The Hybrid Machine Learning Support for Entropy Based Network Traffic Anomaly Detection

Valentina Timčenko[*], Juma Ibrahim[**] and Slavko Gajin [**]

[*] School of Electrical Engineering, Mihailo Pupin Institute, University of Belgrade, Belgrade, Serbia

[**] School of Electrical Engineering, Belgrade, Serbia

valentina.timcenko@pupin.rs, jumaibrahim04@yahoo.com, slavko.gajin@rcub.bg.ac.rs

Abstract— **This research relies on the proposed comprehensive flow based anomaly detection architecture, which is a complex solution that encompasses support modules for entropy calculation and for machine learning processing. The focus of this paper is on different machine learning algorithm performances in real-network scenarios. The research relies on the use of the modified CTU-13 dataset, entropy-based data preprocessing and performance analysis of a range of machine learning algorithms for modelled anomaly scenarios with synthetically generated flows. The architecture is an original solution, which is planned for further real-network application, targeting the possible support for a range of different use cases.**

I. Introduction

The large data volumes, fast, real-time application needs, and dynamic nature of the current network technologies tendencies (e.g. Internet of Things, IoT), combined with a rising trend of the security concerns, have imposed the need for reliable and accurate anomaly detection system that would monitor and analyze the network traffic behavior. This study targets the issue of providing an adequate solution for intrusion and anomaly detection in such environments. The recent research initiatives in this area have shown a great tendency towards the granulation of the systematic procedures within the anomaly and intrusion detection system architecture, thus an exhaustive data pre-processing is one of the system quality cornerstones, while at the other side there is need for strong, robust and accurate machine learning (ML) algorithms. The large data volumes, fast, real-time applications, dynamic nature of the network technologies (e.g. IoT), rising trend of the security concerns (malicious activities, rise of bandwidth IoT botnets) demand for reliable and fast NA/IDS.

Our focus is on the network environments for which of great importance is to keep security and efficiency at the highest levels. Thus, it is hard to provide a simple solution that would easily cover all the gaps and potential risks while providing fast response. The recent research tendencies indicate the need of merging several different techniques in order to approach the most efficient results. We have proposed the comprehensive flow based anomaly detection architecture that comprises two general parts: (1) the entropy based data flows pre-processing and (2) Hybrid Machine Learning. This paper is focused to the Hybrid Machine Learning (HML) module. It seeks the modalities for providing the support for the known and unknown attacks and anomalies, by implementing HML approach based on different supervised and unsupervised ML algorithms. For the needs of running a set of experiments we have used MATLAB and Weka environments, two prevalently used analysis platforms for this type of the research [1, 2].

The spotlight of this paper is the ML algorithm choice for network intrusion detection, considering the modified CTU-13 dataset [3]. Our focus is on the group of clustering algorithms, while for the needs of the further comparison, we have also provided analysis of a choice of supervised algorithm representatives [4-8]. The performance of the classifiers is further evaluated and compared in terms of their ability to correctly classify anomalous network behavior.

The remainder of this paper is structured as follows. Section 2 provides the information related to the applied methodology and related work. In section 3, we address the entropy based traffic analysis which is related to the Entropy Module of the proposed solution. The section 4 provides details on the considered ML methods, while the focus is on the clustering algorithms. In section 5 we present the solution with the spotlight on the HML Module, while in section 6 we provide comprehensive analysis of the obtained results. We conclude the paper with some final remarks and future work directions.

II. Methodology and Related Work

The malicious activity and network anomaly correspond to two different event categories, but from the intrusion detection system point of view we have to consider both of them with the same cautions and commitment. Both are in a range of actions that can compromise the integrity, confidentiality or availability of network, data, storage/CPU/energy resources, and last but not least it can jeopardize the guaranteed levels of the security and privacy. The potential exploitation of the system vulnerabilities is a long standing issue, whose detection has to be solved on the run, bearing with each new, previously unseen malware or anomaly.

