely on just a few existing labelled data sets, which limit its application in real-life network traffic. Therefore, the practical implementation of anomaly detection is still not well explained in the literature, which leaves an open room for further research in this area.

The entropy-based methods that rely on the NetFlow feature distributions have gained a major interest [4]. The authors in [5] reported a strong correlation of address and port features, emphasizing better detection abilities of degree features. They also suggested the usage of bidirectional data flows to avoid the biases arising from unidirectional flow analysis.

A group of authors in [6] proposed the Entropy-Based Network Anomaly Detection method for detection of modern botnet-like attacks. They applied parameterized entropy-based anomaly detection with supervised machine learning approach, gaining the low false positive rate. The study is based on the use of the proprietary data set containing labelled flows, generated with the special purpose Python tool. The obtained results indicate the dominance of the parametrized Tsallis entropy [1] and Rényi entropy [2] over the commonly used Shannon entropy [3], mostly in better detection of peak or tail in the feature distributions, depending on the used parameter.

A recent research proposes a method for anomaly and attack detection that is based on the Shannon and Rényi cross entropy [8]. It analyses the distribution characteristics of the alert features in order to detect

Copyright 2019 by Information Society of Serbia - ISOS, Serbia | Creative Commons License: CC BY-NC-ND

attacks and gain low false alert rate. A group of authors proposed a solution for short time predictions and efficient reduction of high-dimensional network traffic to a single metric [9]. The described algorithm detects the abrupt changes in network entropy time series, by applying the simple exponential smoothing algorithm. In [10], a case of merging the entropy-based system and anomaly detection system for multilevel Distributed Denial of Service (DDoS) detection is proposed. In order to diagnose the anomaly, this solution efficiently calculates the degree entropy for feature distributions. Another study provides a framework for IDSs that are based on the information theory analysis [11]. Within the framework the performances of anomaly based and signature based IDSs can be unified, and provides the static/dynamic fine-tuning in order to achieve optimal IDS operation. The authors in [12] propose an entropy-based A-NIDS, providing the structured and comprehensive overview of their research.

In [13], authors have proposed anomaly detection engine that is based on k-nearest neighbor's (K-NN) and K-Means Clustering (KMC) machine learning algorithms and is applied for KDD99 dataset. This approach applies the information entropy measures while ranking the network connection features by their importance for the process of attack detection.

III. Proposed Solution

We contribute to the above mentioned research problem by proposing a comprehensive architecture for practical implementation of the flow based anomaly detection solution for real life use cases, based on the combination of entropy and machine learning techniques, with the ability to model the labelled training data set. The source of the network information usage is the NetFlow accounting data, exported from the network routers and stored to the central collecting server. The NetFlow is a network traffic accounting and exporting technology originally developed by Cisco [4], but today it represents a de-facto industry standard, and often addresses similar standardized or vendor specific protocols (IPFIX by IETF [14], Jflow by Juniper [15], NetStream by Huawei [16], Cflow by Alcatel-Lucent [17]).

The entropy calculations are used to indicate unusual traffic, while machine learning algorithms are applied with a goal to minimize false positive and false negative alarms, all in order to achieve better performances. The proposed architecture is modular, flexible and open for implementation of different entropy calculations (Shannon, Tsallis, Rényi) and machine learning (ML) algorithms, where the unsupervised ML is used for clustering the data instances and supervised ML is applied for gaining better recognition and data classification. The architecture is specially designed for practical implementation.

The main building blocks of the architecture are illustrated in Fig. 1, and described in more details in the rest of this section.

