Data Analytics for Modeling and Visualizing Attack Behaviors: A Case Study on SSH Brute Force Attacks

Chengchao Yao
Faculty of Computer Science
Dalhousie University
Halifax, NS, Canada
Email: ch250407@dal.ca

Xiao Luo
Purdue School of Engineering and Technology
Indiana University-Purdue University Indianapolis
Indiana, USA 46202-5160
Email: luo25@iupui.edu

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

***Abstract*—In this research, we explore a data analytics based approach for modeling and visualizing attack behaviors. To this end, we employ Self-Organizing Map and Association Rule Mining algorithms to analyze and interpret the behaviors of SSH brute force attacks and SSH normal traffic as a case study. The experimental results based on four different data sets show that the patterns extracted and interpreted from the SSH brute force attack data sets are similar to each other but significantly different from those extracted from the SSH normal traffic data sets. The analysis of the attack traffic provides insight into behavior modeling for brute force SSH attacks. Furthermore, this sheds light into how data analytics could help in modeling and visualizing attack behaviors in general in terms of data acquisition and feature extraction.**

I. Introduction

Traditionally, network traffic analysis is performed for the purpose of performance, security, or general network operations and management. For the purpose of security, intrusion/malware detection and prevention is one of its application domains. In general, these security systems can be classified into two classes: signature based detection and anomaly based detection. Machine learning algorithms have been investigated for both signature and anomaly based detection. However, Sommer et al. [1] suggested to strengthen future research on anomaly detection is to provide insight into the decision process of the anomaly detection systems [1]. In this paper, our objective is to use data analytics, specifically machine learning algorithms to provide insight into analyzing and modeling attack behaviors. In doing so, our aim is to develop support systems for human experts of the network operations and management teams to visualize, analyze and model new threats and attacks to come.

There are two major components of the proposed system. First, we employ Self-Organizing Map (SOM) which is an unsupervised learning algorithm to investigate the distributions of the traffic flows on a topographical two-dimensional map. Then, Association Rule Mining (ARM) is employed to provide insight into the clusters on the SOM by generating association rules between the features of the traffic flows. Finally, we present trajectories on top of the SOM clusters based on the time stamps of the traffic flows. This provides us the means to visualize the behaviors of the network traffic both in spatial and temporal presentations. It is worth to note that the attack traffic flows and the normal traffic flows are fed into the system separately. This ensures that the attack traffic flows and normal traffic flows can generate their own maps, rules, and trajectories specifically and separately, creating ‘fingerprints’ for the visualized behaviour. This is similar to the ‘fingerprints’ owned by individuals. Different attacks’ behaviors can be recognized and visualized by the means of the ‘fingerprints’. In this research, as a case study, we specifically analyze the SSH brute force attacks behaviors and normal SSH traffic behaviors by using the proposed system. In other words, we generated ‘fingerprints’ for each one of these behaviour by using our proposed system. The SSH Brute force attack is one of the most prevalent attacks in computer networks. The attacker’s objective is to gain SSH access to target machine by trying many passwords or passphrases. Based on the experimental results, the proposed system could model the differences between the attack and normal behaviors of the same application, namely SSH. One valuable finding is that the patterns (fingerprints) generated from the normal traffic are totally different from the patterns (fingerprints) generated by the SSH brute force attacks traffic. The contributions of this work include: a data analytics based system to provide easily interpretable insight into the traffic flows, and a mechanism of generating ‘fingerprints’ for different behaviors in order to support further analysis for malicious behavior detection.

The rest of the paper is organized as follows. The related work in the literature is summarized in Section II. The methodologies of the proposed system are detailed in Section III. Section IV presents the experiments and results. Finally, conclusions are drawn and the future work is discussed in Section V.

II. Literature Review

Flow based or packet based data analytics algorithms such as unsupervised and supervised learning algorithms have been investigated for intrusion detection system (IDS) and anomaly

This is the author's manuscript of the article published in final edited form as:
Yao, C., Luo, X., & Zincir-Heywood, A. N. (2017). Data analytics for modeling and visualizing attack behaviors: A case study on SSH brute force attacks. In 2017 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1–8). https://doi.org/10.1109/SSCI.2017.8280913

detection systems (ADS) for years. Usually the performance of the designed algorithms are evaluated using detection rate and false alarm (positive/negative) rate. Many researches [2] [3] [4] [5] [6] proposed and explored different mechanisms to improve the detection rate and meanwhile reduce the false positive rate on detecting different types of attack behaviors. However, detection and false positive rates do not provide security analysts any insight to understand the behaviors of the attacks or to support decision making in real time. We believe that tools and techniques that will assist to visualize, model and understand different attack behaviors are necessary to support security analysts. To this end, some researchers have employed machine learning and visualization algorithms to model and understand different aspects of attack behaviors.

Atkison et al [7] used information retrieval techniques to first organize the Telnet traffic by packets and sessions, then the sessions with different attack or normal labels were loaded into a database. Given an attack session, the similarity scores between the given attack and the known attacks within the database were then calculated. To visualize and understand behavior similarity between the given attack and the known attacks, the similarity score lists were loaded into a visualization system. This approach was demonstrated on three types of attacks and on a small corpora constructed by the authors.