We have proposed an all-encompassing anomaly and intrusion detection solution by combining the powerful techniques and methodologies for flow data pre-processing and ML analysis. In this study we explore our solution capacities on the use of traffic anomaly modelling, based on the modified CTU-13 dataset for normal traffic and synthetically generated traffic flows [9, 10]. For the needs of the proper definition of the algorithms and necessary parameter values that are used by HML Module, a particular submodule is using the feedback loop for providing continuous update of the training dataset with the newly labelled and with the synthetic flows. The synthetic flows are generated by the means of the Flow Generator tool, whereas for obtaining high-quality labelled flows there is a possibility to apply several models of the attacks and anomalies [9].

There are few studies that base their research in similar entropy-ML merging concept. In [11] the authors have utilized two-level stationary wavelet entropy (SWE) to extract a range of features from the available images dataset, for further ML classifier analysis, encompassing the decision tree, k-nearest neighbors (kNN), and support vector machine (SVM) algorithms.

At the other side, there is a possibility of using this merging for the needs of software behavior prediction, where a group of authors compare ML based regression techniques for predicting bugs using entropy of changes [12]. In [13], an importance of pattern recognition for imbalanced dataset issue is confronted, proposing an entropy-based matrix ML for imbalanced data sets. The authors have adopted the Matrix-pattern oriented Ho–Kashyap ML with regularization learning (MatMHKS) as the base classifier. Actually, the issue of the pattern recognition is a challenging task, especially if dealing with patterns that are associated with faults and malicious activities. This is mostly due to the highly discrete nature of network traffic. According to [14], there are three different approaches that can be applied: signature-based, statistical and informational/theoretical analysis.

III. ENTROPY BASED TRAFFIC ANALYSIS

The fast, dynamically changing network environments of the modern systems have put significant weight to already complex detection of the changes in the traffic pattern characteristics. The increase of the data volumes that are necessary to be processed and transmitted in real-time and the blooming of novel and heterogeneous technologies, have put a stronger light to the issue of protecting such environments. Entropy based detection techniques can handle large amounts of data and are highly related to the real-time traffic analysis. Entropy represents a measure of the uncertainty and randomness of a certain stochastic process. It is a measure of diversity or similarity in network traffic patterns, thus the characteristics of the traffic may be affected when tuning the values of certain traffic features. Actually, the changes in entropy values can indicate occurrence of the malware activity, attack, or anomaly. In order to disclose the regularity in traffic flows, the use of the probabilistic measure of entropy is proposed in [15]. According to this hypothesis, and taking the case of the scanning host, the entropy in a defined time window will change. Thus, when dealing with a number of flows with the same source IPs, it will result in sudden decrease of the entropy in the distribution of the source IP addresses. At the same time, this scanning host will try to bond with a number of IP addresses at the destination, and if possible with different destination ports, which will produce an increase in entropy measurements. The continuous monitoring of multiple entropy variations provides possibility of more accurate attack detection [16].

The first step towards the accurate and reliable detection of the malicious activity or network anomaly in traffic patterns is the proper selection of the set of optimal attributes. In that context, the primary issue is the choice of the criterion to evaluate the considered set of the attributes. Depending on the traffic pattern structure, characteristics and attack categories, there are multitude of possibilities to analyze and evaluate the dataset or real-time traffic, with the final goal to decrease complexity and increase the IDS capabilities. For proper functioning of the Entropy Calculation module, the proposed comprehensive flow-based anomaly detection architecture relies on the application of the Shannon, Tsallis, or Rényi entropy measures [17-21].

The basis of the concept information entropy concept is introduced in 1948, by Claude Shannon [19]. According to theory, the entropy provides “an absolute limit on the shortest possible average length of a lossless compression encoding of the data produced by a source, and if the entropy of the source is less than the channel capacity of the communication channel, the data generated by the source can be reliably communicated to the receiver”. Hence, time has brought some generalizations and also specific modifications according to certain area of research, such as the Rényi entropy, that generalizes the Shannon entropy and forms the basis of the concept of generalized dimensions [20]. On the other hand, the Tsallis entropy is increasingly used for the needs of a range of the natural, artificial and social complex systems that confirm the predictions and consequences that are derived from this non-additive entropy which generalizes the Boltzmann–Gibbs theory [21]. In [10] we have provided detailed information on the theoretical and applicative characteristics of the entropy mechanisms.

