User Behavior Analysis in Campus Area Networks through Kohonen Self Organizing Feature Maps

Nelson Victor Cruz Hernández

May 2017

1 Introduction

Inbounds

1.1 Background

1.2 Justification

1.3 Problem

1.4 Hypothesis

1.5 Objectives

1.5.1 General Objectives

1.5.2 Particular Objectives

2 State of the Art

2.1 Machine Learning Algorithms and Computer Security

2.2 Profiling and User classification

3 Theorical framework

3.1 Proxy

3.2 Machine Learning algorithms

3.2.1 Learning methods

Supervised Training Methods Obtain the information from "Artificial Neural Networks An introduction Kevin L. Priddy and Paul E. Keller" Chapter 2.1

Unsupervised Training Methods Obtain the information from "Artificial Neural Networks An introduction Kevin L. Priddy and Paul E. Keller" Chapter 2.3

3.2.2 Gaussian function

3.3 Self-organizing Maps

~~The concept, design, and implementation techniques of Self-Organizing Maps are described in detail in [25].~~ The Self-Organizing Map algorithm performs a nonlinear, ordered, smooth mapping of high-dimensional input data manifolds onto the elements of a regular, low-dimensional array [25]. The algorithm converts non-linear statistical relationships between data elements in a high-dimensional space into geometrical relationships between elements in a two-dimensional map (lattice), called the Self-Organizing Map (SOM)[1]. A SOM can then be used to visualize the clusters, of an input space. Each element at SOM is a neuron, and is a representation of a multidimensional vector with a cartographic position denoted with x and y. If elements in the input space are characterized using k parameters and represented by ~~k~~-dimensional vectors, each neuron in the SOM lattice is also specified as k-dimensional vector.

Cursiva

Cursiva

Cursiva

3.3.1 Learning

In the learning or training phase, the neurons in SOM try to model the input space. Self-Organizing Maps differ from from other artificial neuronal networks as they apply competitive learning as opposed to error-correction learning, such as back propagation with gradient descent, also apply a cooperative schema, using a neighborhood function to preserve topological properties of the input surfface.

* Un parrafo o idea de que vienen 3 conceptos.

a) **Competitive** Each element of the train data set is shown to every neuron in the SOM lattice. Each neuron has a response, to the shown element, the neuron that gives the best response is called the "winning" neuron, and takes it's k dimensional values adjusted so in the future it responds better to a similar input.

b) **Cooperative** Once the winning neuron has adjusted it's k dimensional values, it's neighborhood is calculated, and all neurons that are in the vicinity of the winning neuron adjust their k dimensional values so in the future they respond better to a similar input.

c) **Distance Measure** Suitable distance measure should be stablished in order to find the winner neuron. Two common used distance measures are dot-product measure and euclidean distance.

In order to use dot-product measure lattice neurons and train element vectors should be normalized. Normalization of a vector $V(v_1, v_2, v_3, ..., v_n)$ is a process

falta algo

of transforming it's components into $(\frac{v_1}{\sqrt{v_1^2+v_2^2+...+v_n^2}}, \frac{v_2}{\sqrt{v_1^2+v_2^2+...+v_n^2}}, ..., \frac{v_n}{\sqrt{v_1^2+v_2^2+...+v_n^2}})$ so that the modules of the normalized vector is unity. The dot-product of the input vector is calculated against all the neurons in the lattice, where dot-product of two vectors $Y(x_1, x_2, x_3, ..., x_n)$ and $Z(z_1, z_2, z_3, ..., z_n)$ is defined to be $\underline{x_1 \cdot z_1 + x_2 \cdot z_2 + x_3 \cdot z_3 + ... + x_n \cdot z_n}$. Using this measure means that the winner neuron is the one that gives the maximum dot-product value.

Poner como formula numerada

In the other hand euclidean distance measure does not need vector normalization and the winner neuron is defined for the minimum obtained distance. For two vectors $Y(y_1, y_2, ..., y_n)$ and $Z(z_1, z_2, ..., z_n)$ euclidean distance is given by $\underline{\sqrt{(z_1 - y_1)^2 + (z_2 - y_2)^2 + ... + (z_n - y_n)^2}}$.

