A Study on Labeling Network Hostile Behavior with Intelligent Interactive Tools

Jorge L. Guerra*
LABSIN - School of Engineering - National University Of Cuyo

Eduardo Veas†
ISDS - Graz University of Technology.

Carlos A. Catania‡
LABSIN - School of Engineering - National University Of Cuyo

Abstract

Labeling a real network dataset is specially expensive in computer security, as an expert has to ponder several factors before assigning each label. This paper describes an interactive intelligent system to support the task of identifying hostile behavior in network logs. The RiskID application uses visualizations to graphically encode features of network connections and promote visual comparison. In the background, two algorithms are used to actively organize connections and predict potential labels: a recommendation algorithm and a semi-supervised learning strategy. These algorithms together with interactive adaptions to the user interface constitute a *behavior recommendation*. A study is carried out to analyze how the algorithms for recommendation and prediction influence the workflow of labeling a dataset. The results of a study with 16 participants indicate that the behaviour recommendation significantly improves the quality of labels. Analyzing interaction patterns, we identify a more intuitive workflow used when behaviour recommendation is available.

Index Terms: Human-centered computing—Visualization techniques—Heatmap—Labeling– Semi-Supervised learning;

1 Introduction

This paper describes an intelligent tool to aid the network security expert in the task of labeling network data. Network security is a challenging field of research. It builds on data-driven methods to develop techniques to identify threats, for example, building predictive models using machine learning or statistical methods [5]. Beyond user authentication, data encryption and firewalls, intrusion detection systems (IDS) serve as an active defense for the network environment, monitoring network traffic to identify security breaches (e.g., Botnet behavior) and initiate countermeasures. Most of IDS use machine learning techniques to adapt to the fast evolution of the network environment [7]. Intelligence-based Network Intrusion Detection System (NIDS) must be trained and evaluated before deployment using real labeled network traffic traces with an intensive set of intrusions or attacks [9].

One of the attacks is Botnet malware, that is one kind of threat of particular interest in network security. It is extremely hard to detect and can be used as starting point for different kinds of attacks: key logging, denegation of service and SPAM are some of them [7]. Hereby, one of the most significant issues during the development of NIDS is the lack of appropriate public datasets [19]. This issue is originated by three major challenges: i) network data contains sensitive information that organizations and individuals are not willing to disclose, ii) labeling all published data requires a major human effort, which can only be carried out by highly trained experts: security specialists. Last, the task is so specialized that there is little prior documentation on what steps security experts follow on their decisions (iii).

Responding to the challenge of releasing network data without revealing sensitive information (i), the Stratosphere Intrusion Prevention System (IPS) proposes an encoding of network behavior to facilitate the release of network data to the community [4].

Our contribution addresses the second challenge (ii) by building an intelligence based visual analytics application – RiskID – to assist the labeling of network traffic datasets. RiskID builds on the Stratosphere IPS encoding to ensure anonymity of network labeled data and combines visualization with machine learning to facilitate the recognition of malicious traffic. In this respect, the application uses several methods to classify, cluster and organize connections according to behaviour similarity as well as to directly predict labels. This intelligent guidance offers a portfolio of possibilities to approach the labelling process. Our second contribution lies in analyzing with an user study the workflows (iii) that participants follow depending on the type of intelligent guidance made available to them and the efficiency they achieve.

We can summarize the contributions of this paper in two ways:

- An interactive tool that through semi-supervised learning and visualization techniques improves the quality and speed of labeling by experts.
- A study of the strategies followed by users in the process of creating a real traffic labeled dataset.

RiskID is released as open source to improve and further develop the security communitys ability to perform botnet threat labeling. The code and data of this paper is publicly available and could be accessed in the following links [1].

2 Related Work

The lack of labeled datasets is a well-known problem in network security and has been addressed considering different aspects. An appropriate division of network traffic dataset labeling strategies depends on whether or not user assistance techniques are performed. Most of the examples found in previous works belong to the variant of automatic labeling (not user assistance): DEFCON [1] generate labeled traffic captured during the ”Capture The Flag” hacker competition, CICIDS2017 [17] using a testbed architecture and the B-Profile system [11] to generate and capturing labeling traffic, Bhuyan et al. [9] and Mukkavilli's et al [14] performed an automatic labeling using *Injection Timing* technique. Injection Timing is a widely used method consisting of recreating a network environment, control the flow during a period of time and take a known-good traffic capture injecting malicious traffic in it. This control of the network environment is not always possible. Even in controlled networks, assuring that the training datasets are correctly labeled or completely free of noise information is extremely hard.

Therefore the use of human experts are essential for annotating but they are an expensive resource, thus the labeling process must use expert time efficiently. Consequently, to reduce human effort

*e-mail: jguerra@uncu.edu.ar

†e-mail: eveas@know-center.at

‡e-mail: carlos.catania@ingenieria.uncuyo.edu.ar

[1] https://github.com/jorgeguerra881215/riskIDemo

in the labeling process it is common to find two main user support: semi-supervised learning strategies and visual applications. Aladin project [21] uses a semi-supervised approach [15] on top of active learning to foster the discovery of the different attack families, and Gornitz et al. [12] use a k-nearest neighbor approach to detect yet unknown malicious connections. Even the task of labeling those unknown connections could be hard labor. In another attempt to improve the manual labeling of connections, Soule's et al. [20] propose a web-based software system. Their tool analyzes raw network traffic, but despite the visual tools for collaborative labeling, the process of labeling a large dataset remains an arduous task. Beaugnon et al. propose a labeling strategy based on interaction with the expert mixing the two approaches: graphical user interface (GUI) and active learning [8].

Our contribution differs from previous works because we use a mixed approach: visualization and machine learning. Our approach supports the notion of a tight coupling between the system and the human [6]. Our system incorporates output from different models: clustering, item-based recommendation and prediction. The visual interface integrates these outputs in a layout that fosters comparison, showing graphically the features extracted from the network behaviour. It is expected that performance will be characterized for the entire system, not just the ML component [22]. Therefore we present an evaluation involving 16 participants to determine the level of improvement contributed by the intelligent system and the workflows that experts follow with the system.

3 System Overview

RiskID is an intelligent interactive system combining machine learning and visualization techniques to assist the user in the process of labeling network connections. To do so, the application organizes overview and detailed views of the network behavior for the user to explore and detect threats related to Botnet traffic.

3.1 Labeling Problem and Connection Characterization

The first approach in designing appropriate visualizations for network traffic analysis and supporting decision making for labeling should be to understand the objectives and needs of analysts. In order to obtain relevant information from experts that refer to the methodology necessary for the task of labelling, we carried out several informal interviews with experts on the analysis and labelling of network traffic. During this period, several prototypes were designed and iteratively tested to meet the requirements. We supplemented this information with a review of previous work focused on both the study of ILAB's role [8] and the identification of different features important for the recognition of botnet behavior. Based on the knowledge gained from the practice of network labeling, we identify a set of data requirements and tasks that must be addressed by our solution. In general, one of the issues founded in the analysis of requirements is the need to preserve the privacy of network information. When conducting a more detailed analysis on the data requirements and tasks, they could be divided into two main categories: i) Early identification of connections and ii) Analysis of connection features (e.g IP, destination Port, Protocol, periodicity with which the same connection is established, size of the package etc.).