The entropy is sensitive to different types of traffic; therefore there is need to tune it with a goal to reduce the number of generated false positives (FP) and false negatives (FN). Thus, we can assume that, at least in the area of network traffic analysis, the bare use of only entropy techniques for the intrusion detection can be efficient but unfortunatelly not enough accurate. These techniques rely on the traffic feature distribution, whils we can categorize them as: (1) header-based, encompassing addresses, ports, flags; (2) volume-based, encompassing IP or port specific percentage of flows, packets and octets; (3) behavior-based dealing with the in and out connections. In order to obtain more accurate results, one of the possible ways is to combine entropy techniques with machine learning mechanisms [6].

IV. MACHINE LEARNING BASED TRAFFIC ANALYSIS

When designing a Network Intrusion Detection System, regardless of the network infrastructure that it will be applied, the key criteria are to obtain high accuracy and low values of false negative rate. The main problem is still the fact that there is need for flexible mechanisms that would help in treating right some variations of the traffic, in a way not to leave any attack or anomaly "below the radar" and undetected. The entropy based techniques are mostly used to indicate unusual traffic patterns, while with the application of the ML algorithms the system is further empowered to minimize false positives and false negatives, while achieving improved performances. The proposed architecture relies on the concept of the modularity and flexibility. It provides data preprocessing in order to build the optimal set of features that are responsive enough to provide fast and accurate input for ML. The HML (Hybrid Machine Learning) module is responsible for the combined implementation of the supervised and unsupervised ML algorithms, in order to provide efficient clustering and accurate traffic data classification. The data is previously preprocessed in the Entropy Calculation module, including traffic sub-classification and event extraction by the means of the root-cause analysis based loopback processing.

As a base support for the process of data and traffic classification, ML and Artificial Intelligence (AI) algorithms represent an essential issue to deal in the process of the proper IDS architecture design. Despite numerous shortcomings when confronted with modern traffic patterns and network characteristics, the supervised ML algorithms are still more present in the actual literature. The issue is that, although these algorithms rely on the labelled data sets, thus being easier for the implementation, some serious limitations are needed to be considered: the lack of truly accurately labelled data sets, limited possibilities of the implementation in the real network production, lack of the mechanisms against zero-day attacks and vulnerabilities caused by unpredictable network traffic behavior.

The most popular supervised ML algorithms, which we have practiced in our research so far, are as following: SVM, the ensemble algorithms that are much appreciated for the case of the imbalanced datasets (Boosting, Bagging, Random Forest (RF), etc.), k-nearest neighbors (k-NN) and Neural Networks based on the Multi-Layer Perceptron (MLP) mechanism [7, 8].

The benefits of the unsupervised ML algorithms rely in the fact that the basic criterion for detection, grouping of the events, or only assumption of malware activity, relies on the similarities and differences in the data structure, time limits, even though there are no categories provided. There is no training data set, thus the sophistication of the interpretation is of major concern.

Unsupervised learning is applied in the case when there is no labelled data available. The unlabelled data usually corresponds to the previously unknown data for certain IDS. It represents the basis for the unsupervised learning methodology, where the goal is now to find specific structural characteristics of the analyzed traffic. The purpose of the algorithm is to divide data into meaningful groups of similar data called “clusters”, strongly relying on the captured nature and structure of the input data. Most of the clustering algorithms rely on the assumption of the fixed number of clusters. In real-network situations it is hard to estimate in advance the number of clusters. If the assumed number is small then there is a big possibility of adding unrelated elements into the same group; if it is large, then there is a higher chance of adding similar data into different groups. Some of the clustering algorithms applied in this research are as follows:

K-means is an iterative clustering algorithm that performs with objective to find local maxima. It initially randomly locates *k* centroids into the area of interest. With the application of the Euclidean distance method, the algorithm further measures the distance between data points and centroid locations, thus assigning data to the closest cluster. Every subsequent iteration recalculates the centers of the clusters, and if necessary provides the data point reassignment where necessary. It highly depends on the proper choice of the centroid number and location, and as the result can be volatile, it can provoke unnecessary increase of the number of iterations. Accordingly, the time and space complexity increases proportionally [4].