In [8], Intarasothonchun and Srimuang used the best-first selection and the greedy stepwise algorithms to find the relevant features to present different attack behaviors based on the KDD'99 data set. Then, different classification algorithms, namely Weighted extreme learning model and Support Vector Machine (SVM) + Genetic Algorithms (GA) were employed. Even though their results were promising, the behavioral patterns of attacks in terms of the spatial and temporal aspects were not modeled or presented.

Jyothi et. al [9] proposed a system that combined hardware statistics, network statistics and application statistics to present the behaviors of the Distributed Denial of Service (DDoS) attacks and then applied different learning algorithms such as K-means and SVM to detect the DDoS. Similar to the other works, the behavior patterns and the rationale behind the patterns were not explored using the learning algorithms.

In summary, to the best of our knowledge, previous research employed machine learning and visualization techniques to analyze attacks and anomalies. However, in this research, our objective is to combine machine learning and visualization to model attack patterns and trajectories by generating fingerprints of different normal and attack behaviours. In doing so, we aim to support security analysts in their decision making process.

III. Methodology

In this work, we employ two unsupervised learning algorithms - Self-Organizing Map (SOM) and Association Rule Mining (ARM) to model and visualize the network flow behaviors of a host. The reason we employ unsupervised learning is that finding labeled traffic to train supervised learning algorithms is very costly in practice. Such an activity implies that somehow a human expert analyzes interesting traffic and labels it as "attack" or something else. This can become a very challenging activity and can only scale to small data sets. However, as the data flow on the Internet grows, it seems more practical to use unsupervised learning which does not require labels during the training phase.

A. Overview of the Proposed System

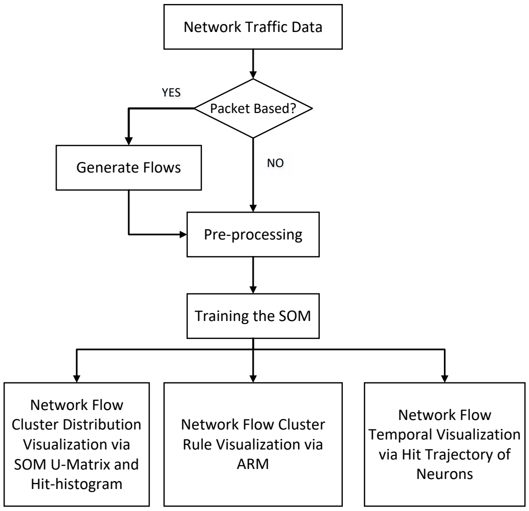
Network Traffic Data
Packet Based?
YES
NO
Generate Flows
Pre-processing
Training the SOM
Network Flow Cluster Distribution Visualization via SOM U-Matrix and Hit-histogram
Network Flow Cluster Rule Visualization via ARM
Network Flow Temporal Visualization via Hit Trajectory of Neurons

Fig. 1. Overview of the proposed system

Figure 1 shows the overview of the system. The proposed system is designed to work with network traffic flows. In this research, we used publicly available traffic and SSH Brute force attack traffic generated by Ncrack [10]. Tranalyzer [11] [12] is employed to export (generate) flows from the traffic based on the captured network packets. Some pre-processing steps that include feature reconstruction and data normalization are employed before the network flows are used to train the SOM. The pre-processing steps include:

- Feature reconstruction: Tranalyzer constructs 93 features per flow [12]. The specific features that were used in this proposed system are detailed in the section 4.
- Data normalization: All the features are normalized into the range of 0 to 1 to avoid learning biases on any of the features. The normalization function we used in this research is linear normalization.

After pre-processing, all network flows are used to train the SOM to identify patterns in the traffic. On top of the trained SOM map, we employ three methods to visualize and understand the network flow behaviors.

- Visualization of the distribution of the network flow clusters based on the techniques provided by the SOM: To this end, SOM U-Matrixes are used as the two dimensional topographic graphs.
- Visualization of regularities between flow features and clusters by using ARM: The ARM algorithm is not feasible if a feature value is not discrete. To transfer the

continuous values to discrete values, we generate a set of ranges based on the distribution of the values. A value in a range is represented by the assigned label or index to the range. A port number could be any value from 0 to 65536. So, based on the distribution of source port numbers in the input network flow data set, a set of ranges: 0 - 500, 500 - 2000, 2000 - 5000, 5000 - 65536 are generated. For example, if the source port number is 22, it is assigned to range 1.

- Visualization of the temporal behaviors of the network traffic using flow trajectories on the trained SOM: Network flows are a set of sequential packets between two hosts and their specific ports over a specific period of time. Thus, it is more likely that a combination of network flows may reflect a pattern for a certain attack activity. In order to visualize such network behaviors within a time period, hit trajectories on the trained SOM is proposed here. This could assist the security analysts to visualize the sequential flows, so that the patterns of network behaviors can be further analyzed.

The details of the SOM and ARM learning algorithms are provided in the following subsections.