Early Identification of connections: In order to make the labelling process more effective and efficient, it is important that users can quickly identify groups of connections that share similar features. The study with the experts revealed four important requirements to identify botnet behavior based on quick search: 1) initially an easy identification of those connections already labeled and unlabeled, 2) those connections established periodically 3) number of connections presenting a short duration 4) number of connections with small or medium size of package.

Analysis of connection: Once those connections were identified with certain features (referent to periodicity, size and period of time established) an in-depth analysis is started. Setting a label for a connection is a process that requires several observations and comparisons. In this case the user requirements are: 1) filtering connections by features (port, protocol, similar IP), 2) comparing connection behavior with another well-known connections group, for both normal and malicious behavior, 3) analysis of features like the established time of the connection, size of packages transferred in the connection, number of connections coming from the same source IP and port in a certain time interval, 4) easy manner of handling labels for each decision taken by the user.

Based on the information provided, we start the development of RiskID to meet the requirements of users for the labeling task: protecting network flow information, identifying connection groups, analyzing features and a simplify the process of labeling network connections.

3.2 System Architecture

RiskID's architecture is developed based on three main modules to cover the gained requirements. The Back-End includes a *Preprocessing Module*, and an *Analytics Module*. The process starts with a raw network traffic dataset, usually in pcap (packet capture) format. The *Preprocessing Module* is in charge of protecting the information doing a transformation of a raw network traffic dataset to internal format –a 10-dimensional feature vector– and passes it to the *Analytics Module*. The *Analytics Module* applies several statistical methods with the goal to group items and favour the early identification of behavioural patterns in the dataset of connections. In the Front-End, the *Visual Analytics Module* receives the feature vectors, statistics and grouping information and organizes them in overview and detailed views following the Visual Information-Seeking Mantra [18].

3.2.1 Preprocessing Module

The Preprocessing Module performs two conversion processes, each inside a specific submodule: the *Network Pattern Extractor* and the *Feature Extractor*. The former takes care of annonymization and the later of feature generation (two user requirements mentioned in section 3.1).

Network Pattern Extractor Submodule: The Network Pattern Extractor Submodule implements the Stratosphere IPS encoding [4] with two purposes: to reduce the usually considerable size of the network traffic data, and to guarantee data anonymity during the labeling process.

The Stratosphere IPS encoding aggregates network flows according to a 4-tuple composed of: the source IP address, the destination IP address, the destination port and the protocol. All network flows aggregated under a single 4-tuple are referred to as *Stratosphere connection* (SC), which represents the temporal behavior from one IP address to a specific service running on a specific IP address. For each flow in a SC, the encoding considers the size, duration and periodicity of packet exchange and uses characters to encode them such that: a letter defines a 3-tuple encoding $< perliodicity, duration, size >$ of a flow, a number indicates the lack of data to confirm the 3-tuple (which is normal at the beginning of a SC), a symbol indicates the time elapsed between flows. The Stratosphere project has been using this model for 5 years. It currently has a thesis and several scientific works that somehow support the importance of these features [4].

Fig. 1 shows a sample SC with symbols representing all the flows for a SC based on TCP protocol from IP address 147.32.84.164 to port 80 of IP address 209.85.148.103. The SC represents 24 flows (count of characters between numbers and letters).

Feature Vector Extractor Submodule: The Feature Vector Extractor Submodule generates a condensed representation of the network traffic dataset and represent the last data arrangement. It summarizes a SC into a 10-dimensional numerical vector denoted as feature

92*S.B.s.Z.Z*Z*Z*z*I*z*z*Z*Z*Z*Z*Z*Z*Z*S.B.s.Z*

Figure 1: The behavioral encoding of connection c-80 from IP address 147.32.84.164 to destination port 80 at IP address 209.85.148.103 using TCP.

vector:
$< x_{sp}, x_{wp}, x_{wnp}, x_{snp}, x_{ds}, x_{dm}, x_{dl}, x_{ss}, x_{sm}, x_{sl} >$. The first four dimensions of the numerical vector represent the periodicity feature (strong periodicity (sp), weak periodicity (wp), weak non periodicity (wnp) and strong non periodicity (snp) respectively), the other three refer to duration feature (duration short (ds), duration medium (dm) and duration large (dl) respectively) and the last three represent the size feature (size short (ss), size medium (sm), size large (sl)). The feature vector for a given connection is generated considering, for the complete symbol sequence, the cumulative frequency of the corresponding values associated with the behavioral encoding. At the end of the sequence, a percent of each feature is calculated and normalized to [0,1]. Formally each x_j where $j \in \{sp, wp, wnp, snp, ds, dm, dl, ss, sm, sl\}$ it is defined as:

$$x_j = \frac{1}{N} \sum_{i=1}^{N} I(t_i \in S_j) \tag{1}$$

Where N is the count of symbols that make up the SC, t_i the i-th symbol in the SC and S_j the set of characters that represents the j feature in whole connection behavioral encoding. Finally $I(.)$ is the indicator function. As example, the feature vector resultant for the connection *c-80* is:
$< sp : 0, wp : 0.13, wnp : 0.21, snp : 0.58, ds : 0, dm : 0.25, dl : 0.66, ss : 0.25, sm : 0, sl : 0.66 >$

Notice that the resulting vector after the transformation provides a similar information level about the SC except for the temporal behavior (i.e. historical information about network flows).

3.2.2 Analytics Module

For relevant information about the set of connections in the dataset the Analytics Module organizes 10-dimensional feature vectors according to standard similarity measures using a specific submodule: the Similarity Module and makes a prediction of label for those unlabelled connections through the Prediction Module.

Similarity Module: The model performs two grouping strategies. The first grouping strategy is based on clustering. Clustering improves the process of comparing SCs by offering a first approximation of similarity inside the dataset. Clustering is implemented using a k-means algorithm based on L2 distance to form the groups. The optimal number of groups is selected by the Elbow method, which consists of increasing the number of clusters until the marginal gain of the variance explained by the model is negligible.

The second grouping strategy is implemented considering the similarities between all the SCs in the dataset. The Similarity Module implements a similarity matrix by iterating over each SC in the dataset and ranking the remaining SCs according to the cosine distance function, much like an item-based recommender system. In this way, once a connection is selected from the list, the remainder connections are arranged by their similarity with the connection selected. This functionality improves the detection of sets of connections with similar features.