Hierarchical clustering algorithm is generating the hierarchy of clusters. It initializes with all data points assigned to a group of their own. Using the Euclidean distance, in each next step this algorithm combines the clusters and merges the nearest groups into the same cluster, until all data is clustered into a single cluster.

Farthest First clustering also relies on proper choose of the centroid points, but the data point assignment to the clusters is provided with maximum distance. This point must lie within the data area. The points that are farther are clustered together first, which speeds up the clustering process, and lowers the number of the necessary iterations for reassignment and adjustment [5].

Filtered Clusterer represents a Weka meta-clustering algorithm which applies an arbitrary filter before applying the random clustering algorithm. As in the case of the clusterer, the structure of the filter depends on the available training data and test instances, which are filtered without changing their structure [1].

Sequential information bottleneck (sIB) clustering supports only the hard clustering scheme. It is mostly applied in the area of unsupervised document organization. For each instance it assigns the data point to the cluster which has the minimum cost/distance to the instance. In the context of document clustering, measure of similarity of two documents is the similarity between their word conditional distributions [22].

V. Hybrid Machine Learning (HML) Module for Entropy Based Anomaly Detection

The actual footprint of the technological development leaves only a small room for any improvisation, thus strong security, along with proper solution for high quality of the interoperability, is of the top imperative. The network anomalies and malicious activities are strong damage holders, thus the IDS relying only on the basic signature detection are slowly dying in the live mud. The uncertainty of the user profile changes, the frequent occurrence of the previously unknown traffic, and appearance of the security measure avoiding intelligent botnets has further raised the already high expectations of the modern security solutions. The main IDS requirements are efficiency, adaptibility, flexibility, applicability to the data-intensive networks. In order to contribute to the solution of this issue, in this paper we are providing the highlights of our solution, with the focus to the Hybrid Machine Learning module (Figure 1) of the proposed architecture [10]. The proposed solution is assumed for further real-world implementation, and considers the core of the anomaly detection application challenges for the real use cases. It combines the benefits of the entropy based techniques for data preprocessing with further implementation of the range of the ML algorithms, thus properly processing the data, labeling and finally providing the clustering and classification as a final result. As it is explained in [10], we are applying entropy mechanisms in order to obtain the indication of the unfamiliar traffic, while ML algorithms provide a core support for the decrease of the number of false positive and false negative alarms, and enhancement of the performances.

The HML Module is a complex unit that relies on the application of a range of different supervised and unsupervised algorithms [7, 8]. HML is responsible for the efficient and accurate differentiation and classification of changes in network behavior pattern indicated by the Entropy Calculation Module. The input data is a complex set of combined data collected from the network, dataset and synthetically modeled data.

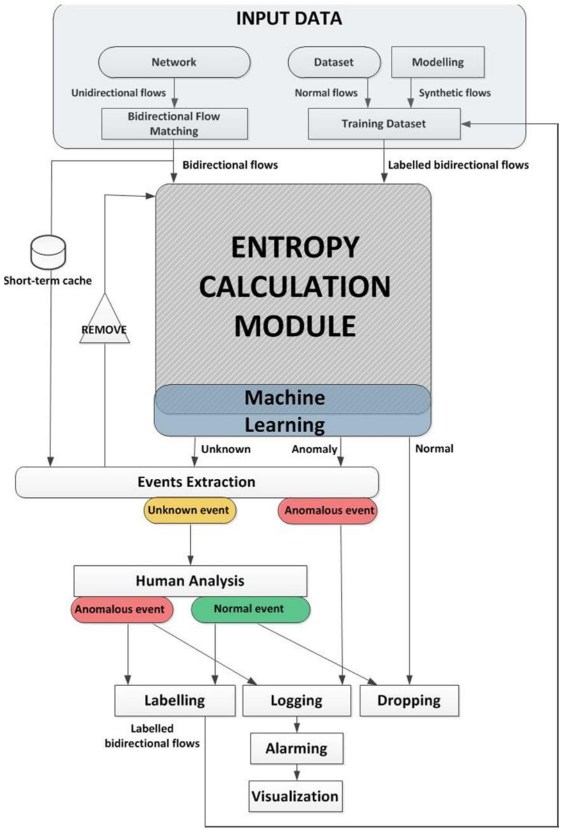

Figure 1. Hybrid Machine Learning Module.