B. Self-Organizing Map and Data Visualization

A basic SOM consists of M neurons located on a low dimensional grid (usually 1 or 2 dimensional) [13]. The algorithm responsible for the formation of the SOM involves three basic steps after initialization: sampling, similarity matching, and updating. These three steps are repeated until the formation of the feature map has completed. Each neuron i has a d-dimensional prototype weight vector $W_i = W_{i1}, W_{i1}, ..., W_{id}$. Given X is a d-dimensional sample data(input vector), the algorithm is summarized as follows:

- Initialization:
  Choose random values to initialize all the neuron weight vectors $W_i(0), i = 1, 2, ..., M$, where M is the total number of neurons in the map.
- Sampling:
  Draw a sample data X from the input space with a uniform probability.
- Similarity Matching:
  Find the best matching unit (BMU) or winner neuron of X, denoted here by b which is the closest neuron (map unit) to X in the criterion of minimum Euclidean distance, at time step n (n^{th} training iteration).

$$b = \arg\min_i ||X - W_i(n)||, i = 1, 2, ..., M \qquad (1)$$

- Updating:
  Adjust the weight vectors of all neurons by using a update formula, so that the best matching unit (BMU) and its topological neighbors are moved closer to the input vector X in the input space.
- Continuation:
  Continue with sampling until no noticeable changes in the feature map are observed or the pre-defined maximum number of iterations is reached.

The most commonly used visualization techniques of SOM are the U-Matrix and Hit histogram. The U-matrix [13] holds all distances between neurons and their immediate neighbor neurons. It gives a direct visualization of the number of clusters and their distribution on a two dimensional space. The hit histogram of the input data set on the trained map provides a visualization that details the distribution of input data across the clusters. Each input data instance in the data set can be projected (hit) to the closest neuron on a trained SOM map. The hit histogram is constructed by counting the number of hits each neuron receives from the input data set. On the hit histogram, the larger the shaded area is on the neuron, the more hits the neuron receives.

C. Association Rule Mining and Identifying Rules Automatically

Traditionally, ARM is used to find items that occur simultaneously and often in database transactions [14]. Given a set of items $I = I_1, I_2, ..., I_m$ and a set of database transactions $T = t_1, t_2, ..., t_n$ where $t_i = I_{i1}, I_{i2}, ..., I_{im}$ and $I_{ij} \in I (i = 1, ..., n, j = 1, ..., m)$ The standard definition of ARM is to find a set of rules expressed in the form of 2:

$$X \rightarrow Y \qquad (2)$$

where $X, Y \subseteq$ I are sets of items called itemsets, and $X \cap Y = \emptyset$.

The significance of the rules that are identified through the learning algorithm are measured in terms of their support rates (S) and confidence rates (C), as shown in equations 3 and 4.

$$S(X \rightarrow Y) = \frac{|X \cup Y|}{|T|} \qquad (3)$$

$$C(X \rightarrow Y) = \frac{|X \cup Y|}{|X|} \qquad (4)$$

where $|\cdot|$ denotes number of items in a itemset.

Support rate implies how often the rule is applicable to a given transaction set. Support rate measures the popularity of the rule. Confidence rate implies how frequently items in Y appear in transactions that contain X. Confidence rate measures the reliability of the inference made by a rule. In the data mining field, thresholds are set for support rate and confidence rate to find the set of rules are are most popular and reliable.

In this work, we use ARM to identify rules automatically in order to facilitate the understanding of network behaviors and features that are hit to one or more neurons on the SOM map. The most frequent and reliable rules are identified by using high thresholds of the support rate and the confidence rate.

IV. Evaluations and Results

In order to evaluate the proposed system, two different SSH brute force attack traffic data sets and two normal traffic data sets are used. One of the SSH brute force attack data sets is generated by Ncrack [10]. The other SSH bruce force attack data set is extracted from the public benchmarking data set: ISCX [15]. The ISCX data set contains traffic that are labelled as either attack or normal. The two normal network traffic data sets are extracted from: ISCX and DARPA (week-1 and week-3), which are all labelled as normal [16]. These data sets are summarized in Table I.

TABLE I
Summary of Data Sets

Data Set Type	Data Set Name	# of Flows
Attack	Ncrack SSH Brute Force	57992
	ISCX SSH Brute Force	10154
Normal	ISCX SSH Normal	4265
	DARPA SSH Normal	1384

A. Ncrack

Ncrack is a network cracking tool that is designed to be a fast and flexible network authentication cracker and can perform brute force attacks across the network [10].

In this research, we used Ncrack to generate all the brute force attack traffic on our network testbed. Specifically, two linux machines were set up on the network. One is the attacker machine, the other is the targeted victim. A password list that contains 110,000 passwords was used to perform a SSH brute force password cracking attack to the victim machine. Wireshark [17] was employed to capture the network packets on the victim machine. Tranalyzer2 [11] was employed to export the flows from the captured traffic. The flow features that are extracted from the Tranalyzer2 are shown in Table II.

