Data Analytics on Network Traffic Flows for Botnet Behaviour Detection

Duc C. Le
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia, Canada
Email: lcd@dal.ca

A. Nur Zincir-Heywood
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia, Canada
Email: zincir@cs.dal.ca

Malcolm I. Heywood
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia, Canada
Email: mheywood@cs.dal.ca

Abstract—**Botnets represent one of the most destructive cybersecurity threats. Given the evolution of the structures and protocols botnets use, many machine learning approaches have been proposed for botnet analysis and detection. In the literature, intrusion and anomaly detection systems based on unsupervised learning techniques showed promising performances. In this paper, we investigate the capability of employing the Self-Organizing Map (SOM), an unsupervised learning technique as a data analytics system. In doing so, our aim is to understand how far such an approach could be pushed to analyze unknown traffic to detect botnets. To this end, we employed three different unsupervised training schemes using publicly available botnet data sets. Our results show that SOMs possess high potential as a data analytics tool on unknown traffic. They can identify the botnet and normal flows with high confidence approximately 99% of the time on the data sets employed in this work.**

I. Introduction

There is a wide variety of network threats on the Internet, with different aims and attack vectors. Among these, botnets have become one of the most dangerous threats [1] [2]. Botnets consist of compromised machines, or bots, dominated by attackers (the botmasters) through command and control (CC) communication channels. Botnets are responsible for many types of attacks these days, including but not limited to spam spreading, distributed denial of service (DDoS) attacks, distribution of malicious software, information harvesting and identity theft.

A botnet maintains its virulence by evolving its structure and protocols over time. One component of a botnet that has been through many evolutions is CC channels. A botnet CC channel accommodates communications between bots and bot masters, which differentiate botnets from other malwares. The communication channels provide botnets the ability of updating its malicious code and protocols, allow bots to perform attacks simultaneously under the control of a botmaster. Thus CC channel one of the targets of security researchers in order to take botnets down. Earlier botnets use Internet Relay Chat (IRC) as their CC protocol. Eventually, as this protocol and botnet structures became obsolete and started to be detected easily, botnets abused a wide range of other protocols from HyperText Transfer Protocol (HTTP), HTTPS (secure HTTP) to Peer-to-Peer (P2P), email, and social network [3][4].

In general, botnets have two main architectures, or CC infrastructures: Centralized and Decentralized. In the centralized architecture, all bots establish their communication channel with one or a few central control servers typically over IRC and HTTP protocols. The obvious advantages of this topology are speedy command propagation and synchronization. However, while most earlier botnets are centralized, decentralized CC is increasingly employed in recent years to overcome central point of failure problem. By utilizing P2P protocols to allow each node in a botnet act as a client or a master, decentralized CC provides great flexibility and robustness. Moreover, a botnet topology can be a hybrid model of the two architectures to combine advantages of both CC models.

Given the threats posed by botnets, botnet detection has become a critical component in network security solutions. Machine learning-based approaches are used for their ability to learn underlying patterns of data and adaptation to the dynamic nature of modern botnets. Moreover, to identify novel botnets in particular, and malicious network activities in general, anomaly detection systems based on unsupervised machine learning methods are gaining more and more interest [5].

In this work, we assess the capability of an unsupervised neural network technique, namely Kohonen's Self Organizing Map (SOM) [6], as an unsupervised learning approach for traffic analysis to identify botnets. We study the effect of different training schemes under unsupervised learning paradigm to identify (detect) botnet traffic. Specifically, we employ the following three SOM training schemes: (i) using traffic flows of both Normal and known Botnet behaviours, (ii) using traffic flows of only Normal behaviours, and (iii) using traffic flows of only known Botnet behaviours. Obtained results demonstrate the promising capability of SOM in separating Normal and Botnet behaviours, as well as in labelling unknown traffic for further investigation.

The remainder of the paper is organized as follows. Section II summarizes the related work on botnet detection and applications of SOM in this field. Section III discusses the methodology, whereas Section IV presents the evaluations and results. Finally conclusions are drawn and the future work is discussed in Section V.