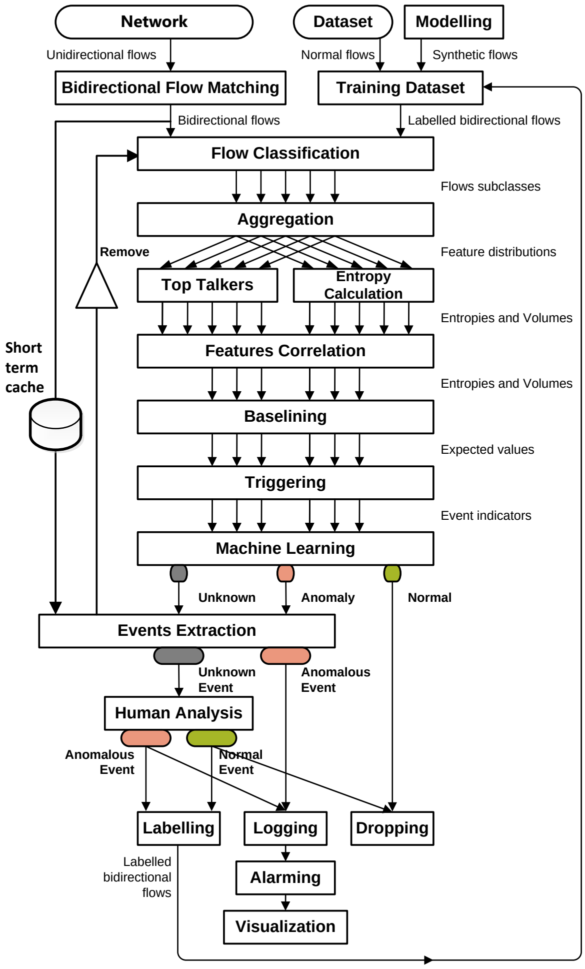

Figure 1. Flow-based anomaly detection architecture

- **Bidirectional Flow Matching** – Collected unidirectional raw NetFlow records (named *flows*) are paired in both directions, recognizing the initiators (clients) and responders (servers) in network communication, which allows achieving better accuracy [5].
- **Labelled dataset** – Previously prepared NetFlow data, with each flow labelled as normal or anomalous, provides a useful source of different network traffic behavior that is suitable for experiments and fine tuning of detection methods.
- **Modelling Training Dataset** – In order to describe different traffic behavior, some specific security threats and other network traffic anomalies can be adequately modelled. The representative synthetic data flows are generated in accordance to the designed anomaly model and traffic intensity, and inserted into normal traffic that forms the training dataset.
- **Flow Classification** – Bidirectional flows are divided into several subclasses that are characterized by different traffic profiles (DNS, email, web, Microsoft services, ICMP etc.). We have shown that this approach leads to more sensitive detection of the

low-intensive anomalies that may stay undetected if the analysis is applied on the whole traffic.

- **Aggregation** – Bidirectional and classified flow data are processed in fixed intervals, the so called epochs. Since each flow record consists of the identification data (IP addresses, port numbers and protocols) and volumetric data (number of packets and bytes), the aggregation process summarizes all volumetric data for distinct appearance of identification data during the period of one epoch. Additional data/information can be further extracted, such as the total number of the flows for each aggregated element, and the second degree features that represent the number of appearance of the other identification fields that are not used as a part of the aggregation key. The output of the aggregation module is a set of feature distributions, that contain summarized or counted values per aggregated elements, sorted in descending order. For example, the aggregation of the data instances by the source IP address in one epoch will generate distributions of the following features:
  - total flow number generated by the source IP address,
  - total number of the bytes and packets sent and received by the source IP address, and
  - total number of different destination IP addresses and ports that are in communication with the specified source IP address.
- **Top Talkers calculation** – the algorithm will summarize data volumes generated by the biggest traffic contributors (one or few of them), for each analyzed feature per epoch.
- **Entropy calculation** – The entropy is calculated over each distribution, for every feature and epoch. In contrast to the Top Talkers calculations, the entropy calculation takes into account the relative contribution of each element in the distributions, resulting into single number that presents uniqueness (differences or similarities) of the element values. Therefore, the entropy value captures the behavior of the biggest contributors as well as of the large number of minor appearances in the feature distributions. Over the time, each observed feature will be presented by entropy time series.
- **Features correlation**s – Some of the calculated features are highly correlated and can be removed from further processing with no information loss.
- **Baselining** – Baselining is performed for Top Talker and entropy time series data, thus for the reduced set of features using specific prediction model, such as Exponential Moving Average (EMA) [18]. These values are treated as an expected normal behavior of the examined feature. The standard deviation is calculated in the same manner, and its baselining is used as a measure of acceptable changes.
- **Triggering** – If the observed feature value is out of the tolerance range which is close to the expected values given by the baselining process, an alarm will be triggered. The tolerance range is determined by the expected standard deviation, multiplied by the certain factor. An alternative way for analyzing changes of the Top Talker features is to setup a threshold relative to the expected baselined values (for instance "trigger an alarm if the value exceeds 100%").
- **Machine Learning** – is a part of the Hybrid Machine Learning (HML) module. It is previously trained to recognize the vector of triggered alarms as an anomalous behavior, in which case the anomaly is classified into certain categories, such as DoS, network or port scanning, TCP SYN flood, brute force attack, etc. With this approach, the isolated alarms can be ignored, thus minimizing false positive and false negative alarms. The output of this process categorizes the network traffic behavior in the observed epoch – it defines it as normal or anomalous. If there is not enough knowledge or certainty to make a precise decision, the conclusion may stay unknown.
- **Events Extraction** – For practical implementation, both indicators of anomaly and unknown traffic patterns need to be enriched with the additional information related to the flows that caused that event. These flows are extracted from the Short-term cache unit and removed from the aggregated data in the specific epoch. The whole process is then performed again, recalculating all features. This loop repeats until all the anomalous or unknown events are extracted from the analyzed dataset.
- **Human analysis** – the previously unknown events are needed to be analyzed by the network security specialist and manually classified as the normal or anomalous. In both cases, the extracted flows of those events are labelled and used to extend the training data set.
- **Logging, Alarming, Visualization** – the detected anomalous events are logged with proper alarming and visualization for further analysis and security event management.
- **Labelling** – the unknown events, that are manually recognized by the network security specialists, are labelled and archived in the dataset for further offline training of the Hybrid Machine Learning module. This feedback loop provides the possibility for relearning and update of the detection process with new knowledge.