After going through the pre-processing steps detailed in section III-A, all the network flows were fed into the SOM to train the map. The training parameters of the SOM are presented in Table III. The U-Matrix and hit histogram of Figure 2 show the clusters - modeling the captured data - on the trained SOM. Four clusters that are manually circled in green lines can be clearly visualized based on the combination of U-Matrix and hit histogram. They are summarized as four different clusters in table IV: top left (TL), top right (TR), bottom right (BR) and bottom left (BL) based on the location of the clusters on the map. These incoming and outgoing traffic generated by Ncrack are clearly separated on the trained SOM map. It indicates that different behaviors can be identified on the learned model given by the SOM.

TABLE II
Features of the network flow

Feature Name	Tranalyzer2 Feature Name
Duration	Duration
TCP Flags	tcpAggrFlags
Source Port	SrcPort
Destination Port	DstPort
Packets	numPktsSnt
Octets	numBytesSnt

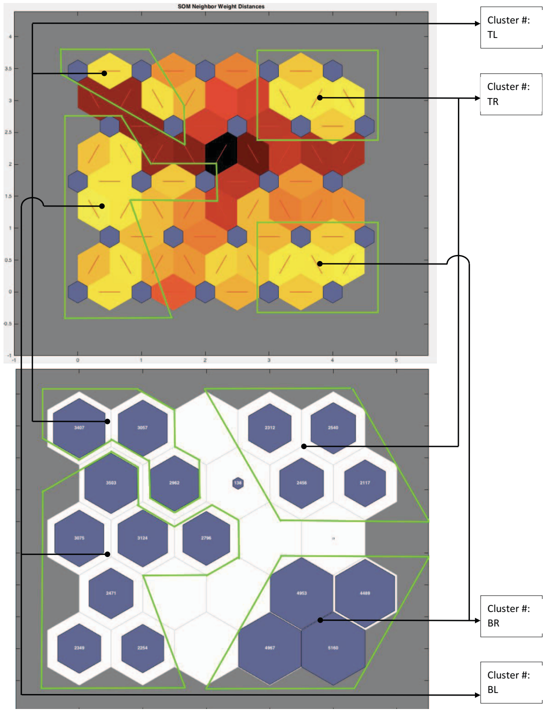

Fig. 2. U-matrix, Hit histogram of SOM - Ncrack SSH Brute Force

TABLE III
SOM Training Parameters

Parameter	Value
Map Size	5 X 5
Initial neighborhood size	3
Layer topology function	Hexagon
Neuron distance kernel function	Link distance
Training Epochs	500,000

In order to to automate the process of finding rules to identify the traffic patterns of each cluster, association rule mining (ARM) is used on the four clusters. Since the values of all features summarized in Table II are not discrete, ranges are generated to transfer a continuous value into a discrete value. By applying support rate threshold 0.1 and confidence rate threshold 1, we identified a set of highly reliable rules for each cluster, Table V. Based on these rules, we identified that flows with larger number of packets and longer duration are within the two bottom clusters (BR and BL). While, flows with smaller number of packets and shorter duration are within the top two clusters (TL and TR).

TABLE IV
OVERVIEW OF CLUSTERS - NCRACK SSH BRUTE FORCE

Cluster	Neuron Indexes	Direction
TL	17, 21, 22	Incoming
TR	19, 20, 24, 25	Outgoing
BR	04, 05, 09, 10	Outgoing
BL	01, 02, 06, 11, 12, 13, 16	Incoming

TABLE V
ASSOCIATION RULE MINING RESULTS - NCRACK SSH BRUTE FORCE

Cluster	Sample Rules
TL	Octets = 0, Packets = (2-4), TCPflags = (15-20), DstPort = (0-500), Duration = (0-2) → TL (S:0.162, C:1.0)
TR	Octets = 0, Packets = (2-4), TCPflags = (15-20), SrcPort = (0-500), Duration = (0-2) → TR (S:0.162, C:1.0)
BL	Packets = (14-16), TCPflags = (25-30), DstPort = (0-500) → BL (S:0.210, C:1.0)
	Octets = 0, Packets = (12-14), TCPflags = (15-20), DstPort = (0-500), Duration = (0-2) → BL (S:0.337, C:1.0)
	DstPort = (0-500), Duration = (12-14) → BL (S:0.194, C:1.0)
BR	Octets = (2500-3000), Packets = (20+), TCPflags = (25-30), SrcPort = (0-500) → BR (S:0.337, C:1.0)
	Octets = (2500-3000), Packets = (20+), TCPflags = (25-30), SrcPort = (0-500), Duration = (12-14) → BR (S:0.194, C:1.0)
	Packets = (20+), TCPflags = (25-30), Duration = (10-12) → BR (S:0.133, C:1.0)

One network flow is not enough to reflect the attacker's behaviors over time. Temporal visualization of network flows changing along with the time could also provide a valuable insight to the attacker's behaviors. Figure 3 shows the incoming and outgoing network traffic by using the time stamps.