Prediction Module: The integration between specialist and computer tools are the key to building a great labeled dataset. Part of the interaction that we can get in RiskID is performed by Prediction Module. As the user interacts with the visual components and sets up the first set of labels, the system can learn about the importance of certain connection's features and their relation with the labels. Hereby, a semi-supervised learning strategy in the Prediction Module learns a model to issue Botnet predictions, whereby (i) label probability for connections without label is implicitly generated from behavior information as labels are assigned, and (ii) a label prediction bar in interface represents the predictions. Therefore, a minimal set of labeled connections is needed. Such first labeling process can be done following a simple selection and comparison strategy using the visual components that we will explain in section 3.3. The Prediction Module monitors the number of labeled connections. If the number of labels rises over two percent, it triggers an autonomous process for learning behavior associated to connections using the available labels. The process is carried out in the background and does not affect the user's interaction with the application.

After a learning cycle, the Prediction Module will include the resulting model to predict Botnet class probability for each unlabeled connection. All unlabeled connections with a probability higher than 0.5 will be predicted as Botnet while those below or equal to 0.5 will be predicted as Normal.

As a basic means of evidence for the prediction, Prediction Module outputs a Support Level (SL) for each prediction. SL of a connection with a predicted label refers to the percentage of connections with a similar port within the training set with which the prediction was made:

$$SL(sc_p) = \frac{|sc_{pt}|}{|sc_{pd}|} \tag{2}$$

Where sc_p refers to a SC with port p, sc_{pt} is the set of connections with port p inside the training set and sc_{pd} the set of connection with port p in whole dataset.

3.3 User Interface

The application design focus on facilitating to the user a set of visual tools to analyze the connection composition and the dataset being labeled. The user can interact with the different components of the application to obtain insights for improving the precision and confidence of the labeling process.

To address users requirements mentioned in section 3.1 the RiskID UI design has two main blocks with different levels of information detail, see Fig. 2. The application displays general information about the dataset composition in the first block together with the Connection Overview shown as a list of SCs (Fig. 2 A). The second block shows a Detailed Connection View (Fig. 2 B). For each connection selected in the list, the connection viewer displays detailed information about the connection including its current label.

The Connection List: Aiming to assist the aforementioned requirements of network threat labeling, in RiskID, we choose to use a multi-grouping Heatmap to represent the pool of SC (see Fig. 2 A) over other visualization methods for several reasons. A Heatmap is most often applied to data gathered from microarrays [16], which is a suitable analogy to the feature vector of an SC. Thus each SC is represented with a Heatmap illustrating its feature vector. Joining all connections constitutes a multi-group Heatmap. Variations in hue represent different feature types: orange:periodicity, green:duration, blue:size. Variations in value represent numerical value (darker is higher). The Heatmap serves many purposes: i) it provides an overview of behaviours in the dataset, ii) it lets the user easily recognize predominant features of each SC and iii) it intuitively relates SCs with similar features.

On the other hand, in order to support the task of analysis and identification of connections, SCs in *The Connection List* are organized into clusters (assigned in the Analytics Module) according to similarity in encoding behavior. Varying background colors help identify cluster boundaries in the list, giving the user a first approximation of similar connections. The label of an SC is shown with a traffic light metaphor (circle on the left): red circle means ”Botnet”, green circle means ”Normal” and yellow circle means ”Unlabeled”. As the user labels a new connection, the color of the circle changes accordingly. The position of the circle and its color facilitate the

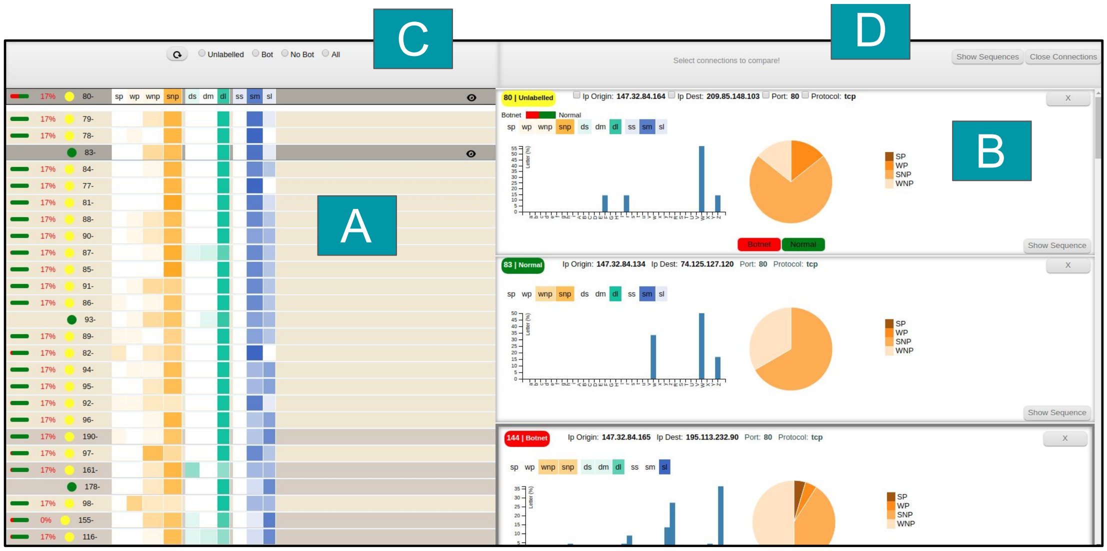

C
D
A
B
○ Unlabelled ○ Bot ○ No Bot ○ All
Select connections to compare!
Show Sequences
Close Connections
sp wp wnp snp ds dm dl ss sm sl
80 | Unlabelled
Ip Origin: 147.32.84.164 Ip Dest: 209.85.148.103 Port: 80 Protocol: tcp
Botnet
Normal
SP
WP
SNP
WNP
Show Sequence
83 | Normal
Ip Origin: 147.32.84.134 Ip Dest: 74.125.127.120 Port: 80 Protocol: tcp
144 | Botnet
Ip Origin: 147.32.84.165 Ip Dest: 195.113.232.90 Port: 80 Protocol: tcp
Letter (%)

Figure 2: RiskID application. Left block (A) displays a visual representation of all connection in the dataset grouped by their similarity (from letf to right: prediction bar, support level of prediction, current label, connection index, color representations of the feature vector). Right block B displays mean details of selected connection (histogram of behavioral model, pie chart of periodicity feature, buttons for labels selection). Sections C and D represent filtering and query options of the behavior model respectively.

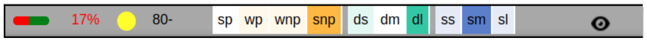


Figure 3: Prediction bar and confidence level added in connection list view after each learning and prediction process.

analysis of groups of connections with same labels. It also helps the user find potential connections to be labeled.