II. Related work

Botnet detection approaches have evolved extensively and expeditiously to cope with the development in botnet architectures and protocols. Early researches and commercial products, e.g. Snort [7], mainly based on comparing signatures with packet content to identify malicious activities. Gu et al. [8] used a botnet life-cycle model to develop BotHunter, which correlates alerts generated using Snort to detect botnets. Wurzinger et al. proposed a botnet detection model based on the observable command and response patterns of the botnet communications [9]. To build the patterns, their approach identifies responses and inspects the preceding traffic. Their results showed that the automatically extracted detection models could outperform BotHunter. Botminer, an approach based on group behavior analysis, combines both packet payload and network flow monitors for botnet detection [10]. The model employs clustering approaches to find similar communication behaviors, as well as network activities. Correlations between the formed clusters is used to identify botnets and infected hosts. Zhao et al. investigated a botnet detection system based on packet header information and time intervals [11]. Decision Tree based machine learning algorithms were utilized to generate detection models using network flow features of traffic packets. On their generated data set focusing on P2P botnets, their method achieved high accuracy with small time windows. Recently, Haddadi et al. employed three machine learning algorithms, namely C4.5 Decision tree, Bayesian Networks and Genetic programming-based SBB, for building detection models [12]. They achieved very high detection rates both for HTTP and P2P based botnets.

While most of the machine learning approaches for botnet detection are based on supervised learning, unsupervised learning approaches have also found their applications in the field, especially in anomaly detection systems. Leung et al. proposed a density-based and grid-based clustering algorithm to discover the characteristics of the majority of connections in network traffic [13]. They used these characteristics to classify future connections. Evaluated using the 1999 KDD Cup data set, the technique produced comparable results to existing supervised approaches. Kayacik et al. proposed an approach to network intrusion detection based on a hierarchy of SOMs [14]. Using 1999 KDD Cup data set for training, two hierarchical SOM architectures were proposed. The first model uses only six basic features from the data set and generates a three-layered SOM hierarchy, where the first layer SOMs are used to generalize data from each feature individually. Output of each first-level SOM is clustered to six clusters for higher-layer training. The second model uses all 41 features to directly train a two-layer SOM model, which is similar to the second and third layers in the first model. Ippolity et al. developed a threshold based training process for Adaptive Growing Hierarchical SOM for building an online network intrusion detection system [15]. In their work, system parameters are adjusted dynamically by using quantization error feedback to adapt to the new training data. The results on 1999 KDD Cup data set show enhancement over performance of previous approaches.

The approaches based on unsupervised learning in the aforementioned works provided comparable results to that of supervised learning approaches. Moreover, unsupervised learning methods enable an intrusion detection system to potentially generalize the learned models (based on training) on novel threats, i.e. anomaly detection. The fact that in practice either there is no labelled data or there is very few labelled data makes employing unsupervised learning approaches preferable in these cases. To this end, an unsupervised approach with visualization would be most supportive for a human expert to analyze the data. Given that SOM has the ability to build a topographic visualization for the data, we believe that it is a good match.

Most of the previous works that employed unsupervised learning techniques for network intrusion detection were tested against outdated data sets (e.g. 1999 KDD Cup). Note that KDD cup data set has many drawbacks [16], and is already well-investigated by proposed detection models using packet statistics. This raises the question about performance of such systems on new publicly available data sets representing modern botnets. Furthermore, since recent botnets also exploit traffic encryption to hide their malicious activities, aforementioned literature, which use packet payload for training, became obsolete. Hence, an approach not utilizing encrypted payload information may improve the state-of-the-art in using unsupervised learning based traffic analysis.

III. Methodology

As discussed earlier, the goal of this work is to assess the capability of using SOMs as an unsupervised machine learning approach in botnet traffic analysis. Our hypothesis in this work is that given sufficient resolution, the trained SOM may form well-separated regions to differentiate distinct botnet communication behaviours (based on the traffic analyzed), each into one or more node regions in the map. While a similar objective, using the SOM as a semi-supervised approach for attack detection, was explored in the past [14], this work focuses on the unsupervised training approach alone, in which only one layer of SOM is used for data projection. We believe that the approach can be applied to more scenarios, including the emerging threats of new types of botnets. To this end, we aim to study the effect of training data sets and their nature on the capabilities of SOM as a data analytics tool for botnet traffic analysis.

The network traffic data used to build the SOM is exported as flows, which are statistics based on the header information but not the payload of traffic packets. A flow is defined as an artificial logical equivalent to a call or connection, which connects a pair of terminals and contains a group of features [17]. A flow is commonly identified by a set of five different attributes (5-tuples), including source and destination Internet Protocol (IP) addresses, source and destination port numbers, and the protocol, over a predetermined duration.

A. Traffic employed

To ensure a wide range of behaviours and botnet categories, five publicly available botnet traffic traces from CTU13 set provided by Malware Capture Facility Project of Czech Technical University [18], and ISOT data set provided by University of Victoria (UVIC) [11] are chosen. The traces contain botnet captures with a wide variety of malicious behaviours and protocols, as well as different botnet architectures and attack targets. By choosing such a diverse set of botnet traffic traces (data sets), we intend to explore the performance of the SOM as an unsupervised learning technique under different scenarios and determine how well the approach is in terms of generalization.