→ como formula numerada

Neighborhood Function

Learning Function

3.4 Redes de Computadoras

3.4.1 Local Area Network (LAN)

3.4.2 Campus Area Network(CAN)

3.4.3 Network topology

3.4.4 OSI Model / TCPIP

3.4.5 Network Security

3.4.6 Intrusion Detection Systems

4 Methodological Development

4.1 Experiment context

Experiment was carried out on a Campus Area Network (CAN) that has a 16-bit network and a Windows domain controller, using a HTTP proxy. Among campus applications web and remote apps are included. Email service is provided by Microsoft Exchange Server which is hosted outside the campus network. The target users were full-time professors who had a computer with a static IP address and a wireless access with a dynamic IP address. Five full-time professors (hereafter denoted as users) were selected for the experiment. For each one, real usage traffic was captured (inside and outside campus activities) during a two labor weeks, and then processed.

Agregar que el → Equipo se usa mayormente en el campus pero tambien fuera del campus

4.2 Explanation

In this work Self Organizing Maps algorithm is used to create an user pattern inside a Campus Area Network. Experiment ~~is~~ are divided in three phases: network data capture, data processing and pattern evaluation. For network data capture

Comentar en revisión

phase a set of raw packages is obtained for each user through tcpdump library, process is explained in [Parres, XX]. For data processing phase each set of raw packages is arbitrary divided in build, train and evaluate sets. Each set is processed to compress raw packages into chunks of a five minutes window (t) represented by three metrics that involve communication protocol, origin and destination ip and total transmitted bytes. From obtained build dat set, a fixed number of elements (n) is randomly selected, this number defines the size of the Self Organizing Map lattice (nxn). For lattice training, a fixed number of elements (e) is randomly selected from train data set. After ten epochs training Self Organizing Map is considered to be fully trained, as a result an user network behavior pattern is obtained. Evaluation phase is done by joining different user network behavior patterns in one lattice, similar to a blanket filled of patches in which each patch is represented by an user network behavior pattern, creating what we define as the organization pattern. An organization evaluation set is build by all the user evaluation sets that belong to each user that conforms the organizational pattern. Each element of the organization evaluation set is shown to the organization pattern, resulting best matching unit is compared against the original user of the shown element to the lattice. Correct match of the user attribute of the shown element and user attribute of the best matching verifies that the shown element is able to recognize it's original user among others. Fig. 1 shows an schema of the complete process.

Parrafote partir en 2 o 3

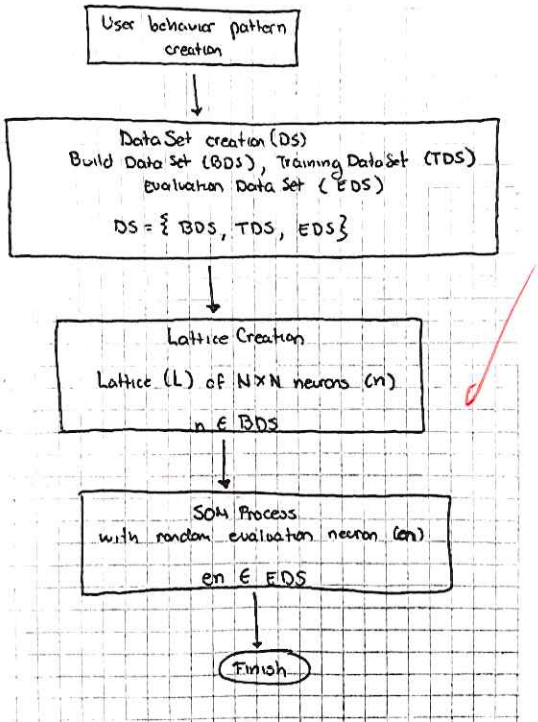
User behavior pattern creation
Data Set creation (DS)
Build Data Set (BDS), Training Data Set (TDS)
Evaluation Data Set (EDS)
DS = { BDS, TDS, EDS }
Lattice Creation
Lattice (L) of N×N neurons (n)
n ∈ BDS
SOM Process
with random evaluation neuron (en)
en ∈ EDS
Finish

Fig. 1

For each phase in the process one or more modules were build, each module's output is used as an input for the next ~~the~~ phase corresponding module.

4.2.1 Network data capture

Network data capture phase duration was ~~of~~ al(ea) two labor weeks, in which network traffic was captured from each user's computer. ~~it's important~~ to say that only ~~the owner has access to the computer and that before starting to capture~~, we checked that no computers had ~~any malicious software~~ installed. Capture module was build using jNetPcap, which works as a client application that once installed enables to capture continuously user network data in an IP packet format and save it in files of twenty megabytes each, naming them with file's creation timestamp. Average size of complete raw captured traffic for each user is three gigabytes, involving more than four million packages. Each register contains the characterization of a network connection by eight parameters organized as follows: way, origin IP, destination IP, used protocol, local used port, remote used port, total transmitted bytes and timestamp.