IV. METHODOLOGY

The regular traffic with large data transfer is not rare traffic behavior in modern network communications. The analysis of the volumetric features (bytes and packets) can detect only extremely large traffic loads of very intensive DDoS attacks. Setting up a lower threshold level for these features would produce large number of false positive alarms, which practically makes volumetric features useless for less intensive anomalies.

Our approach is therefore oriented towards extracting the hidden patterns in network behavior, which mostly affects the second degree features. While one or more identification features, namely source and destination IP address and ports, are used as keys for the aggregation, the remaining features can be used as second degree features.

Not all of the combinations of identification features are useful for the aggregation. The behavior pattern analysis of different types of anomalies will extract the most useful aggregation keys and features selection that can provide the best detection performances. For that reason, an important part of the proposed methodology corresponds to the anomaly modelling with the synthetic flows generation, combined with the used normal traffic dataset. For the needs of this research, as the main dataset with normal traffic instances we have used the CTU-13 labelled datasets [20], and applied the following modifications:

- Long flows are proportionally fragmented into shorten equivalent flows that last up to 60 seconds, what was set as a duration of one epoch.
- Background traffic, taken from the real university network, is by the authors originally left as unrecognized. We have manually analyzed this large portion of dataset (using our methodology) and labelled the minor anomalies. The rest of the background traffic is labelled as “normal” and further used as a regular traffic.
- Timestamp format is changed from string to milliseconds since the UNIX epoch (January 1, 1970 00:00:00 UTC).

The modelled synthetic traffic is created by a flow generator which is developed by Bereziński [6]. It is additionally modified in accordance to our own dataset format.

As a proof of the concept of our methodology we have developed a Java software, the EntropyCalculator for aggregation and calculation of the Top Talkers and entropy. The dataset format is configurable with JSON meta file, which describes the feature types, and denotes them as identification or volumetric features. The software supports interactive manual settings of the aggregation and processing parameters, but more efficient and useful usage is achieved with the bulk processing, where all the settings are defined in separate JSON file. The software also provides the data filtering, the definition of the serialized aggregation by different keys, the selection of the output features and entropy types.