The CTU13 botnet traffic data sets were captured in 2011. The goal was to have a large database of real botnet traffic mixed with normal traffic and background (unknown) traffic. These data sets consist of thirteen traffic traces of different botnet samples. Under each scenario (botnet sample), a specific malware was executed where each of them established connections on several protocols and performed different actions. Four chosen traces in CTU13 data sets include Murlo, Neris, Rbot, Virut. These are referred to as captures 8, 9, 10, and 13 respectively. The Murlo trace (capture 8) contains mainly port scans as malicious behaviour, with proprietary command and control protocol, Net-BIOS and STUN traffic. Neris botnet found in capture 9 contains Spam spreading, ClickFraud and Port scanning, while Rbot sample found in capture 10 contains UDP DDoS attack traffic. Both of them are based on IRC protocol. On the other hand, Virut botnet found in capture 13 contains mainly Spam spreading and port scanning actions, which are based on HTTP protocol.

García et al. discuss in [18] that the labelling process in CTU13 ensures that all flows labelled as normal and botnet are definitely normal / botnet, while flows labelled as Background may contain traffic from both types. This means that in each CTU13 data set, there is an unlabelled portion for further exploration. We refer to this portion (background) as the unknown portion of the data.

On the other hand, ISOT data set is the combination of several publicly available malicious and non-malicious data sets, including Lawrence Berkeley National Laboratory's traffic traces for legitimate [19], and background traffic and Storm and Waledac botnet traffic from the French chapter of honeynet project [20]. Both botnets in ISOT data set employ decentralized architectures, while Waledac is a P2P based botnet, Storm utilizes HTTP and Fast-flux techniques based on the DNS protocol. These botnets generate SMTP Spam and UDP traffic. It is also noteworthy that most of botnets in this work exploit traffic encryption for hiding malicious actions. Hence, the analysis and detection of such traffic behaviours is not trivial.

From the given flow records for the CTU13 data set [21], we employ all numerical features, as well as protocol and protocol dependent state fields. The features are: the duration, port numbers, the direction, source and destination types of services, the number of packets, the number of bytes, the number of source bytes in numerical format, and the protocol, connection states in binary format. By using only the provided basic flow characteristics, we intend to test the performance of our proposed approach using minimum *a priori* information. By minimizing the *a priori* information, we aim to minimize the blind sights and not to miss the new (unknown) malicious behaviours. On the other hand, we employ all numerical features, extracted from ISOT data set using Tranalyzer with default configuration [22]. It is also noteworthy that our approach does not use IP addresses or port numbers as input attributes to build the SOM.

B. Self-Organizing Maps

Self-organizing map, or Kohonen's map is one of the most popular unsupervised neural network models [6]. The algorithm is based on unsupervised, competitive learning to produce a $d_1 \times d_2$ two-dimensional map (grid) projection of multi-dimensional input space. Basically a SOM consists of components called nodes or neurons. Each node has a weight vector with the same number of dimensions as the input vector, as well as a fixed a position in the map plane, which is typically a hexagonal or a rectangular grid. Then for each training vector, the algorithm calculates distances between input vectors and SOM nodes to choose the best-matching unit (BMU), and updates the weight vectors of the BMU (the hit) and its neighbours accordingly for the training process. The basic iterative learning procedure can be summarized as follows:

1) Assign a weight vector to each map node (unit) w_{ij} randomly or linearly.
2) At each training step, a random input vector x is presented to the lattice. Distances, typically Euclidean, between x and all the nodes in the SOM are computed.
3) The winning node w_c is identified by minimum distance to the input vector. $d(w_c, x) = min(||x - w_{ij}||)$, where $||.||$ is the Euclidean norm.
4) Weight vectors of the winning neuron and its neighbors are adjusted according to the input vector: $w_{ij}(t+1) = w_{ij}(t) + hc_{ij}(t)(x - w_{ij}(t))$, where hc_{ij} is a non-increasing neighborhood function around the winner w_c. In case of Gaussian neighborhood:

$$hc_{ij}(t) = \alpha(t).exp^{-\frac{||w_{ij}(t) - w_c||^2}{2\sigma(t)^2}},$$

   where learning rate function $\alpha(t)$ is a decreasing function of time and $\sigma(t)$ is the neighborhood radius.
5) Repeat steps (2) - (4) by a predetermined number of iterations or until the convergence criterion is satisfied.