A new dataset initially has all connections unlabeled. Once a user sets enough labels, *The Prediction Module* comes into action and an alert notifies the user about label recommendations. Each unlabeled connection receives a prediction bar with the red color indicating the percentage of probability of Botnet. Green color indicates the percent of probability of Normal. Next to the bar, a numerical value indicates the support level of that prediction. This minimalist visual cue aims to make it easy to compare predictions over several SCs and decide where to continue labeling. Fig. 3 illustrates the label recommendation bar that appears next to an unlabeled connection (c-80, in the example). Finally, the user can customize the list of connections using a set of filtering options that appear at the top of the list (Fig. 2 C),e.g., filtering by label.

The Detailed Connection View: A second information block aims at a more in-depth analysis of connections. The Detailed Connection View, located on the right section of the application (see Fig. 2 B), displays detailed information about selected connections, including: Origin and Destination IP Addresses, Destination Port and Protocol. This network information can be used for filtering the Connection List,e.g., select from List of Connections all connections with SMTP protocol. Hereby, the user can inspect a particular subset of connections that share the selected network features.

To perform a deeper analysis, the Detailed Connection View includes also a bar chart describing the frequency of occurrence of each character from the behavioral encoding (see Fig. 6 C.6). Looking at the bar chart, the user can easily observe the differences between the character distribution along different connections. Besides, Garcia et al. emphasized the importance of periodicity for recognizing Botnet behavior [10]. Hence, the detailed connection view includes a pie chart illustrating the distribution of periodicity for each connection (Strong Periodicity, Weak Periodicity, Weak Non-Periodicity and Strong Non-Periodicity) (see Fig. 6 C.3). For users that prefer to see raw information, the original symbol-based behavioral encoding is available upon clicking a button (see Fig. 6 C.4). Finally, once decided, the user can establish the chosen label with a button click (see Fig. 6 C.5). Label changes are immediately reflected in the application and can be edited at any moment.

3.4 Interaction design

In this section, we present an example use-case that was carried out by a fictional analysts, Marie. The key goal with this section is to illustrate how analysts would label threat connections on RiskID. Maries task is to label a new dataset of connections captured during a networking day. The ultimate goal is to get as many connections as possible labeled as either Botnet or Normal. The resulting dataset will help train an IDS for early threat detection.

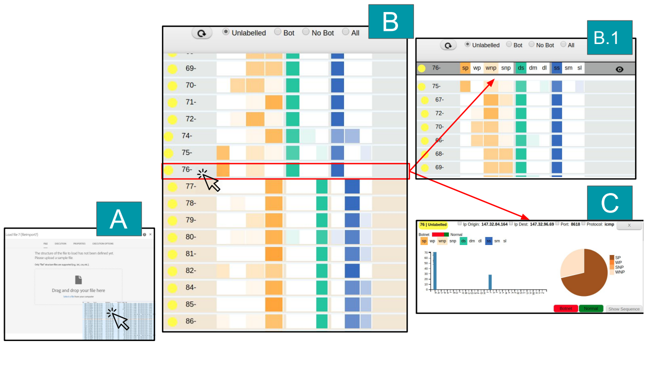

A
B
B.1
C
○ Unlabelled ○ Bot ○ No Bot ○ All
69-
70-
71-
72-
74-
75-
76-
77-
78-
79-
80-
81-
82-
84-
85-
86-
Drag and drop your file here

Figure 4: Screenshots of RiskID showing the steps taken by Marie to upload a new pcap file, study the Heatmap generated with the information loaded and select the connection with ID 76.

Marie is a specialist in network traffic analysis and receives a notification from the network administrator that she has the May 22nd network traffic capture available. In Fig. 4.A she loads the raw file pcap into RiskID. The application internally starts (in background) the preprocessing of the information that includes: extraction of features and coding of the network traffic in the Stratosphere model of letter, generation of the feature vectors, grouping of the vectors, creation of the similarity matrix and finally displaying the Heatmap on the main screen (see Fig. 4.B). Once the preprocessing is finished, the user can start interacting with the application. Marie starts looking at the Heatmap and notices that there is a small group of very similar connections that have a periodic behavior during a short time and small traffic size of package. Then Marie clicks the connections with ID *76*, trigering a reorganization of the Heatmap according to similarity with the chosen SC (Fig. 4.B.1). A second view (inside the Detailed Connection View) displays relevant information about the selected connection (see Fig. 4.C). In addition she consults information of the selected connection: the protocol *icmp*, port *8618* and origin IP *147.32.84.164* (see Fig. 6). To reduce the analysis spectrum she filters the Heatmap of connections by the *icmp* protocol. Then she observes in the list of connections the remaining connections that have a Heatmap (each connection is a Heatmap in itself) similar to the one she is analyzing to see their features too. Basing on comparisons she makes on the features of the connections and the behavior that these present, Marie concludes that the connection with ID *76* has a malicious behavior and she labels it as Botnet.

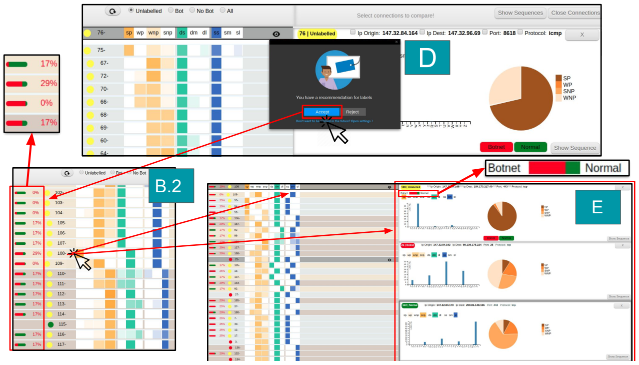
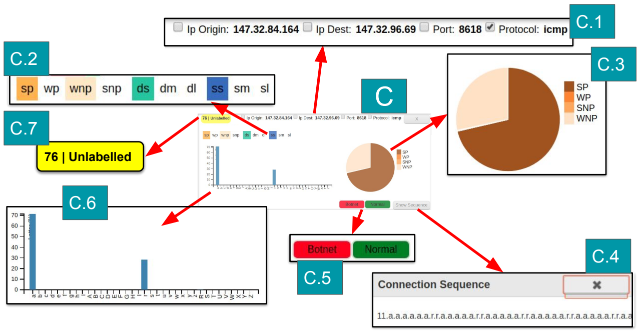


Figure 5: Screenshots of RiskID showing the steps taken by Marie to accept recommendations, identify connection (ID 108) with high Botnet probability and analyze the available detailed views to finally decide a label.

Marie, using the previous strategy, continues to label connections that she finds similar to the others. Once she has completed 2% of the dataset she receives a notification that a label recommendation is ready to be displayed (see Fig. 5.D). Automatically accepted the recommendation, a Botnet/Normal behavior recommendation appears to the left of each unlabeled connection in the Heatmap (Fig. 5.B.2).