We support calculation of Shannon, Tsallis, and Rényi entropy. Normalized Shannon entropy is defined by following equation [3]:

$$\mathrm{H_S(X)} = \frac{1}{\log_a n}\sum_{i=1}^{n} \mathrm{p(x_i)} \log_a \frac{1}{\mathrm{p(x_i)}} \tag{1}$$

Where n is a total number of elements in the distribution, while $p(x_i)$ is a probability of element x_i, calculated by sum of all values divided by the contribution of element x_i. It gives values between 0 and 1, where balanced values in distribution result in entropy near 1, while significant deviation in distribution values results to lower entropy value.

Tsallis and Rényi entropies are parametrized with parameter α, given by the following equations [1][2]:

$$\mathrm{H_{R\alpha}(X)} = \frac{1}{1-\alpha}\log_a\left(\sum_{i=1}^{n} \mathrm{p(x_i)}^{\alpha}\right) \tag{2}$$

$$\mathrm{H_{T\alpha}(X)} = \frac{1}{1-\alpha}\left(\sum_{i=1}^{n} \mathrm{p(x_i)}^{\alpha} - 1\right) \tag{3}$$

The Exponential Moving Average (EMA) technique is used for baselining of Top Talkers and entropy time series, including a baselining of their standard deviation. The predicted value for epoch n+1 is calculated recursively, taking into account previously baselined data B_n and observed data D_n in epoch n:

$$B_{n+1} = (1\text{-}\ \alpha_b)\ B_n + \alpha_b D_n \tag{4}$$

The coefficient α_b represents the degree of weighting decrease, and falls in range between 0 and 1. A lower value for α_b indicates stronger influence of the previously baselined value, resulting to the smoother baselining.

Similarly, the baselining of the standard deviation S is given by:

$$BS_{n+1} = (1\text{-}\ \alpha_s)\ BS_n + \alpha_s\ S_n \tag{5}$$

And finally, the tolerance range is defined by the predicted value of the feature B_{n+1} and predicted standard deviation for that feature,BS_{n+1}, as follows:

$$M = [B_{n+1} - k_t BS_{n+1},\ B_{n+1} + k_t BS_{n+1}] \tag{6}$$

Where k_t is the multiplication factor that makes the range wider, the so-called *factor of tolerance*. For any data D_{n+1} that falls out of the range M, an alarm is triggered as an indication of the potential anomaly.

With proper fine tuning of the used parameters, the proposed methodology achieves high efficiency of the anomaly detection. However, our architecture leaves an option to leverage ML techniques for further anomaly classification based on the trained datasets and learned anomaly patterns. The Hybrid Machine Learning module can overcome the needs for parameters fine tuning by detecting relative deviation from the margin of tolerance, and this way reducing the false alarm rate.

V. EXPERIMENTS AND RESULT ANALYSIS

Different types of anomalies have different behavior patterns and footprints of the calculated features, which trigger specific combinations of the alarms. The anomaly detection accuracy, with low rate of the false alarms, highly depends on the settings of several baselining and threshold factors. All these parameters must be fine-tuned according to specific network traffic, characterized by its unique behavior and intensity. Therefore, the alarms need to be treated as indicators of the anomaly, which require further correlated analysis, either by manual observation or by means of Hybrid Machine Learning module, which is proposed in our architecture as the complete solution [19]. For the reason of proofing the concept of our architecture, in this paper we have presented experimental results of entropy based part of our solution, leaving Machine Learning module as an option for improvement of detection accuracy.

Our experiments were based on CTU-13 dataset, which was additionally cleaned, leaving around 850.000 flows of real-life “normal” traffic, collected during 4 hours period, and divided into 60 seconds epochs. For the purpose of repeating the experiments with different anomaly models and scenarios, the flows in this datasets were randomized over the epochs, resulting in randomization of the noise in the regular behavior pattern, with no significant changes in the observed features.

The three anomaly types were modelled and further analyzed, namely DDoS NTP amplification [21], DDoS SYN Flood [22] and Network Scan attack [23]. The goal was to model low intensive anomalies and analyze features that are triggered by the corresponding traffic pattern. Therefore, we assumed the low data load given in packets and bytes measures, while gradually increasing the unique appearances of other features, starting from 10, 25, 50 up to 100. Each series lasted 3 epochs, and was repeated in 20 epoch interval. By default, Top Talkers count was 3. Baselining coefficients for the main data were α_b=0.1 and α_s=0.05 for the standard deviation, with the factor of tolerance k_t=4, by default.