In practice, the SOM training process usually consists of two phases: coarse training, during which the topographic order of the SOM is formed, and fine training, for obtaining a more accurate final state with the same total number of training steps as original training procedure.

The trained SOM preserves the topological properties of the input space, and therefore can be used as a data analytics tool to visualize and analyze the high-dimensional data. Moreover, SOM has the ability to generalize data from the training set. Characteristics of each new input can be derived by identifying its BMU and quantization error.

C. *Data Analytics on Traffic Flows*

While supervised machine learning-based approaches have found success in botnet detection applications [3][12], in this work, we explore the utility of employing SOMs, an unsupervised learning approach, to analyze unknown / botnet behaviours as the emerging novel threats. Moreover, though SOM's capabilities have been proven in malicious detection applications [14][15][23], it was utilized mostly as semi-supervised approach. In this research, we employ SOMs in an unsupervised manner to enable the approach to suite better to unidentified threats. Hence, we apply only one layer of SOM for data projection with minimum labelling information.

Botnet masters are employing more and more sophisticated techniques to hide botnet's fingerprints. This results in the botnet traffic becoming more and more similar to legitimate traffic, making the identification established in previous works blurry. Hence, to shed light into this phenomena and to analyze it, we train our SOMs using three different schemes based on the chosen training data:

(i) use both known normal / legitimate and known malicious traffic for training purposes, as done in the previous supervised learning approaches [14][15];

(ii) use only normal / legitimate traffic for training purposes as done in the previous unsupervised learning (anomaly detection) approaches [13];

(iii) use only malicious (botnet / CC) traffic for training purposes as done in some of the one-class classifier approaches [24].

In all schemes, only the well-identified flows, for which the ground-truth is known, are used for training. In doing so, we are able to indentify with certainty the input data for constructing the SOMs. Then we examine the capability of the trained SOMs in separating Normal, Botnet, and CC traffic in the testing portions containing labelled flows before using them to assess the unseen data for which the ground-truth is unknown. This not only represents the real-life security conditions for us but also sheds light into understanding the performance gains / losses under different types / amounts of labelling information, i.e. ground-truth. For example, using honeypots are usually for collecting only malicious data. On the other hand, in idealistic cases of networks where there are no attacks, the data collected contains only legitimate traffic. Moreover, even when a threat is discovered in the collected traffic, we are generally not able to fully identify the extension of it and label the data for training.

We employ three different trained SOMs (based on the aforementioned training schemes) for exploring the unknown / unlabelled (Background) traffic present in the aforementioned data sets. By analyzing the distribution of the background traffic on the trained SOMs, we intend to investigate the ability of the different training schemes on inspecting / analyzing unknown traffic for different attack and normal (legitimate) behaviours. This is the basic step toward an unsupervised system for automatically detecting anomalous behaviours in everyday traffic. It is noteworthy that our aim is to help the human expert to analyze the unknown traffic, but not to guarantee what is inside the unknown traffic. In our system the final decision will be made by the human expert.

IV. Evaluations and Results

A. *Parameters and Performance metrics*

Table I shows the distribution of classes in each data set. Since there are too few CC flows in Rbot capture, the malicious flows in this set are considered to be the sum of botnet / CC. In each set, 40% of the labelled data is used for training and the remaining 60% is used for testing. Given that the nature of SOM is based on distances between data vectors and nodes, traffic features are normalized with zero means and unit variance before they are used for training.

TABLE I
Data specifications

Data Set		**Number of Flows**			
		Normal	**CC**	**Botnet**	**Background**
CTU13	**8 - Murlo**	72822	1074	5053	2875281
	9 - Neris	43340	5099	179880	2525565
	10 - Rbot	15874	37	106375	1187592
	13 - Virut	31939	1202	38791	1853217
ISOT		**Normal**	**Storm**	**Waledac**	
		212203	18721	33598	

The model is built based on SOMToolbox from Aalto University, Finland, which is developed and recommended by the authors of SOM [25][26]. Learning parameters of SOM are summarized in Table II. With training scheme using both legitimate and botnet traffic, the map size is 30×30; otherwise the parameter is set to 20×20. These map sizes are determined empirically. SOMs are trained by a two-phase batch-training process, including coarse training and fine tuning, over 500 iterations per phase. The neighborhood radius is decreased linearly from the initial to final value. Finally, for each training scheme, a threshold α is applied for determining a set of important map units, which account for at least α percent of training samples, for each training class. The sets of important units are then used to label unseen flows as Normal, Botnet, or Anomaly. The threshold is set to 99%, 92%, and 90%, when both normal and botnet traffic, or just normal traffic, or just botnet traffic is used for training, respectively.