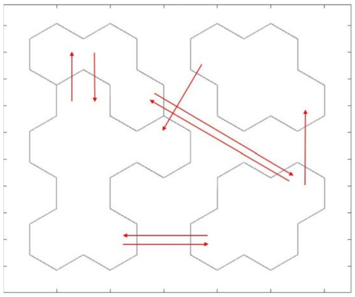

Fig. 3. Flow trajectory on SOM - Ncrack SSH Brute Force

If the trajectory is presented by using a sequence of clusters, it will be as $BL \rightarrow BR \rightarrow BL \rightarrow BR \rightarrow TL \rightarrow BR \rightarrow TR \rightarrow BL \rightarrow BR \rightarrow BL \rightarrow TL \rightarrow BL \rightarrow BR \rightarrow TR \rightarrow BR \rightarrow BL \rightarrow \ldots$. This shows that most of the time, the traffic flows are back and forth between the incoming and outgoing clusters on the map. However, sometimes one incoming traffic could be followed by another incoming traffic. This seems to be because of the way Ncrack runs the SSH brute force attack using multi-threads. Two or more attack actions can be initiated from different source ports subsequently, and some attack actions might consist of a higher number of packets, while some might consist of a smaller number of packets.

B. ISCX SSH Brute Force

ISCX data set simulates user behaviors which were abstracted into profiles [15]. The attack scenarios were designed to show the real-world cases of malicious behaviors. This data set consists of the seven days of network activity. Three days contain only normal traffic. The other four days contain four different attacks and normal activity. In this work, the SSH brute force attacks are extracted from this data set to use in our evaluations. We used the same features, pre-processing, normalization and training parameters as before to represent these traffic flows to the SOM.

TABLE VI
OVERVIEW OF CLUSTERS - ISCX SSH BRUTE FORCE

Cluster	Neuron Indexes	Direction
BL	01	Outgoing
TL	07, 13, 17, 18, 21, 22, 23	Outgoing
R	05, 09, 10, 15, 20,	Incoming

The U-Matrix and hit histogram in Figure 4 show the clusters modeling the data on the trained SOM. In this case, there are three clusters: bottom left (BL), top left (TL) and right (R). Based on the U-Matrix and hit histogram, the cluster distribution and the reason behind the distribution are the same as Ncrack - the incoming and outgoing traffic flows are clearly separated on the trained SOM map. Table VI summarizes the neurons of each cluster. Further investigation shows that there are two clusters for incoming flows and one for outgoing flows. After applying ARM algorithm to the traffic flows for each cluster, it is identified that the BL cluster contains all the incoming traffic that has a short duration, extremely small number of packets, and TCP flag values that are in the range (20,25). This is very different from the rest of the network traffic. The outgoing flows are all in the R cluster that has similar number of packets and the same range of TCP flag values. Table VII shows the rules with confidence rate of 1.

TABLE VII
ASSOCIATION RULE MINING RESULTS - ISCX SSH BRUTE FORCE

Cluster	Sample Rules
BL	Octets = (0-500), TCPflags = (20-25) SrcPort = (0-500) → BL (S:0.120, C:0.9)
TL	TCPflags = (25-30), SrcPort = (0-500) → TL (S:0.262, C:1.0)
	Octets = (2000-2500), SrcPort = (0-500) → TL (S:0.262, C:1.0)
	Octets = (2000-2500), TCPflags = (25-30), Duration = (2-4) → TL (S:0.262, C:1.0)
R	Octets = (1000-1500), DstPort = (0-500) → R (S:0.352, C:1.0)
	TCPflags = (25-30), DstPort = (0-500) → R (S:0.450, C:1.0)
	DstPort = (0-500) → R (S:1.0, C:1.0)

Figure 5 shows the incoming and outgoing network traffic over a period of time. The detailed sequence is as $R \rightarrow BL \rightarrow R \rightarrow TL \rightarrow R \rightarrow \ldots$. It should be noted here that the trajectories of the traffic flows are switching between

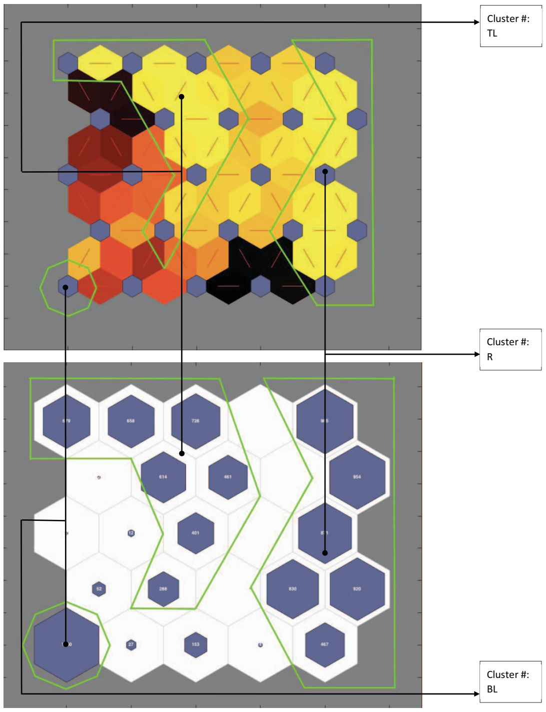

Fig. 4. U-matrix, Hit histogram of SOM - ISCX SSH Brute Force

the incoming and outgoing clusters. The incoming flows that represent the attack traffic are sometimes extremely short duration flows followed by a regular duration flow.