A. Scenario 1 – DDoS NTP amplification attack

NTP amplification attack [21]exploits the weaknesses of some open NTP server, forcing it to reply to queries from unauthorized external hosts. The spoofing of the source IP address in the queries sent to many NTP servers will results in the generation of the replies from each of them to that single targeted host. The model of this kind of attack is characterized by UDP protocol, several source IP addresses (in our case we have defined 10, 25, 50 and 100), single source port (123), single destination IP address and random destination port. There is no answer from the opposite direction. We have modelled all of these series with modest total flow number of 200.

The behavior of this kind of anomalies has an impact to several features. Obviously, the targeted host and its IP address will experience the increase of the number of unique destination ports that should receive the traffic. Aggregation by destination IP address with the second degree defined with destination port can easily capture this pattern. Fig. 2 depicts the changes in volumetric Top Talkers features, while Fig. 3 shows the corresponding calculated entropy. Almost equally, the randomized degree by destination ports (DstPort) results with entropy value near 1, with small standard deviation for the regular traffic. The anomaly is therefore, without any doubts, detected and recognized even with the smaller values of the factor of tolerance k_t.

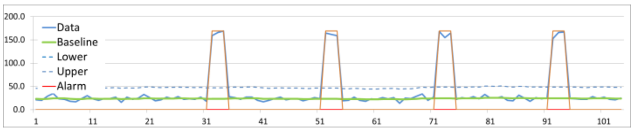

Figure 2. Aggregation by DstIP, Top Talkers of degree by DstPort

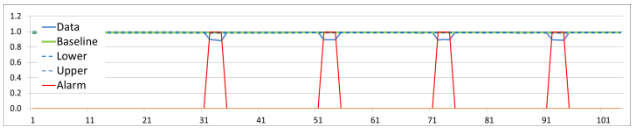

Figure 3. Aggregation by DstIP, entropy of degree by DstPort

The aggregation by DstPort exposes the opposite behavior during the anomalies – a lot of unique destination ports with one or few appearances of other degree features. This will not lead to changes in Top Talkers view, but will increase the entropy values of the corresponding degree features, for example the entropy of degree by destination IP addresses (DstIP), shown in Fig. 4.

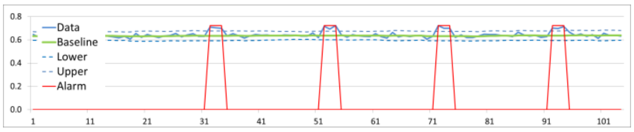

Figure 4. Aggregation by DstPort, entropy of degree by DstIP

The other features can also successfully trigger the alarm for this kind of anomaly, such as in the case when we set the aggregation by SrcPort and degree by DstPort and FlowCount or with the aggregation by DstIP-SrcPort pair and degree by SrcIP, DstPort and FlowCount.

B. Scenario 2 – DDoS SYN Flood attack

The synchronization flood is a kind of DDoS attack that is initialized from many source IP addresses and different source ports to one specific target destination IP address and port [22].It attempts to open and keep active as many TCP connections as possible. We have modelled the DDoS SYN Flood attack targeting the TCP port 443, related to the HTTPS web traffic. This communication pattern is very similar to the regular traffic generated by many commonly used services, such as web, email or DNS. For that reason, only the highly intensive anomalies can be detected. However, in our architecture we propose the flow classification into smaller subsets, where the detection process can be more sensitive to anomalies with lower intensity.

In our experiments, the anomalies with total of 200 flows with unique source IP addresses (generated from the pool of 10, 25, 50 and 100 defined addresses), stay undetectable in a dataset with more than 2000 flows and 600 unique source IP addresses of regular traffic per epoch. However, when the flow classification is applied, the regular web traffic consumes approximately 120 flows and 60 unique source IP address per epoch. When aggregating by destination IP address, the flow count feature successfully detects all four anomalies (Fig. 5). The similar results are obtained when applying the degree by source port, while in the case of degree by source IP address the algorithm is able to detect only the third and fourth anomaly, with 50 and 100 source IP addresses (not shown in the figures).