TABLE II
SOM training parameters

Parameter	**Value**
Map size	20×20 or 30×30
Lattice	Hexagonal
# of iterations	1000
Training neighborhood radius	8 to 2
Neighbornood function	Gaussian

The performance of our approach is quantified by True positive rate (TPR) per class to overcome the unbalanced nature of the data sets. TPR for each class is calculated as $TPR = TP/(TP + FN)$, where TP (FN), True Positive (False Negative), denotes the number of test instances of the class correctly (incorrectly) identified.

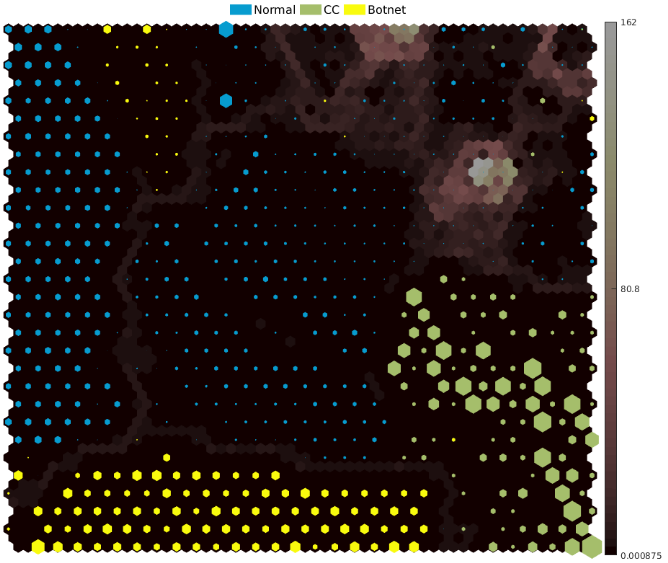

Fig. 1. Hit histogram of the SOM trained using scheme (i) on capture 8 of CTU13 data set. The background color denotes SOM Umatrix, where the color bar on the right shows the distance range.

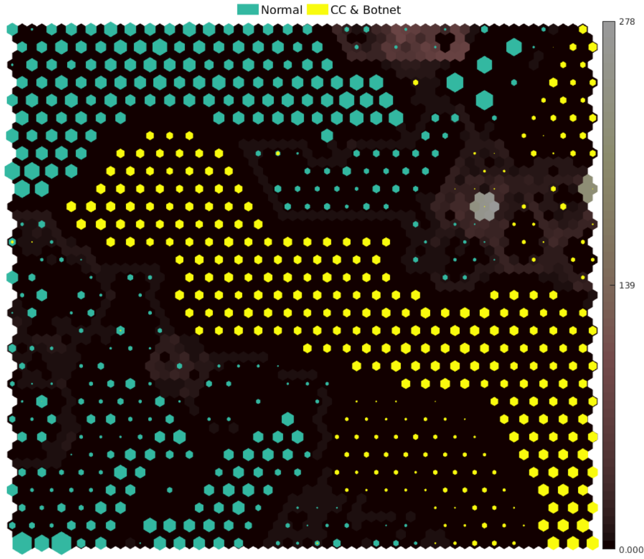

Fig. 2. Hit histogram of the SOM trained using scheme (i) on the capture 10 of CTU13 data set.

B. Results

Table III presents the performance of the SOM training schemes, which is obtained on the test partitions of the data sets. As expected, SOM training scheme using both normal and botnet traffic gives the highest results. This scheme achieves high performance with a clear separation between non-overlapping groups of BMUs of traffic classes on the trained SOM. For example, in Figures 1 and 2, which visualize data distribution of different traffic classes in capture 8 and capture 10 on the SOMs, it is clear that the different classes are either separated by lighter area in SOM Umatrix[1], which indicates large inter-node distances, or empty nodes. Moreover, in all cases, most of the incorrectly classified botnet (CC) traffic is still labelled as CC (botnet). This supports our hypothesis on the ability of the SOM in separating malicious traffic from normal traffic. On ISOT data set, if we consider there are only 2 classes (legitimate and botnet), the TPRs are 98.67% and 98.29% for legitimate and botnet traffic, respectively. The figures are comparable to the results in [11], where the detection rates of REPTree classifier with reduced subset were 97.9% and 98.1%. It is noteworthy that our approach does not use any labels for SOM training, while in [11], a supervised learning approach was employed which requires labels for training purposes. Moreover, the high performance of the approach on unseen test sets shows that it successfully avoided overfitting problem.