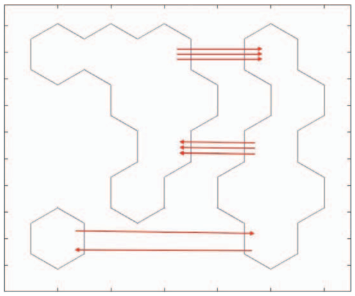

Fig. 5. Flow trajectory on SOM - ISCX SSH Brute Force

C. ISCX SSH Normal

In the ISCX data set, there are three days of traffic including only normal network activities and another four days of traffic of both attack and normal network activities. For comparison purposes, we extract all the SSH normal network flows from the seven days of normal activities.

Then, the data are pre-processed and normalized before training the SOM. Figure 6 shows the U-Matrix and hit histogram of the trained SOM. This is very different from the cluster distributions shown with four SSH brute force data sets. There is no obvious big clusters on the trained map. The incoming and outgoing traffic flows are not clearly separated into clusters. Although there are four neurons that receive more hits from the traffic flows than the other neurons, overall, they do not seem to form coherent clusters.

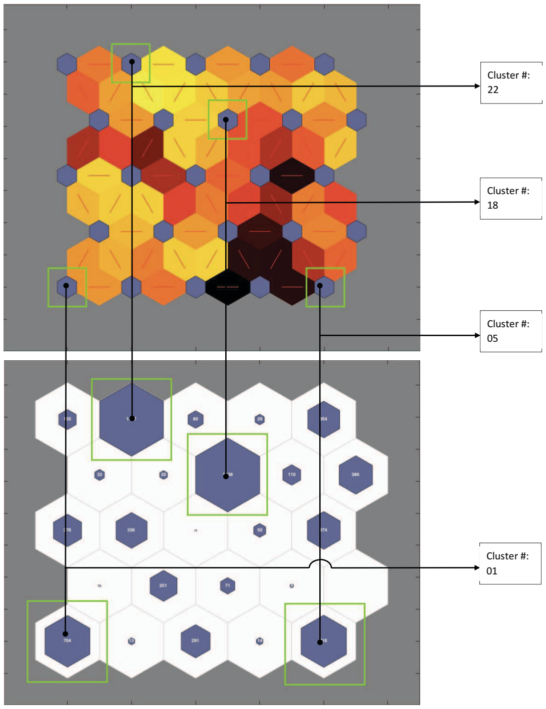

Fig. 6. U-matrix, Hit histogram of SOM - ISCX SSH Normal

TABLE VIII
OVERVIEW OF FOUR POPULAR NEURONS - ISCX SSH NORMAL

Neuron Index	Direction
01, 22	Outgoing
05, 18	Incoming

TABLE IX
ASSOCIATION RULE MINING RESULTS - ISCX SSH NORMAL

Cluster	Sample Rules
01	Octets = (0-500), TCPflags = (15-20) → 01 (S:0.1, C:1.0)

	Octets = (0-500), Packets = (4-6)
	Duration = (30+) → 01 (S:0.1, C:1.0)

	TCPflags = (15-20), Duration = (30+) → 01 (S:0.1, C:1.0)
05	Packets = (0-2), DstPort = (0-500) → 05 (S:0.1, C:1.0)
	Packets = (0-2), Duration = (30+) → 05 (S:0.1, C:1.0)

	Octets = (0-500), Packets = (0-2), TCPflags = (0-5),
	Duration = (30+) → 05 (S:0.1, C:1.0)

18	TCPflags = (25-30), DstPort = (0-500) → 18 (S:0.2, C:1.0)
	Duration = (4-6), DstPort = (0-500) → 18 (S:0.169, C:1.0)

	TCPflags = (25-30), DstPort = (0-500),
	Duration = (4-6) → 18 (S:0.166, C:1.0)

22	SrcPort = (0-500), TCPflags = (25-30) → 22 (S:0.19, C:1.0)

	Packets = (20+), TCPflags = (25-30),
	SrcPort = (0-500) → 22 (S:0.187, C:1.0)

	Octets = (5000+), SrcPort = (0-500) → 22 (S:0.165, C:1.0)

Given that, understanding and visualizing the network behavior is the objective of this work, we investigated four neurons that received the most hits on the trained SOM to understand this behavior better. We identified that among these four neurons, two are hit by incoming flows, and the other two are hit by outgoing flows. Table VIII presents the neurons and traffic directions on the SOM.

Again, ARM is used to automatically find the rules to identify the differences between the neurons. We found that some of the flows hit to one of the neurons (neuron 05) have really long duration (30+ secs), but small number of packets (0-2 packets). However, some of the flows hit to another neuron (neuron 22) have short duration (2-4 secs) but larger number of packets (20+ packets). This kind of behavior was not available on any of the SSH brute force data sets. Some of the rules for the four neurons are given in Table IX with support rates and confidence rates.