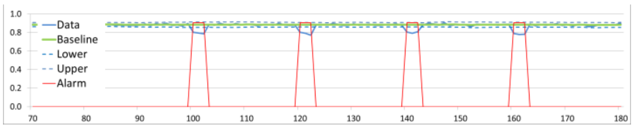

Figure 5. Aggregation by DstIP, entropy of FlowCount

C. Scenario 3 – Network scan for open SSH port

Very often the attackers are looking for the hosts running specific services, i.e. hosts with open targeted TCP port, in a way to exploit vulnerabilities on those services. We have modelled this kind of network scan by generating the TCP communication from fixed source IP address and random source port to various destination IP addresses (for the cases of 10, 25, 50 and 100 unique addresses) and SSH service with fixed destination port number of 22. We assumed 10% of successful attempts to open a connection, resulting in generation of the data transfer in opposite direction in order to complete TCP handshaking process.

This scenario represents an example when low intensive anomalies cannot be detected when exploring the total traffic, but can be analyzed only if the data is classified in smaller partitions. In this case, SSH traffic is extracted and analyzed separately from the other traffic instances. Since the regular traffic has just a few SSH flows per epoch, all series of anomalies are easily noticeable. However, low traffic load makes another problem since small number of specific data instances can lead to the unstable entropy values and unreliable results. In that case, the volumetric Top Talker features are more useful. Fig. 6 shows the degree by destination IP address when the aggregation is done by source IP addresses. Similar results are obtained when taking the degree by source port and flow count features, as well as with aggregation by destination port.

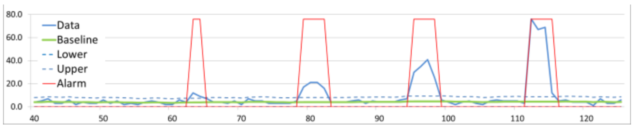

Figure 6. Aggregation by DstIP, entropy of FlowCount

VI. CONCLUSIONS

In this paper we have presented the architecture of flow-based anomaly detection system and methodology that can be applied on real-life use-cases in cybersecurity threats detection. Our experiments confirm that entropy based approach can successfully detect anomaly patterns of different well known security attacks, but they reveal that two problems still need to be solved for its practical implementation. Firstly, initial parameters for different features must be fine-tuned to adapt certain network traffic load, attack type and intensity in order to achieve acceptable accuracy with low number of false alarms. Secondly, a holistic analysis of all affected features is required to properly recognize the exact type of the attack and its consequences. For that reason our architecture involves a machine learning module and supporting analytics modules for alarming events visualization and management [19].

We also contribute to better understanding of hidden properties in flow-based datasets, extracted in the form of second degree behavior features. We have shown that these features outperform traditionally used volumetric and flow-count features for lower intensity attacks.

Our further work will be conducted in the directions of modelling new security attacks and anomalies, analysis of its influences and application of different types of decision processes, such as fuzzy-logic and neural networks.

ACKNOWLEDGMENT

This research is partially done by the project “Technical solution for security threat detection in computer networks”, supported by Innovation fund of Republic of Serbia, through Innovation voucher number 240 (2018-2019), and grants TR32037, TR32025, TR36035 supported by Ministry of Education, Science and Technological Development of Republic of Serbia.

VII. REFERENCES

[1] C. Tsallis, “Possible generalization of Boltzmann–Gibbs statistics,” *J. Stat. Phys.* 1988, 52, pp. 479–487.

[2] A. Rényi, “On measures of entropy and information,” In Proc. of the 4th Berkeley Symposium on Mathematics, Statistics and Probability; Neyman, J., Ed.; University of California Press: Berkeley, CA, USA, 1961, pp. 547–561.

[3] C. E. Shannon, “A mathematical theory of communication,” *Bell system technical journal*, 27(3), pp. 379-423, 1948.

[4] Claise, Benoit. Cisco systems netflow services export version 9. No. RFC 3954. 2004.