Among the two remaining training schemes, the results on CTU13 data sets are generally better with the scheme using Normal data only. Using 92% threshold to assign Normal label to map units, TPRs of bot flows are in the range from 80% to 95%. On the other hand, the training scheme using only Botnet traffic observes poor performance on captures 8, 9 and 13 of CTU13 data set. Using 90% threshold, only 62%, 5% and 8% of Legitimate flows are correctly classified. However, the results are fairly good on Rbot trace (capture 10) with all training schemes, suggesting that Rbot botnet is considerably easier to detect.

On ISOT data set, the trend is reversed, where SOM training by only botnet flows gives far better result than SOM trained by normal data only. However, considering that ISOT data set is a combination of a legitimate / normal data set provided by LBL and malicious data captured using Honeypots [11], the results are based on data captured at different locations under (potentially) different topologies and conditions. This might be the reason why the trend is reversed. On CTU13 data sets, normal and botnet traffic were captured on the same network at the same time, and our results also indicate this condition.

One other interesting observation from the experiments is that CC flows and Botnet flows in the CTU13 data sets are relatively different, Figure 1. This may come from the essence of these two traffic types. While Botnet flows represent attacks and malicious activities, CC traffic is for maintaining the botnet and issuing attack orders. This seems to cause the CC flows to be more similar to normal data than Botnet flows. When only normal data is used for training, CC flows are more likely to be misclassified as Normal than Botnet flows. In particular, while only 3%, 7% and 14% of Botnet flows in Murlo, Neris, and Virut traces are misclassified as Normal, the rates of CC flows are 99%, 87%, and 76%, Table IV. This supports the fact that Botmasters make use of typical protocols such as HTTP, P2P for concealing botnet communications. This observation suggests that independent detection strategies for Botnet and CC traffic may improve the classification performance.

[1] Umatrix is a graphic display to illustrate the degree of clustering tendency on the SOM via distances between SOM nodes [26]. Longer distances indicate less similarity between the data points.

TABLE III

CLASSIFICATION PERFORMANCE (TPRS) OF THREE SOM TRAINING SCHEMES UNDER THE TEST PARTITION

Data set	**Training scheme (i)**			**Training scheme (ii)**		**Training scheme (iii)**	
ISOT	**Legitimate**	**Storm**	**Waledac**	**Legitimate**	**Botnet**	**Legitimate**	**Botnet**
	98.67	94.19	96.96	91.94	27.68	94.97	89.87
	Legitimate	**CC**	**Botnet**				
Murlo	99.89	99.55	99.78	91.81	79.56	61.60	89.82
Neris	99.77	99.19	97.43	91.85	92.41	4.67	92.50
Rbot	99.84	98.99		91.01	95.39	100	89.56
Virut	99.75	98.30	96.71	91.84	85.44	8.04	89.90

TABLE IV

PERCENTAGE OF BOTNET AND CC TRAFFIC MISCLASSIFIED AS NORMAL BY SOMS TRAINED USING SCHEME (II)

Data set	**Misclassified percentage**	
	Botnet	CC
Murlo	3.47	99.55
Neris	6.69	87.25
Virut	13.56	75.63

In the second set of experiments, we analyze the distribution of Background (unlabelled / unknown) data in CTU13 on trained SOMs from the first set of experiments, Table V. Based on the promising results obtained by using both Legitimate and Botnet flows for training, we employed the SOM trained using scheme (i) to analyze the ”background” data portion of the CTU13 data sets. Note that there is no ground-truth provided by CTU for this portion of the data sets. Our results show that most of the Background flows are Legitimate (61%-79%, depending on the CTU13 data set analyzed). However, the rest of the Background traffic flows seem to be very different (confirmed by the BMU quantization error) from both Legitimate and Botnet/CC. So we suggest to label most of these as anomalies for further investigation. Our SOM based data analytics system labels only a small fraction as botnet/CC. Manually inspecting the background flows labelled as Anomaly, we found that many of them have unfamiliar protocols that were not seen in the training data, for example ARP, RTCP, RTP, and IGMP.

SOMs trained by only Normal data (scheme (ii)) show very similar Background data distributions to scheme (i). On average, 86% of the Background flows labelled as Normal by the SOMs trained using scheme (ii) are also labelled as Normal by the SOMs trained using scheme (i). On the other hand, SOMs trained by only Botnet/CC data (scheme (iii)) label most of the Background flows as Botnet.