Returning to the visualization, once there is a recommendation, the prediction probability and support level aim to help the expert decide which connection to pick next. We believe that once the first recommendation has been made, the labelling process is sped up favouring subsequent recommendations. It is a human-computer system that is becoming increasingly effective. Now Marie has a new feature that will speed up her labeling task. Using the recommendation, Marie realizes that connection with ID *108* has a high Botnet probability and a 29% of confidence. Upon choosing the connection *108*, it is brought atop the detailed connection view, the connection list is reorganized by similarity and two other SCs are brought to the detailed connection view: the most similar ”Botnet” labeled SC and the most similar ”Normal” labeled SC. The idea is to offer the expert a quick way to access the character distributions and periodicity charts for quick comparison (see Fig. 5.E).


□ Ip Origin: **147.32.84.164** □ Ip Dest: **147.32.96.69** □ Port: **8618** ☑ Protocol: **icmp**

76 | Unlabelled

Connection Sequence

11.a.a.a.a.a.a.r.r.a.a.a.a.a.a.r.r.a.a.a.a.a.r.r.a.a.a.a.a.a.r.r.a.a.a.a.a.a.r.r.a.a

Figure 6: Elements that are part of the detailed view of a SC. C1 shows features like IP Origin, IP Destiny, Port and Protocol. In addition, C1 allow users filter connection by these features. C2 represent the Heatmap of the feature vector. C.3 display a pay-chart for the periodicity feature. C4 allow users consult the Stratosphere encoding. C5 are the buttons to set the decided label for the connection. C6 means the letter distribution. C7 represent the current label for the connection, in this example the connection has not been labeled yet.

Marie analyzes the features of the selected connection and the connections followed by this in the detailed view. After a comparative analysis, she decides that the connection with ID 108 shares many similarities with the connection previously labeled as Botnet. Marie finally accepts the recommendation given by the system and labels the connection 108 as Botnet. After some time of labeling connections and receiving label recommendations from the application, Marie decides that the dataset has an acceptable number of labels. Finally, Marie downloads the resulting dataset with most of the labeled connections. She then makes the labels available to the administrator to train the next IDS.

4 User Study

We conducted an online user study to assess the value of the visualization techniques combined with a machine learning labeling prediction strategy (Learning Prediction Module). The use of the prediction Module is then contrasted to labeling just with the visual tools and features of the networks flows. The study compares four versions of the tool, offering a view of gradual inclusion of intelligent guidance. While our target is to assess the influence of the Learning Prediction Module, we do so considering the entire labeling task. Hence, we analyze the workflow and labeling strategies in each case, considering that the use of a prediction tool can increase the complexity of the UI, and potentially lead to a system that is too difficult to understand and use. In this study, we address such concerns in detail.

4.1 Evaluation Methodology

The study used four version of RiskID to measure the impact of each feature inside the system: (i) *Simple* users set labels in a semi-labeled dataset only using filter and visual tools without support from the analytics module, (ii) *CSM* users set labels as in (i) but the Heatmap reorders by cosine similarity upon selection of an SC (form analytics module only using the Similarity Module), (iii) *LPM* users set labels as in (i) but with a label predictions added (form analytics module only using the Prediction Module), and (iv) *Full* all functionalities are available (filters, visual tools, Heatmap reordering and prediction).

4.2 Dataset Description

The study was conducted using the CTU-13 group of datasets [3], which consists of a group of thirteen different malware dataset captures conducted on a real network environment taken from Czech Technical University in Prague (VUT) university campus networks. Datasets are publicly available as part of the Malware Capture Facility Project (MCFP) [2]. For the purpose of the study, the thirteen datasets were merged to create a unique CTU-13 Dataset.

4.3 Participants

We recruited participant from the last year of Informatics Engineering career in the National Technological University in Mendoza, Argentina. The RiskID study was part of a final evaluation of the Computer Security course. This guarantees that participants had knowledge of computer networks and security fields. Since the task required dealing with network flows, it was crucial that participants have an appropriate background but that they were not security experts. A total of 16 people took part in the study, from which 4 users worked with Simple version, 5 worked with CSM version, 3 worked with LPM version and the remaining 4 whit the Full version. Demographic information is summarized as follows:

- *Age*: 15 [20-29] years old, 1 [30-39].
- *Gender*: 4 female, 12 male.
- *Highest level of education*: 16 Bachelor student.
- *Worked in the field of Network Security*: 16 none
- *Created datasets for analysis of network threats before*: 16 none.
- *Labelled datasets identifying bot activity before*: 16 none.
- *Used visual analysis tools to label datasets before*: 4 yes, 12 none.

4.4 Procedure

The study was carried out in a single session with each of the 16 participants using an individual computer. Each participant was assigned to a condition randomly. Participants first received a step-by-step video tutorial introducing the main features of RiskID. The tutorial covers explanatory features such as color-codes in the Heatmap, behavioral model and prediction bar (for participants using LPM and Full). Thereafter, they had five minutes to get used to the system, e.g get familiar with Heatmap, connection selection, adjusting filter parameters. Participants were then presented with the task of labeling as many connections as possible. We asked participants to imagine they are network security workers and had to detect normal and botnet connection flows in the network. The dataset had 25% of labels to simplify the task and foster comparison. The LPM and Full version have a total of 2145 connections recommended as Normal (less than 0.5 Botnet probability) and 4843 connections recommended as Botnet (equal or more than 0.5 Botnet probability). The prediction has an accuracy value of 0.9. Participants worked over 45 minutes with the app version assigned. They basically ahd to find ”unlabeled” connections and label them either as ”Normal or Botnet”. Participants chose an anonymous username in the app, and a screen showed a scoreboard with the count of labels for anonymous name. Showing the scores with anonymous usernames encouraged participants to label more SCs.

4.5 Measurements

User Behavior. The session was logged. The start time and the time for each label event were logged as well as UI actions such as selecting a connection, opening details for a connection, etc. At the end of the session, participants had to fill a NASA TLX questionnaire [13] and a post-questionnaire asking about the interpretation of the interface visual features and the workflow followed during labeling task.

System Accuracy. We computed the True Positive Rate (TPR) (correctly ”Botnet” labels selected), True Negative Rate (TNR) (correctly ”Normal” labels selected), False Positive Rate (FPR) (set incorrect ”Botnet” labels) and the False Negative Rate (FNR) (set incorrect ”Normal” labels) for each connection labeled. Using TPR, TNR, FPR and FNR we estimated the final accuracy for each app version.

Table 1: Number of labels obtained by the users in every tested app version (Simple, CSM, LPM, and Full).

Version	Users	Labels(Correct)	Normal(Correct)	Botnet(Correct)
Simple	4	98(71)	53(28)	45(43)
CSM	5	68(45)	26(14)	42(33)
LPM	3	87(73)	36(25)	51(48)
Full	4	106(104)	45(43)	61(61)

$$Accuracy = \frac{(TPR+TNR)}{(TPR+TNR+FPR+FNR)} \tag{3}$$

Hypotheses

Intuitively, versions without Prediction Module (Simple and CSM) require more fine-tuning actions to identify a real pattern in the connection and set a label. We then expected to observe differences in system performance, labeling workflow and user satisfaction when the users try to complete the labeling task. For example the only difference in control features between version with and without a prediction module is the prediction bar as part of connection list. Clearly, participants working with versions using label prediction (LPM and Full) have another factor that influence their decisions, but that bias could affect or improve the performance. We then build the study based on the following hypotheses:

H1: Prediction of labels significantly improves labeling performance. We hypothesized higher system performance for the prediction tool with labeling interactions, though influenced by the amount of correct and incorrect labels.