→ Es mi código!! no jNetPcap. y revisar

Delimitar

4.2.2 Data processing

Data processing phase is divided in four stages: a) divide raw data, b) data set creation, c) Self Organizing Map algorithm implementation and d) User network behavior pattern creation. Two modules were build for it, data set and train module. Data set module involves "divide raw data" and "data set creation" stages and Train module involves "Self Organizing Map algorithm implementation" and "User network behavior pattern creation".

Divide raw data As explained in section 3.3 Self Organizing Map algorithm is a not supervised algorithm due this different information is needed en each phase of the algorithm. Complete user raw captured data (α) is divided in three subsets: build package set (β), train package set (γ), and evaluation package set (ϕ). As data captured is divided in files containing continuous user network data, dividing the complete raw data set in subsets, enables having en each subset different days of user behavior. Complete raw data set is divided equally between each subset $\beta \cup \gamma \cup \phi = \alpha$. formula Numerada.

Data set creation Using TCP package as the working unit is not possible due the great volume of packages, and time consuming for processing [Reference, XX] each one, instead packages are processed and turned into chunks that will conform a data set. Fig 2 shows the process of dataset creation.

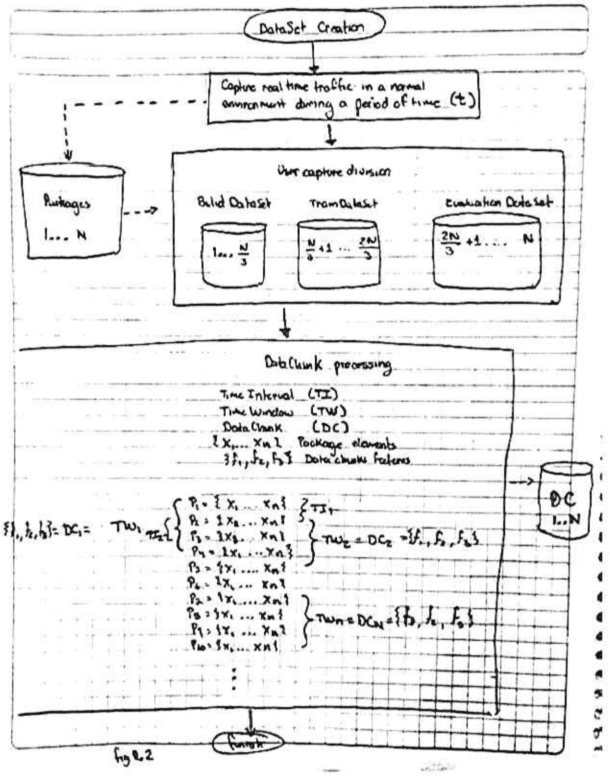

fig 2.2

Each package set (β, γ, ϕ) is processed into chunks. A chunk is defined as set of packages with a continuity in their timestamp field $P(p_1, p_2, ..., p_n)$ that represent a time window (t) of a fixed time t_1. Processing packages as a chunk allows getting a summary of the information sent in a t_n period of time, such as: total bytes sent, total bytes sent through TCP and UDP protocol, total bytes sent for web traffic destination along 80, 443 and 3128 ports and as we are working inside and Campus Area Network total bytes sent to internal destination (same backbone ip, our case 189.230.4.163). This data is condensed into three metrics: a) TCP-UDP metric, represents the ratio between total bytes sent through both protocols and total bytes sent in the chunk , b) Internal IP metric, represents the ratio between total bytes sent to CAN proxy ip and total bytes sent in the chunk and c) Web traffic metric, represents the ratio between data sent through web ports, and and total bytes sent in the chunk.

Redactar.

**** Por trabajar Para que usamos despues estas metricas El intervalo entre cada chunk Data as total bytes sent, in the t_1 time, are obtained,from which collected data is processed and three metrics are obtained and defined as follows: a) TCP-UDP metric Each data set is created by 1...N data chunks. A data chunk has a set of continuous captured packages P1... P(N) which represents a fixed time window tw of five minutes measured by the packet timestamp, in which three metrics are obtained: a) TCP/UDP metric, represents the ratio between total bytes sent through both protocols and total bytes sent in the chunk b) bytes to Internal IP metric, represents the ratio between total bytes sent to CAN proxy ip and total bytes sent in the chunk and c) web traffic metric, represents the ratio between data sent through web ports, and and total bytes sent in the chunk. This metrics will be the features which SOM algorithm will arrange the clusters. After tw is processed, a time interval ti of 10 seconds is given to start the data chunk process creation until no more packages are available. Each dataset is conformed by XXX data chunks. Due the big amount of captured packages and the complexity of processing each packaged as unit in the Self Organizing Map processing packages was needed **** Por trabajar