To further investigate the Background traffic, we calculate the average quantization error for each identified class (Normal, Botnet, Anomaly) of the Background traffic. Quantization error of each data instance is defined as the distance to its BMU, hence quantifies the differentness between the instance and the SOM [6]. The average quantization errors of flows labelled as Normal is 0.69 (sd 1.90), while the figure of flows labelled as Botnet is 1.13 (sd 1.33). These low quantization errors demonstrate that SOMs trained using scheme (i) label the Background traffic as Normal and Botnet with high confidence, considering that they are calculated over 47 features, which results in overall average quantization error of 1.80 (sd 5.31) and 99% quantile of 44.25. On the other hand, for the flows labelled as Anomaly, the average quantization error is 4.32 (sd 8.40). This higher value confirms our observation that anomaly traffic contains very different behaviours / patterns that were not present in the training data. This is further confirmed by our manual analysis of these flows and the different protocols we identified as a result of this analysis. Similarly, SOMs trained using scheme (ii) give average quantization errors of 1.18 (sd 2.53) and 5.89 (sd 7.16) for flows classified as Normal and Anomaly. On the other hand, training scheme (iii) produces SOMs with much higher quantization errors when applied on the Background traffic. On average, the Background flows are classified as Botnet and Not Botnet with quantization error values of 14 (sd 5.52) and 22 (sd 22.16), respectively. These very high error values indicate that SOMs trained using scheme (iii) are not suitable for Background / unknown data analysis.

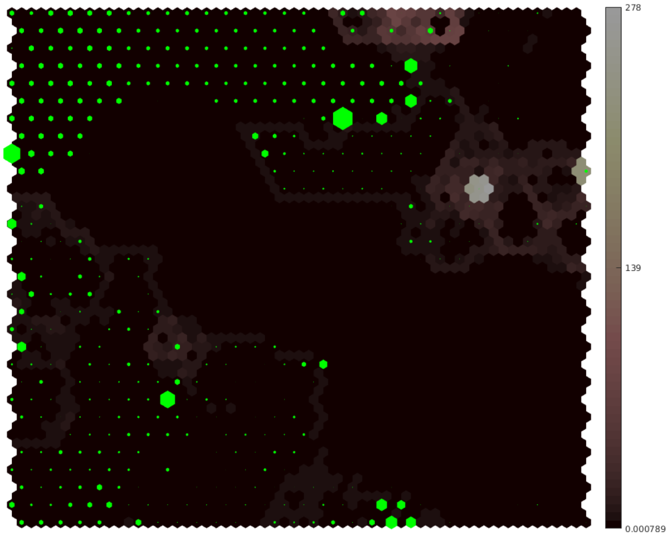

Fig. 3. Hit histogram of Background flows on the SOM trained using scheme (i) on capture 10 of CTU13 data set.

V. CONCLUSION

Our main objective in this work w to investigate the capability of SOMs as an unsupervised data analytics system for analyzing unknown / unlabelled traffic. Using three

TABLE V
DISTRIBUTION OF BACKGROUND TRAFFIC FLOWS ON THE TEST PARTITION FOR THE THREE SOM TRAINING SCHEMES

Data set	**Training scheme (i)**			**Training scheme (ii)**	**Training scheme (iii)**
	% of Normal Flows	**% of Anomalous Flows**	**% of Malicious (Botnet) Flows**	**% of Normal Flows**	**% of Malicious (Botnet) Flows**
Murlo	61.86	34.74	3.40	71.89	60.76
Neris	68.56	29.57	1.85	65.99	92.34
Rbot	78.30	18.88	1.20	76.81	55.08
Virut	62.29	35.14	2.41	62.64	81.72

different SOM training schemes, we analyzed and evaluated the capabilities of this SOM based approach on publicly available data sets of modern botnets. The obtained results are comparable to that of previous supervised machine learning-based approaches, even though our approach is based on unsupervised learning paradigm. Detection rates of Botnet and Normal classes are up to 99.78% and 99.89% with training scheme using both classes. Moreover the technique showed its potential for building a strong data analytics system for unknown traffic analysis.

Our data analytics results on unknown traffic suggest that when a complete set of training data is not available, SOMs can be trained on normal data only and still achieve a high performance, given that the data is diverse enough to cover most part of the legitimate traffic. However, for higher accuracies, data analytics systems trained on both malicious and normal behaviours should be preferred.

Future work will investigate the ability of filters based on both SOM hit counts and quantization errors in reducing the noise in data and increasing the accuracy. Moreover, self-growing SOMs could also be employed to automate the process of tuning SOM training parameters. Finally, performance of an SOM-based data analytics system can be studied against other data sets, to examine its potential of detecting other types of network attacks and malicious activities.

ACKNOWLEDGMENT

We gratefully acknowledge the supports from the Natural Sciences and Engineering Council of Canada and the Canadian Safety and Security Program.