H2: The number of correct labels increases over time for users using the versions with prediction (LPM and Full). Users of the versions LPM and Full would increase the system understanding over time. The experience with the system, in this case, could influence the detection of groups of connections and the speed of complete the labeling task. The quality of labeling must be determined by the cumulative experience and the acquired ability to identify similar patterns.

H3: Participants will follow a different labeling workflow in those version of the system including prediction. Participants in the prediction conditions LPM and FULL have a hint regarding what the system considers the label of a connection would be, based on available evidence. Therefore, we assume that they will perform less actions for each label and / or they will rely on different actions for their decision. However, we make no assumption as to which actions are more appropriate to successfully label a dataset.

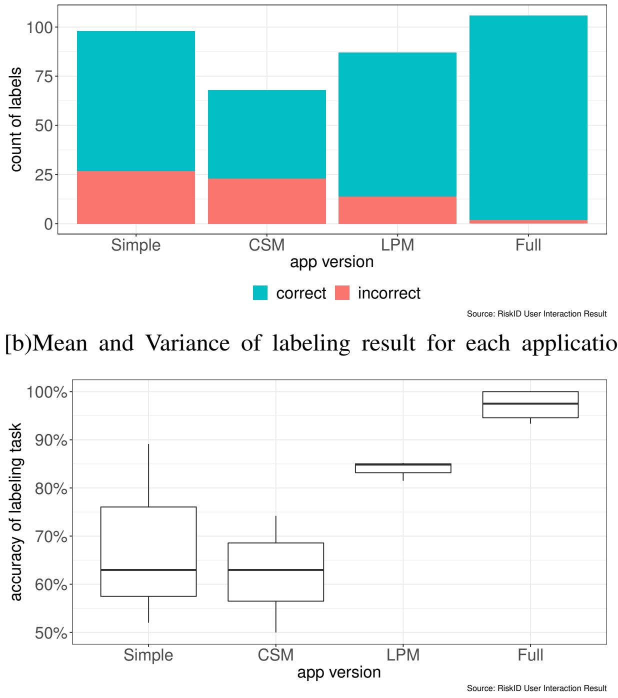
4.6 Results

This study aimed to analyze the workflow and decisions taken while looking for undesired behavior in network logs. We performed analyses of performance, effort and time spent, workflow analysis, and the personal experience.

H1: Prediction of labels significantly improves labeling performance.

We analyze performance based on the quality of labels obtained for each version during the 40 minutes (first 5 minutes were for with the system). The whole study resulted in 359 SCs labeled, 160 Normal and 199 Botnet. The 82% of the connections (293) were correctly labeled and the remainder 18% (66) incorrectly classified.

Fig. 7 a) shows for each condition the distribution of connections correctly and incorrectly labeled (green and red respectively). At first glance, versions with prediction (LPM and Full version) seem to obtain better results. Table 1 shows the exact results in accuracy. The first column represents the versions, the second column the number of participant for each version, in the third column the count of labels established, columns four and five show the number of Normal and Botnet labels respectively (in parenthesis the number



[a)Distribution between correct and incorrect labels, for each application.]

[b)Mean and Variance of labeling result for each application.]

Figure 7: General labeling result for each application.

Table 2: Prediction utility for users that worked with version using prediction feature (LPM and Full version).

Prediction utility	LPM	Full
Useful and effective prediction	72	103
Useful and not effective prediction	2	1
Not useful prediction	0	1
Not useful but effective prediction	11	1

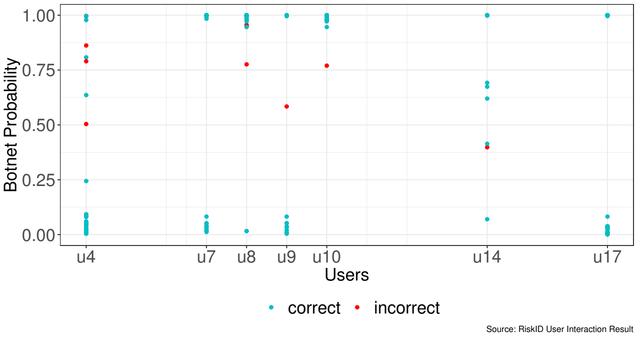


Figure 8: Labels according to prediction got by users that worked with versions using prediction feature (LPM and Full version).

of correct labels). Participants using Full version of the application labeled more connections and with almost a 100% of success (104 connection correctly labeled over 106 labels). Mann-Whitney U tests revealed no significant difference between Simple and CSM (p ≈ .82). Neither between LPM and Simple nor between LPM and CSM (p ≈ .66 and p ≈ .08). Instead, participants with Full were significantly more accurate ($p < .05$) compared to all other versions. The boxplot in Fig. 7 b) confirms these results. Participants working without prediction had relatively poor performance and broad variance.

Prediction Utility
Fig. 8 shows labels chosen in accordance with system prediction. The X-axis shows participants that worked with prediction (LPM and Full) and the Y-axis displays the Botnet probability (prediction) for each unlabeled connections. A green point in the Fig. 8 represents an SC correctly labeled and a red point an SC incorrectly labeled. First, note that most of the labeled connections lie in an extreme of Botnet probability (close to 0=Normal or close to 1 = Botnet). Therefore most of users followed the label recommendation to complete the task. Furthermore, some connections were labeled despite confusing recommendation (connection with probabilities between 0.25 and 0.75). Most of the labeling mistakes (red dots) occur in connections having a confusing recommendation.

Table 2 presents a prediction utility comparison between the LPM and Full version. It considers whether the participant followed the prediction and whether it was correct. The first column represents our prediction utility indicators (*Useful and effective prediction*: the user set the same label that was indicated in the prediction and this is correct, *Useful and not effective prediction*: the user relies on the prediction, setting the same label but this is incorrect, *Not useful prediction*: the prediction is incorrect but the users establishes correct labels avoiding the recommendation, and *Not useful but effective prediction*: the users setting wrong labels avoiding the recommendation). In both LPM and Full, participants mostly trust the prediction. Most labels were set over *Useful and effective prediction*. When participants did not rely on the prediction they obtained a poor results. Participants in LPM wrongly labeled 11 SCs obviating the prediction (*Not useful but effective prediction*). One notable aspect is that, while prediction improves the results, only the full version that also incorporated similarity recommendation obtained significantly better results. All this evidence supports **H1**.