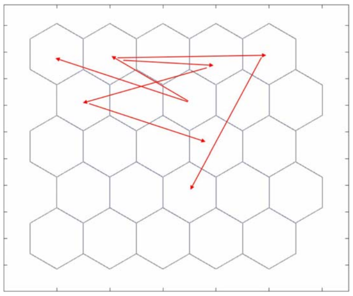

Fig. 7. Flow trajectory on SOM - ISCX SSH Normal

The further analysis of the flow trajectories generated by the incoming and outgoing normal traffic flows are shown in Figure 7. We think that these normal SSH traffic flows are generated by a mixed number of normal users using the SSH services with different objectives. The behavior patterns of these normal users are different from those of the attackers. There is no clusters of incoming and outgoing traffics that can be identified by SOM, and the trajectories of the traffic flows are not switching between the incoming and outgoing clusters. Thus, what we observe on the SOM maps of the attack behaviors are totally different than the SOM maps of the normal traffic behaviors. In short, our proposed approach with trajectories is able to create ‘fingerprints’ and a good separation of these different behaviors.

D. DARPA SSH Normal

The other normal SSH traffic data set that is used in this research is from the DARPA data set [16]. There are three weeks of network traffic data, and two of them do not contain any attacks. In this study, normal SSH network traffic flows which are using port number 22 are extracted from these normal only weeks. The original DARPA data set is not flow based, so Tranalyer2 [11] was used to export the flows based on the packets and the same steps as before are employed to train the SOM.

Figure 8 shows the U-Matrix and the hit histogram of this trained SOM. In this case, there are two clusters that are at the top right (TR) and middle left (ML) of the map. However, these two clusters do not contain most of the traffic flows within the data set. A big number of the flows actually hit the bottom part of the map. Again, this is different from the cluster distribution of the SSH brute force attack data sets. Table X summarizes the neurons within the two clusters that can be visualized on the U-Matrix and the hit histogram.

TABLE X
OVERVIEW OF TWO CLUSTERS - DARPA SSH NORMAL

Cluster	Neuron Indexes	Direction
TR	20, 24, 25	Incoming
ML	16, 17, 18	Outgoing

TABLE XI
ASSOCIATION RULE MINING RESULTS - DARPA SSH NORMAL

Cluster	Sample Rules

ML	Octets = (0), Packets = (0-2),
	SrcPort = (0-500) → ML (S:0.261, C:1.0)

	Octets = (0), SrcPort = (0-500)
	Duration = (0-2) → ML (S:0.261, C:1.0)

	Octets = (0), Packets = (0-2), TCPflags = (15-20)
	SrcPort = (0-500), Duration = (0-2) → ML (S:0.261, C:1.0)

TR	Octets = (0), Packets = (0-2),
	DstPort = (0-500) → TR (S:0.266, C:1.0)

	Octets = (0), DstPort = (0-500)
	Duration = (0-2) → TR (S:0.266, C:1.0)

	Octets = (0), Packets = (0-2), TCPflags = (15-20)
	DstPort = (0-500), Duration = (0-2) → TR (S:0.266, C:1.0)

ARM algorithm is then applied to the two clusters to automatically create the rules for the traffic flows that hit to these two clusters. Some of the rules are shown in Table XI. Both the incoming and outgoing traffic flows in these two clusters are short duration and small number of packets and the TCP flag values are within the same range.

Furthermore, the flow trajectory, as shown in Figure 9, demonstrates that the traffic flows are not switching between the identified incoming and outgoing clusters. Instead, they switch between the neurons around the map. Similar to our observations with the previous data sets, the trajectories and association rules learned are quiet different between the normal and attack behaviors. That means the patterns fingerprinted in the normal data are significantly different from those fingerprinted from the attack data.

V. CONCLUSIONS AND FUTURE WORK

In this research, we propose a data analytics based system to analyze and model the network traffic flows in order to provide insight to understand the attack behaviors. To this end, instead of building a system that aims to achieve a high detection rate and a low false alarm rate, we focus on developing a system that can analyze the network traffic flows and support decision making process for the human experts via automatic rule mining and visualization based on network traffic behaviors.