REFERENCES

[1] Kaspersky lab, "Kaspersky DDoS Intelligence Report for Q2 2016," August 2016. [Online]. Available: https://securelist.com/analysis/quarterly-malware-reports/75513/kaspersky-ddos-intelligence-report-for-q2-2016/

[2] RSA Security LLC, "Cybercrime 2015: An inside look at the changing threat landscape," EMC, Tech. Rep., Apr. 2015.

[3] S. S. Silva, R. M. Silva, R. C. Pinto, and R. M. Salles, "Botnets: A survey," *Computer Networks*, vol. 57, no. 2, pp. 378 – 403, 2013.

[4] E. J. Kartaltepe, J. A. Morales, S. Xu, and R. Sandhu, "Social Network-Based Botnet Command-and-Control: Emerging Threats and Countermeasures." Springer Berlin Heidelberg, 2010, pp. 511–528.

[5] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[6] T. Kohonen, *Self-Organizing Maps*, 3rd ed., ser. Springer Series in Information Sciences. Springer Berlin Heidelberg, 2001, vol. 30.

[7] "Snort." [Online]. Available: https://www.snort.org/

[8] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "Bothunter: Detecting malware infection through ids-driven dialog correlation," in *Proc. 16th USENIX Security Symposium*, 2007.

[9] P. Wurzinger, L. Bilge, T. Holz, J. Goebel, C. Kruegel, and E. Kirda, "Automatically generating models for botnet detection," in *Proc. 14th European Conference on Research in Computer Security*, 2009, pp. 232–249.

[10] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Proc. 17th USENIX Security Symposium*, 2008, pp. 139–154.

[11] D. Zhao, I. Traore, B. Sayed, W. Lu, S. Saad, A. Ghorbani, and D. Garant, "Botnet detection based on traffic behavior analysis and flow intervals," *Computers & Security*, vol. 39, pp. 2–16, 2013.

[12] F. Haddadi and A. N. Zincir-Heywood, "A Closer Look at the HTTP and P2P Based Botnets from a Detector's Perspective," in *Proc. 8th International Symposium on Foundations and Practice of Security (FPS 2015).* Springer International Publishing, 2015, pp. 212–228.

[13] K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters," in *Proc. Twenty-eighth Australasian conference on Computer Science*, vol. 38, 2005, pp. 333–342.

[14] H. Gunes Kayacik, A. Nur Zincir-Heywood, and M. I. Heywood, "A hierarchical SOM-based intrusion detection system," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 4, pp. 439–451, 2007.

[15] D. Ippoliti and X. Zhou, "An adaptive growing hierarchical self organizing map for network intrusion detection," in *Proc. 19th International Conference on Computer Communications and Networks*, 2010, pp. 1–7.

[16] J. McHugh, "Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 262–294, 2000.

[17] N. Brownlee, C. Mills, and G. Ruth, "RFC 2722 - Traffic flow measurement: Architecture," IETF, Tech. Rep., Oct. 1999. [Online]. Available: https://tools.ietf.org/html/rfc2722

[18] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Computers & Security*, vol. 45, pp. 100–123, 2014.

[19] Lawrence Berkeley National Laboratory and ICSI, "LBNL enterprise trace repository," 2005. [Online]. Available: http://www.icir.org/enterprise-tracing

[20] "The Honeynet Project. French Chapter." [Online]. Available: http://www. honeynet.org/chapters/france

[21] "The CTU-13 Dataset. A Labeled Dataset with Botnet, Normal and Background traffic." [Online]. Available: http://mcfp.weebly.com/the-ctu-13-dataset-a-labeled-dataset-with-botnet-normal-and-background-traffic.html

[22] "Tranalyzer." [Online]. Available: http://tranalyzer.com/

[23] S. Sarasamma, Q. Zhu, and J. Huff, "Hierarchical Kohonenen Net for Anomaly Detection in Network Security," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 2, pp. 302–312, Apr 2005.

[24] P. Winter, E. Hermann, and M. Zeilinger, "Inductive Intrusion Detection in Flow-Based Network Data Using One-Class Support Vector Machines," in *Proc. 4th IFIP International Conference on New Technologies, Mobility and Security*, Feb 2011, pp. 1–5.

[25] J. P. Esa Alhoniemi, Johan Himberg and J. Vesanto, "SOM Toolbox." [Online]. Available: http://www.cis.hut.fi/somtoolbox/

[26] T. Kohonen, *MATLAB Implementations and Applications of the Self-Organizing Map*, 2014. [Online]. Available: http://docs.unigrafia.fi/publications/kohonen_teuvo/