H2: The number of correct labels increases over time as users use the versions with prediction ((LPM and Full)).
Fig. 9 illustrates labeling accuracy over time for each version and participant (in left the number of labels established by users of the different versions over each period times and in right the accuracy of labeling for the same period time). For this analysis we divided the participant interaction time in eight intervals. The X-axis represent the eight intervals and the Y-axis represents participants. The intensity of the blue color represents the accuracy/presence of labels for a participant in the corresponding time interval. Note that every participant in Simple took 3 intervals to the first label and only one participant in CSM started right away. In most of cases participants set the first label at or after the second time interval, approximately 6 minutes after starting. Users of Simple and CSM improved partially their labeling result. Users of LPM improved towards the end of the process (most intense blue after the sixth interval). Also, we perceive a lack of stability in labeling quality for participants in Simple and CSM version (constantly changes in color intensity). In case of Full version, two participants set labels in the first minutes of the study getting good results. Generally, the users of the Full version presented a regular performance in the whole process (most of intensive blue color). These results provide evidence that supports **H2**.

H3: Participants will follow a different labeling workflow for applications with prediction.
We performed action analysis of logged activity to study the workflow participants follow for labeling. To this end, we categorized actions in five types (filtering, details, letter, overview and labeling). This actions are in-line with known visualization workflow (overview, filtering, details). All filtering actions (filter by IP, filter by port, protocol, etc.) fell under the filtering category. We distin-

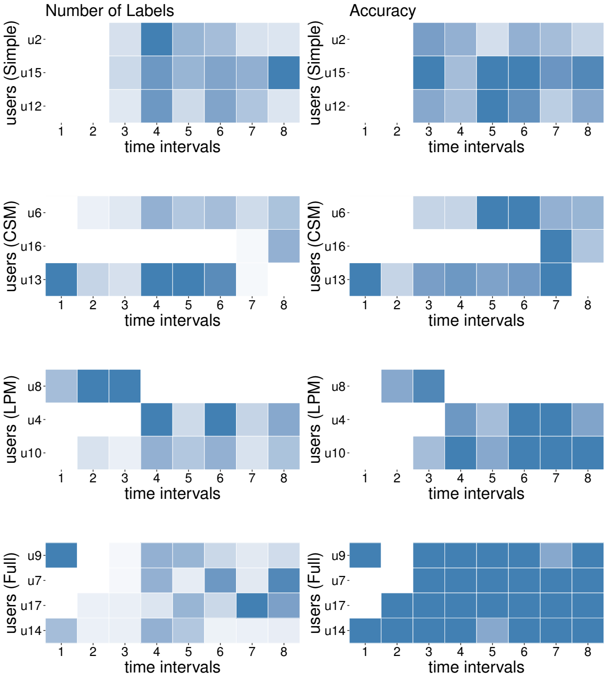


Figure 9: Labeling result for each version (Simple, CSM, LPM, Full) in different period of times. Left column shows the number of labels established by users into 8 time periods. Right column shows the labeling accuracy for each users in thee same intervals.

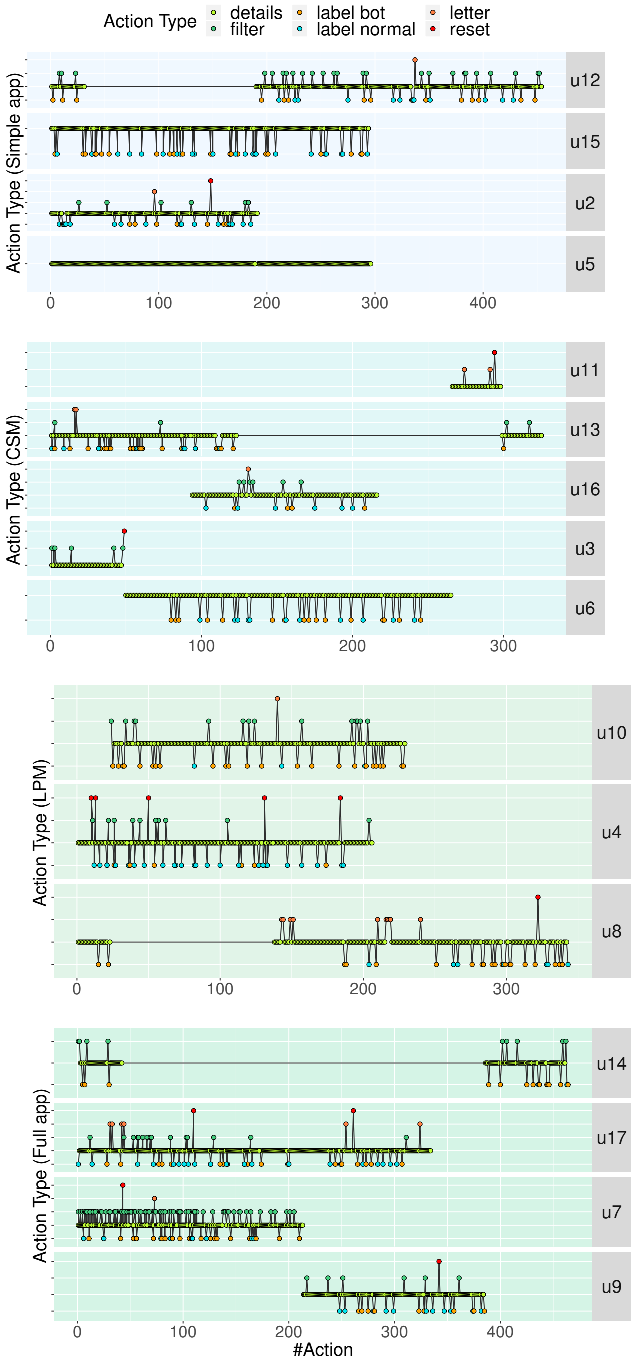


Figure 10: Workflows for each set of users (from top to down: Simple, CSM, LPM, Full version). Actions are distributed from top to bottom in five levels: filtering, details, letter behaviors, reset and labeling. Participants followed two different strategies, users that worked with prediction feature concentrated on filtering while users without these features used details and comparison.

guish two labeling actions (label botnet, label normal) and separate the detail actions (connection details=details, connection symbol sequence=letter). The overview actions represent the use of option of clean the filters or reset the connection details block, and going back to the overview display.

Fig. 10 illustrates the different strategies each user followed to accomplish the labeling task using Simple, CSM, LPM and Full versions respectively. In Simple and CSM (first and second graphs) most users relied on multiple comparisons (detail actions) to arrive at label (e.g. users u15, u6, u13). In some cases, the letter sequences (behavioral model) was opened and filter actions applied by different fields, but they did not seem to influence the labeling: in most cases filter or letter action a were not followed by label actions. Instead, users continue to make comparisons to arrive at the decision of a label (e.g. u2, u13, u16). Otherwise, u12 (in Simple) made use of filter and letter actions to establish the labels. For this case, it is notable that several label actions occur shortly after having used a filter or having observed the letter behavior. Another aspect that is striking is that three participants failed to label any connection: one in simple (u5) and two in CSM (u11, u3).