HML module therefore recognizes: (1) “Normal” indicator, which does not require any further actions; (2) “Anomaly” indicator, which is classified by its type; and (3) “Unknown” indicator, which is related to the events for which there is no sufficient reliability for proper decision.

For “Anomaly” and “Unknown” outputs, a root cause analysis is performed by extracting the flows that caused a specific event (using **Event Extraction Module**). Those events are now enriched with full set of information for further processing. The successfully detected and isolated “Anomaly” events are directed towards the **Logging** block and further to the **Alarming** block for short-term/long-term analysis and visualization. Manually analyzed events are recognized as “Normal” or “Anomaly”, and further processed by the **Labelling** block for generating specific loop input to the **Training Dataset,** as a part of “self-learning” process. We are eliminating from the further analysis all the events that are not necessary to be pushed for evaluation or logging. Thus, the data is continuously updated and refreshed, providing the input to the **Training Dataset block.** This input is formed by a combination of the „Normal flows“, „Synthetic flows“ and „Labelling“ data obtained as output from the HML module.

Another segment related to the HML module corresponds to the modeling of the anomalous traffic and generation of the synthetic flows. The anomalous and malicious traffic instances are generated in order to further provide better training routine of the machine learning algorithms. These are generated with the Flow Generator software, which provides modelling of different traffic profiles. Flow Generation module is based on the tool provided for scientific use from the Polish National Centre for Research and Development [9]. It encompasses different synthetic traffic models, including the DDoS, brute force attack, port scan, exploits, etc.

The results obtained from counting the confusion matrix values for correctly and incorrectly detected events (that belong either to the normal or attack/anomaly class), represent the basis for calculating the clustering and classification efficiency measures. Confusion matrix provides information on true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). These are used to measure accuracy, sensitivity, specificity, precision, and ROC [8]. The accuracy is the percentage of correctly classified instances (1).

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

As the ROC Area Under Curve (AUC) values are calculated based on sensitivity and specificity of the IDS (2), (3), the ROC graph represents "Sensitivity vs. (1 − Specificity)" plot. The Sensitivity or Recall, stands for true positive rate (TPR). It is a metric that provides information on the proportion of all the instances of a particular class that are correctly classified as that class. The Specificity defines true negative rate (TNR):

$$sensitivity = TPR = \frac{TP}{TP + FN} \tag{2}$$

$$specificity = TNR = \frac{TN}{TN + FP} \tag{3}$$

VI. Experiments and Result Analysis

The experimental performance analysis of the chosen clustering algorithms over the modified CTU-13 dataset is carried on in Weka (v3.6) and MATLAB R2017b (v9.2), in Windows environments on 3GHz Intel(R)Core(TM)and 8GB RAM. For the needs of HML evaluation, we have tested the clustering algorithms K-means, Farthest First, Hierarchical clustering, sIB, Filtered Clustering, and supervised algorithms, namely SVM, RF, Random Tree (RT), Logit Boost (LB), and MLP.

A. Dataset CTU-13

The CTU-13 dataset is composed of 13 different malware captures in a real network environment. It includes 7 botnet malwares, which are producing email spam, click fraud, and DDoS activities. The original dataset includes labels for: Background, Botnet, C&C Channels and Normal traffic instances. The background traffic corresponds to the events that the authors have left undecided as malicious or not. It contains 14 features, while with our modification we have introduced several new derived features. It is slightly unbalanced dataset, thus needs specific preprocessing and cautious choice of the ML algorithms. The main limitation is with background traffic, as it is said that is obtained from a university router, but there is no sufficient information related to the topology or services. CTU-13 is a good representative labelled dataset for the analysis of the botnet-like environments (including IoT). It is further processed in order to obtain NetFlows. It encompasses several categories of botnet traffic, namely Neris, Rbot, Sogou, Murlo, and Menti, extracted from the *Malware Capture Facility Project*, a research project with the purpose of generating and capturing botnet traces in long term [23]. For the needs of this study and for greater validity of the used dataset, there was need to improve its characteristics, and put the efforts to provide a number of enhancements:

- Cleaning, labeling other anomalies, provide the flow fragmentation, and generation of new features.
- Dataset expansion with model-dependent synthetic flows.
- ML analysis in order to consolidate and improve Entropy Calculation module results.

B. *Entropy Calculator application*

Entropy Calculator application is functional Java based software developed for the needs of research activities on the Innovation fund project *Technical solution for security threat detection in computer networks*. It allows to the user the aggregation of the network traffic according to the desired attributes and further calculates the entropy. The software provides possibilities of setting individual performance analysis parameters, and provides efficiency optimization in anomaly/attack detection. The performance analysis of the entropy calculation is based on the needed memory resources along with the necessary execution time. Besides, Entropy Calculator is exporting the calculated data for further processing and analysis.

C. *Flow generator*

Flow Generator software provides modelling of different traffic profiles [9]. It encompasses different synthetic traffic models, including the DDoS, brute force attack, port scan, exploits, etc. The procedure for synthetic flows generation and use encompasses: (1) model generation; (2) incorporation of the synthetic traffic into the main data source; (3) estimation of the available features for generation of the important features list; (4) application during the HML production mode.

One of the challenges is the optimal selection of the entropy based features for further ML analysis. MATLAB provides a possibility of generating the Parallel Coordinates Plot (PCP) that can be used for easier feature selection. It can be generates either as normalized or standardized plot. The datasets that are based on the larger number of variables bring an issue of more difficult feature direct visualization. The use of the PCP permits the all-together display of the variables, allowing the analysis of their higher-dimensional relationship. An example with selected features is provided in Fig 2. A display with this much data cannot be used to explore the details, but it can be used to search for predominant patterns and exceptions. This helped us to narrow down the choice of the calculated entropy features from 54 newly generated features based on the original dataset.

This number is reduced in further analysis; each observation is represented by the sequence of its coordinate values plotted against their coordinate indices.

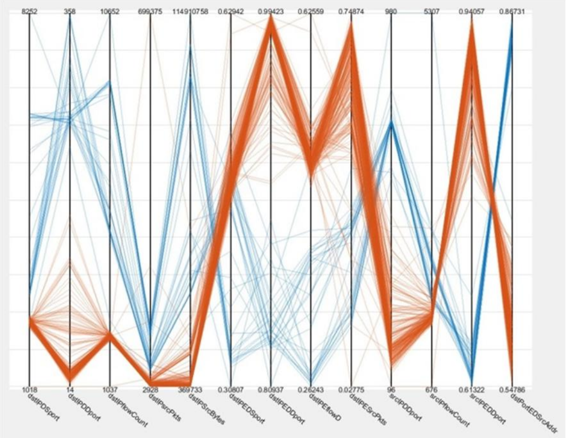

Figure 2. Normalized PCP for modified CTU-13 dataset analysis.

Finally, the data contains 13 chosen measurements. The position where one of these lines intersects a vertical axis indicates that product's value for that variable, from two types of traffic, normal(red)/botnet(blue), and for each of the following entropy measurements: dstIPDSport, dstIPDDport, dstIPflowCount, dstIPsrcPkts, dstIPsrcBytes, dstIPEDSport, dstIPEDDport, dstIPEflowD, dstIPESrcPkts, srcIPDDport, srcIPflowCount, srcIPEDDPort, dstPortEDSrcAddr. The measures that make clear distinction between the two traffic types are assumed as beneficial for further analysis.