Self Organizing Map algorithm implementation For SOM implementation one layer square matrix of 100 x 100 neurons is used. As explained on section 4.0.1 each neuron has an individual feature vector, with specific weights. Our selected vector is conformed by three features, which summarize the total information sent in a range of time Δt over the network. For winner neuron evaluation euclidean distance is used. Winner and neighbor neurons weights are updated by a gaussian function. using random initialization

1) Initialize the map using random input vectors of fixed dimension .

2) Searching for the winner neuron.

Select an input vector x randomly from the the training data set. Search for the neuron ???? which is associated to the closest vector ???? to x which minimize the quantization error |x ? m|.

3) Updating the winner neuron and its neighboring units. For the winner neuron ???? and its neighbor U ? w, update the features vector using the following equation: ????=????+?????? Œ?Œ ??????? where ???? ?? is neighborhood function which is the decreasing function of distance d between ???? and ????? and ? is the learning rate.

4) Repeat Step 2, Step 3 with decreasing neighborhood function ???? ?? and learning rate ? until the quantization error converges enough or during the pre-defined iterations

User network behavior pattern creation This phase creates a user network pattern that represents its behavior in the network. Many pattern instances could be created from the user build dataset, as elements for creating it are randomly selected. Each neuron of the lattice is represented by an element of the Build Data Set, in which features are the three mentioned metrics in section 5.3.2. The lattice is has an arrange of 100 x 100 neurons, and a stop condition if 10 epochs.

—Paper [1] ANDSOM Module - Training

4.2.3 Pattern evaluation

Comparison between two different lattices of the same user

Comparison between different lattices of multiple users

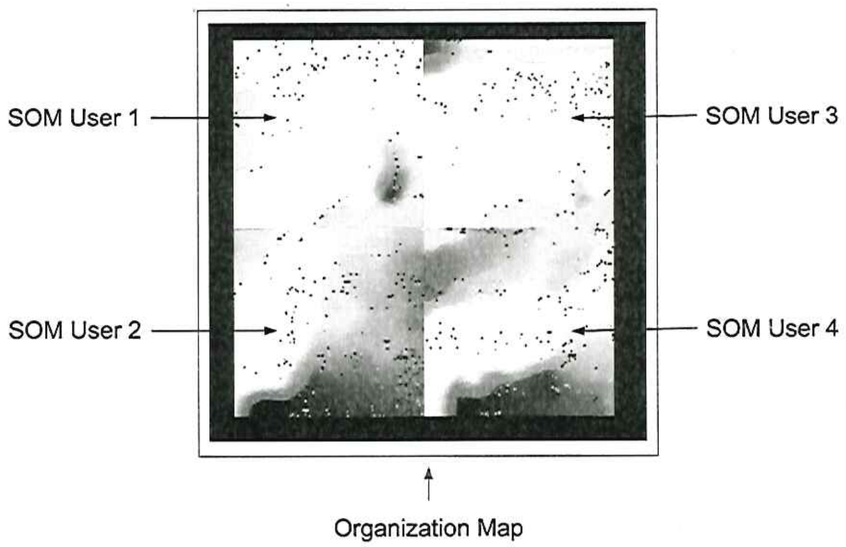

Fig.1

5 Results and Discussion

Results presentation, how the results are interpreted, and what we can do with data. The results will explain, how the user is able to recognize itself in the organization map.

6 Conclusions

6.1 Future work

Due hardware limitation, SOM training is done with ten epochs. A much longer training of about one thousand epochs would give a more precise user pattern, helping in a better user detection in the organization map. Also formulas are not completely following the standard of a gaussian function so a new implementation would be great.

7 Bibliography

[1] Ramadas, M., Ostermann, S., Tjaden, B. Detecting Anomalous Network Traffic with Self-organizing Maps. [8] Dozono, H., Itou, S., and Nakakuni, M. (2007). Comparison of the adaptive authentication systems for behavior biometrics using the variations of self organizing maps. International Journal of

Computers and Communications, 1(4), 108-116. [25] T.Kohonen. Self Organizing Maps. Springer, third edition, 2001.