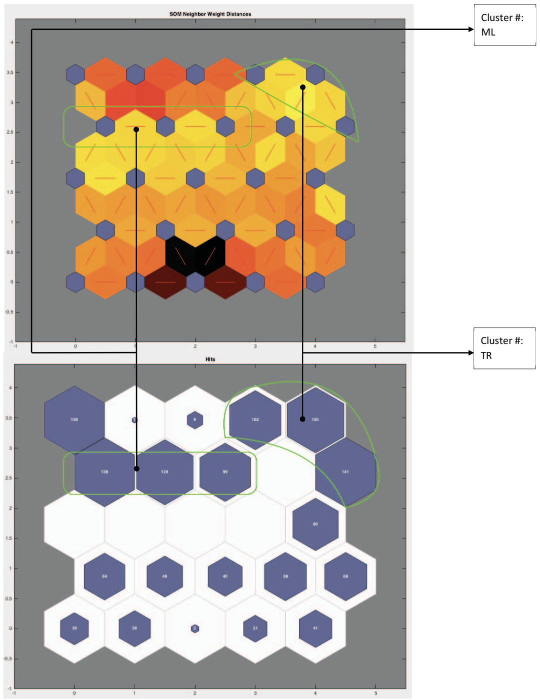

Fig. 8. U-matrix, Hit histogram of SOM - DARPA SSH Normal

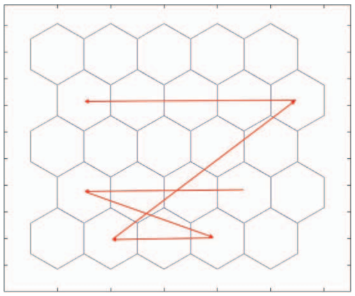

Fig. 9. Flow trajectory on SOM - Darpa SSH Normal

By making use of the proposed system, a human expert could benefit from visualizing the behavioral patterns in the network traffic and understanding the rules behind the patterns. These patterns can serve as ‘fingerprints’ of specific attack traffic, if high similarity between new network traffic and the ‘fingerprints’ are found, the new network traffic could potentially be modeled as suspicious (attack) traffic. As a case study, the proposed system has been analyzed on SSH network traffic flows by using four different types of SSH brute force attack data sets and two normal SSH network data sets. Our evaluations demonstrate that the SSH brute force attacks have their own behavioral patterns that are different from the behavioral patterns of the normal SSH network traffic flows. In the future, we will evaluate the robustness of the proposed framework based on the frequency of the SSH attacks. We also plan to expand this system to apply to other attack behaviours, namely distributed denial of service attacks, botnets and insider threats. Supervised learning algorithm will also be investigated to work with the trajectories to automate the intrusion detection in combining with the visualization.

Acknowledgement

This research is partially supported by the Natural Science and Engineering Research Council of Canada (NSERC) grant, and is conducted as part of the Dalhousie NIMS Lab at http://projects.cs.dal.ca/projectx/.

References

[1] V. P. Robin Sommer, “Outside the closed world: On using machine learning for network intrusion detection,” *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, pp. 305–316, 2010.

[2] A. F. Mansour Sheikhan, Zahra Jadidi, “Intrusion detection using reduced-size rnn based on feature grouping, neural computing and applications,” *Neural Computing and Applications*, vol. 21, no. 6, pp. 1185–1190, 2012.

[3] H. Akramifard, L. M. Khanli, M. ABalafar, and R. Davtalab, “Intrusion detection in the cloud environment using multi-level fuzzy neural networks,” *Proceedings of International conference on Security and Management*, pp. 152–159, 2015.

[4] G. G. N.Ch.S. N. Iyengar, Arindam Banerjee, “A fuzzy logic based defense mechanism against distributed denial of services attack in cloud environment,” *International Journal of Communication Networks and Information Security*, vol. 6, no. 3, pp. 233–245, 2014.

[5] M. Sheikhan and Z. Jadidi, “Flow-based anomaly detection in high-speed links using modified gsa-optimized neural network,” *Neural Computing and Applications*, vol. 24, no. 3, pp. 599–611, 2014.

[6] M. H. HG Kayacik, AN Zincir-Heywood, “On the capability of an som based intrusion detection system,” *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 1808–1813, 2003.

[7] T. Atkison, K. Pensy, D. E. Charles Nicholas, R. Atkison, and C. Morris, “Case study: Visualization and information retrieval techniques for network intrusion detection,” *Proceedings of the Joint Eurographics — IEEE TCVG Symposium on Visualization*, pp. 283–290, 2001.

[8] S. Intarasothonchun and W. Srimuang, “Improving performance of classification intrusion detection model by weighted extreme learning using behavior analysis of the attack,” *Proceedings of 2015 International Computer Science and Engineering Conference (ICSEC)*, pp. 1–5, 2015.

[9] S. K. A. Vinayaka Jyothi, Xueyang Wang and R. Karri, “Brain: Behavior based adaptive intrusion detection in networks: Using hardware performance counters to detect ddos attacks,” *Proceedings of 2016 29th International Conference on VLSI Design and 15th International Conference on Embedded Systems (VLSID)*, pp. 587–588, 2016.

[10] “Ncrack - High-speed network authentication cracker - Nmap,” https://nmap.org/ncrack/, 2016, [Online; accessed 19-July-2016].

[11] “Tranalyzer2 - a lightweight flow generator and packet analyzer,” https://tranalyzer.com/, 2016, [Online; accessed 19-July-2016].

[12] A. N. Z.-H. Fariba Haddadi, “Benchmarking the effect of flow exporters and protocol filters on botnet traffic classification,” *IEEE Systems Journal*, vol. pp, no. 99, pp. 1–12, 2014.

[13] T. Kohonen, *Self-Organizing Maps*. Springer, 1997.

[14] A. S. Rakesh Agrawal, Tomasz Imieliski, “Mining association rules between sets of items in large databases,” *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, vol. 22, pp. 207–216, 1993.

[15] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, “Toward developing a systematic approach to generate benchmark datasets for intrusion detection,” *Computers and Security*, vol. 31, no. 3, pp. 357–374, 2012.

[16] “DARPA Intrusion Detection Evaluation Data Set,” https://www.ll.mit.edu/ideval/data/1999data.html , 2016, [Online; accessed 19-July-2016].

[17] “Wireshark,” https://www.wireshark.org/ , 2016, [Online; accessed 19-July-2016].