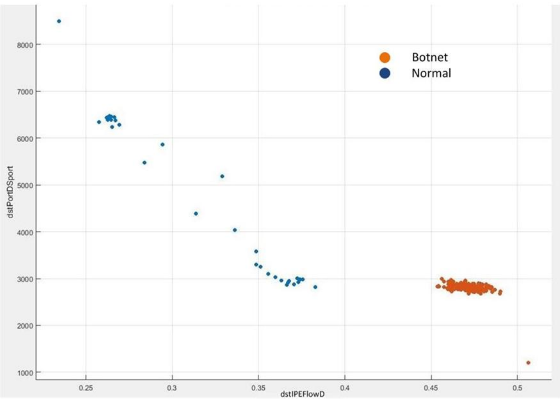

Figure 3. MATLAB Scatter Plot results.

In this paper we are providing a set of the initial results obtained so far. We have proceeded with MATLAB analysis for SVM, Random Forest, Random Tree, MLP and LogitB and then analyzed the performances that are achievable when applying a range of clustering algorithm in Weka environment. From the obtained set of features, we have further evaluated their mutual correlations. Figure 3 shows an example of the scatter plot obtained while proceeding with MATLAB analysis. The features of interest are those that when combined in analysis provides better separation of the two traffic types. Table 1 provides information on the obtained results counts for correctly/incorrectly classified/clustered data, provided in percent. The first column corresponds to the results obtained in the case of the first pass through the algorithm, without applying entropy calculation and elimination of any instance from further analysis. The second and third column correspond to the obtained results for the cases when first, and second time applying the entropy calculation and elimination of some of the instances that are categorized as not providing anomaly or attack harassment (procedure provides cleansing of the input data for HML thus allowing better performances). The experimental results provide the proof of the concept of the solution, demonstrating that for each subsequent pass through the ECM and HML module, the algorithm provides enhanced results, better accuracy, and procedure acceleration.

TABLE I. CORRECTLY/INCORRECTLY CLASSIFIED EVENTS

Algorithm	Initial dataset	1^{st} calc.	2^{nd} calc.
MLP	**98.58/1.42**	**98.74/1.26**	**99.58/0.42**
SVM	**99.07/0.93**	**99.16/0.84**	**99.16/0.84**
K means	**72.7/27.3**	**93.28/6.72**	**98.74/1.26**
FarthestFirst	**84.8/15.2**	**94.53/5.47**	**94.11/5.89**
Filtered	**72.7/27.3**	**88.6/11.4**	**98.74/1.26**
Hierarchical	**85.4/14.6**	**89.45/10.55**	**95.38/4.62**
sIB	**76.9/23.1**	**90.75/9.25**	**98.74/1.26**
LogitBoost	**98.74/1.26**	**99.58/0.42**	**99.58/0.42**
Rand. Forest	**99.58/0.42**	**99.58/0.42**	**99.58/0.42**
Rand. Tree	**97.9/2.1**	**98.32/1.68**	**99.16/0.84**

Among the clustering algorithms the best performances are obtained with sIB algorithm, as it provides highest values in the 2^{rd}, and also at the initial dataset pass. When properly labelling the instances (for ML comparison reasons), we can use the supervised algorithms but it requires manual labelling and additional verification which is time and CPU demanding. These are initial results with smaller amount of input data, and we expected some differences with further dataset modifications and integration of attack flows.

VII. CONCLUSION AND FUTURE WORK

The main contribution of this research is the proof of the concept for the ML detection and classification of network anomalies, based on a set of the modelled scenarios [10]. It relies on the proposed comprehensive flow based anomaly detection architecture, properly modified CTU-13 dataset which is additionally expanded with model-dependent synthetic flows, entropy calculation and ML with the application of different supervised/ unsupervised algorithms. This architecture is a potential solution that would rely on the plexus of the ML approaches with high accuracy, decreased false alarm, low memory and computation consumption. The focus is on the unsupervised ML algorithms, as these, when optimally chosen, can provide high score results for unknown traffic profiles. We are putting the efforts on improving the input sources, the usefulness of the data information and better integration of HML with Entropy Calculation module. These initial results have confirmed the expected behavior, as the entropy calculation preprocessing has brought better results to the ML module. The solution is planned for implementation as a result from the project "Technical solution for security threat detection in computer networks", supported by Innovation fund of Republic of Serbia, Innovation voucher number 240 [5].