[5] Nychis, G.; Sekar, V.; Andersen, D.G.; Kim, H.; Zhang, H. 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; pp. 151–156.

[6] P. Bereziński, B. Jasiul, and M. Szpyrka, “An entropy-based network anomaly detection method,” *Entropy*, vol. 17, no. 4, pp. 2367–2408, 2015.

[7] W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, “Proceedings of the ninth international conference on dependability and complex systems DepCoS-RELCOMEX. June 30 – July 4, 2014, Brunów, Poland,” *Adv. Intell. Syst. Comput.*, vol. 286, pp. 47–48, 2014.

[8] T. Liu, Z. Wang, H. Wang, and K. Lu, “An entropy-based method for attack detection in large scale network,” *Int. J. Comput. Commun. Control*, vol. 7, no. 3, pp. 509–517, 2012.

[9] Winter, P., Lampesberger, H., Zeilinger, M., & Hermann, E. (2011, October). “On detecting abrupt changes in network entropy time series”. In IFIP International Conference on Communications and Multimedia Security (pp. 194-205). *Springer*, Berlin, Heidelberg.

[10] A. S. S. Navaz and V. Sangeetha, “Entropy based Anomaly Detection System to Prevent DDoS Attacks in Cloud,” *International Journal of Computer Applications* vol. 62, no. 15, pp. 42–47, 2013.

[11] G. Gu, P. Fogla, D. Dagon, W. Lee, and B. Skoric, “Towards an Information-Theoretic Framework for Analyzing Intrusion Detection Systems,” pp. 1–20.

[12] J. Santiago-paz, “On Entropy in Network Traffic Anomaly Detection,” no. November 2015, 2016.

[13] H. M. Shirazi, “Anomaly Intrusion Detection System Using Information Theory , K-NN and KMC Algorithms,” *Knowl. Creat. Diffus. Util.*, vol. 3, no. 3, pp. 2581–2597, 2009.

[14] Quittek, Jürgen, et al. Requirements for IP flow information export (IPFIX). No. RFC 3917. 2004.

[15] Juniper Networks, "Juniper Flow Monitoring: J-Flow on J Series Services Routers and Branch SRX Series Services Gateways," http://www.juniper.net/us/en/local/pdf/app-notes/3500204-en.pdf, Mar. 26, 2011, 10 pages.

[16] Huawei NetStream Analysis, Monitoring, and Reporting. 2019. Available: https://www.solarwinds.com/topics/netstream-analyzer

[17] Clowfd overview: 7950 XRS Router Configuration Guide. Available: https://infoproducts.alcatel-lucent.com/html/0_add-h-f/93-0073-10-01/7750_SR_OS_Router_Configuration_Guide/Cflowd-Intro.pdf

[18] Lawrance, A. J., and P. A. W. Lewis. "An exponential moving-average sequence and point process (EMA1)." *Journal of Applied Probability* 14, no. 1 (1977): 98-113.

[19] V. Timčenko, J. Ibrahim, S. Gajin, “The Hybrid Machine Learning Support for Entropy Based Network Traffic Anomaly Detection,” ICIST 2019, unpublished.

[20] Sebastian Garcia, Martin Grill, Jan Stiborek and Alejandro Zunino, "An empirical comparison of botnet detection methods" *Computers and Security Journal, Elsevier*. 2014. Vol 45, pp 100-123. http://dx.doi.org/10.1016/j.cose.2014.05.011

[21] Kührer, M., Hupperich, T., Rossow, C., Holz, T. (2014). Exit from Hell? Reducing the Impact of Amplification DDoS Attacks. In 23rd USENIX Security Symposium (USENIX Security 14) (pp. 111-125).

[22] Mirkovic, Jelena, and Peter Reiher. "A taxonomy of DDoS attack and DDoS defense mechanisms." *ACM SIGCOMM* Computer Communication Review 34.2 (2004): 39-53.

[23] Leckie, Christopher, and R. Kotagiri. "A probabilistic approach to detecting network scans." In NOMS 2002. IEEE/IFIP Network Operations and Management Symposium'Management Solutions for the New Communications World'(Cat. No. 02CH37327), pp. 359-372. *IEEE*, 2002