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), grants TR32037, TR32025 and TR36035 supported by Ministry of Education, Science and Technological Development of Republic of Serbia.

REFERENCES

[1] E. Frank et al, "Data Mining: Practical Machine Learning Tools and Techniques," The WEKA Workbench, M. Kaufmann, 2016.

[2] Classification Learner - MATLAB. The MathWorks.

[3] S. Garcia, V. Uhlir, "The CTU-13 dataset. A Labeled Dataset with Botnet, Normal and Background Traffic," 2011.

[4] A. Coates, Y. Ng. Andrew, "Learning feature representations with k-means," *Neural networks: Tricks of the trade*. Springer, Berlin, Heidelberg, pp. 561-580, 2012.

[5] M. Panda, MR. Patra, "A novel classification via clustering method for anomaly based network intrusion detection system," Int. Journal of Recent Trends in Engineering 2.1 (2009): 1.

[6] R. Jenssen, et al., "Clustering using Renyi's entropy," Proc. of the Inter. Joint Conference on Neural Networks, 2003. Vol. 1. IEEE.

[7] V. Timčenko, S. Gajin, "Machine Learning based Network Anomaly Detection for IoT environments," ICIST 2018 Proceedings vol.1, pp.196-201, 2018.

[8] V. Timčenko, S. Gajin, "Ensemble classifiers for supervised anomaly based network intrusion detection," 13th IEEE Int. Conf. on Intelligent Computer Communication and Processing (ICCP), pp. 13-19, 2017.

[9] P. Bereziński et al, "Network anomaly detection using parameterized entropy," Int. Conf. on Computer Information Systems and Industrial Management, Springer, pp. 465-478, 2014.

[10] J. Ibrahim, V. Timčenko, S. Gajin, "A comprehensive flow/based anomaly detection architecture using entropy calculation and machine learning classification," ICIST 2019, unpublished.

[11] Y. Zhang et al., "Comparison of machine learning methods for stationary wavelet entropy-based multiple sclerosis detection: decision tree, k-nearest neighbors, and support vector machine," *SIMULATION*, 92(9), pp. 861–871, 2016.

[12] A. Kaur, K. Kaur, D. Chopra, "An empirical study of software entropy based bug prediction using machine learning." International Journal of System Assurance Engineering and Management 8.2, pp. 599-616, 2017.

[13] C. Zhu, Z. Wang, "Entropy-based matrix learning machine for imbalanced data sets," Pattern Recognition Letters 88, pp. 72-80, 2010.

[14] R. E. Eimann, "Network event detection with entropy measures (Doctoral dissertation, ResearchSpace@ Auckland), 2008.

[15] A.Wagner, B. Plattner, "Entropy based worm and anomaly detection in fast IP networks", In Proc. 14th IEEE Int. Workshops on Enabling Technologies Infrastructure for Collaborative Enterprise, 2005.

[16] A. Sperotto et al., "A labeled data set for flow-based intrusion detection," Int. Workshop on IP Operations and Management. Springer Berlin Heidelberg, 2009, pp. 39-50.

[17] J. Amigó, S. Balogh, S. Hernández, "A Brief Review of Generalized Entropies,". *Entropy*, 20(11), 813, 2018.

[18] C. F. L. Lima, F. M. Assis, C. P. de Souza, "A comparative study of use of Shannon, Rényi and Tsallis entropy for attribute selecting in network intrusion detection," Int. Workshop on Measurements and Networking Proceedings, pp. 77-82, 2011.

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

[20] 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.

[21] C. Tsallis, "Possible generalization of Boltzmann–Gibbs statistics," J. Stat. Phys. 1988, 52, pp. 479–487.

[22] N. Slonim, N. Friedman, N. Tishby, "Unsupervised document classification using sequential information maximization," Proc. of the 25th Int. ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 129-136, 2002.

[23] S. García, Malware Capture Facility Project. CVUT University. Dataset CTU-Malware-Capture-Botnet-1. 2013.