The bottom two charts in Fig. 10 represent the strategies employed by participants using the prediction feature (LPM and Full version respectively). Note that for these versions all users established labels. This sub-group of users favored filter followed by letter actions as a means to find unlabeled connections that shared characteristics with labeled ones. In most cases we see a use of the filters and the letter preceding labeling actions (eg. u10, u4, u14, u17, u7).

Deeper into the user workflow, in Fig. 11 we analyzed a set of variables that influenced labeling. From top to bottom the topmost two charts show the average time and average actions per label. The remainder charts display the average number of three specific actions per label (filters, details, and letter). Note in the fist chart that Simple version and CSM version has the less mean time per labels than LPM and Full versions. In case of actions by labels (second chart), the difference between versions is less observable. We can also see a difference in the use of filters by labels for the Full version respect another (third chart). We can not say the same for detailed actions (four chart) because there is not notable differences.The last chart reflects a greater use of behavioral model queries by the

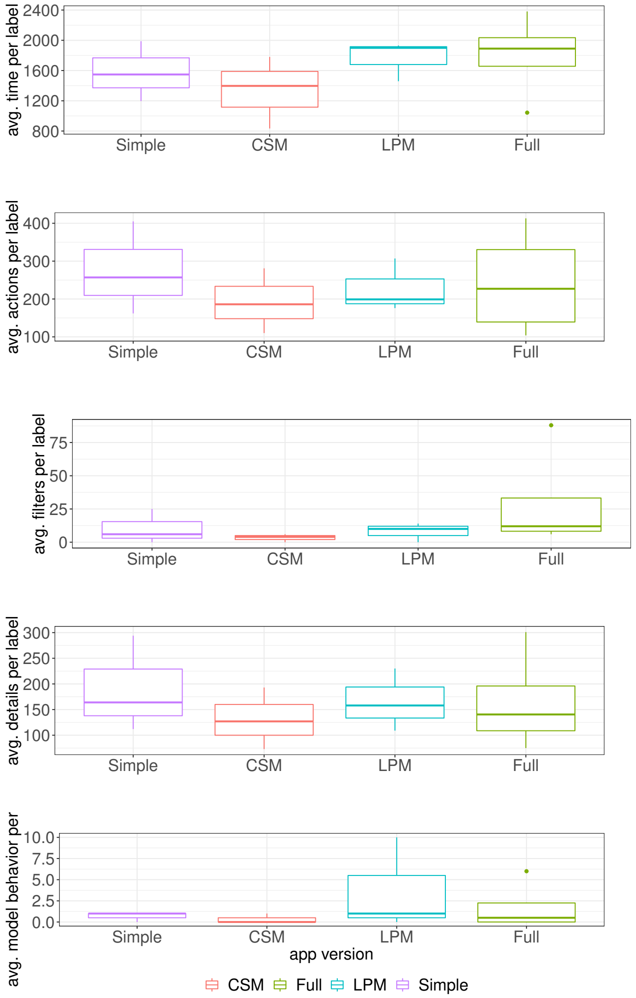


Figure 11: Percent of time, actions, filters, details and behavioral model respectively during the labeling task.

LPM version. Despite this observed differences a Mann-Whitney U test revealed that there is not significant differences between each version. These result provide evidence that partially supports **H3**.

4.7 Discussion and Limitations

In this study we assessed overall accuracy and user workflow for a visual labeling system: RiskID. A group of participants interacted indistinctly with different versions of the application to measure the impact of the different visual and intelligent components. Users had the task to label a real network dataset. We measured the accuracy of this labeling process. Analyzing the patterns of users while selecting connection in a list, searching patterns to make comparisons and setting Normal or Botnet labels. One of our contributions is in analyzing the different strategies that lead users to label flows of real connections.

Accuracy in the labelling task was acceptable considering that users were working with the application for the first time. A total of 359 connections were labeled between all versions obtaining an 81% of correct labels. The study design allows to emerge how accuracy improves as intelligence based features are added in the system. Participants in Simple versions got an 72% of good labels. The CSM version added the recommendation features, re-ordering by similarity the connection list when a connections is selected. Participants working with CSM obtained an 66% of good labels. The LPM extends the Simple version with a label prediction feature. Participants working with LPM obtained a 83% of good labels. Finally participants in the Full version obtained 98% of correct labels. This result evidences that integration of all feature significantly improvement the labeling result.

In terms of adaptability to the system, participants mostly began the labeling task after the first six minutes of interactions. Three of the users (one in Simple version and two in CSM version) did not feel comfortable enough with the system to issue any label. Time to the first label was reduced in prediction versions LPM and Full. Participants learned early to trust the prediction, but note that participants performed more actions and spent equal amount of time per connection in the Full version. We suspect that participants used several actions and time to check whether the prediction was correct.

In terms of workflows we expected to see different labelling strategies. The tools for predicting and grouping similar connections provide the user with a more complete interface for pattern detection. In spite of not observing significant differences in the workflow of each version we can notice that in the case of the versions with prediction the filter actions precede labeling actions. Seeing that these versions obtained good performance in the labeling we can say that the filters were influential to classify a large part of the connections of these versions.

As for shortcomings, the prediction feature is an advantage but could be a limitation. When a new dataset is loaded in system to be labeled the users initially have a CSM version to established the first labels. Only when the number of tagged connections exceeds two percent of the dataset, users begin to have the first label recommendation. It is widely known that a classifier performance will be influenced by the quality of the labeled data used. Since the Prediction Module builds the learning model with connections labeled according to users opinion, the quality of the labels will impact directly in the final prediction.

5 Conclusions

In this paper we introduced RiskID, a visual analysis tool to aid the network security expert in the task of labeling network data. Visual representation of network flows, machine learning and recommendation algorithms are put together to help the understanding of the main characteristics of real connection sets and consequently improve the quality of the resulting dataset.

A user study was conducted to observe the effects of the algorithms workflows and the contribution of the different components included inside RiskID. Four different RiskID versions were provided the users. According to results from post questionnaires, the visual and interaction design of the four versions were well received by the users. From the usability and workload point of view, RiskID was relatively simple to learn and usable for all participants. Moreover, users reported low effort in using the tool. In particular, study results showed clear evidence of the benefits provided by the RiskID version including visualization techniques and the intelligence based strategies (Full version). However, no significant differences were observed in user workflows when comparing different versions of RiskID.

Future works should be oriented towards analyzing the impact of RiskID in front of different scenarios: prediction robustness working with noisy data, learning rate, and others.

Acknowledgments

The authors would like to thank the financial support received by Argentinean ANPCyT- FONCYT through the project PICT 1435-2015 as well as Argentinean National Scientific and Technical Research Council.

References

[1] The shmoo group. http://cctf.shmoo.com/, October 2011. [Online; accessed April-2018].

[2] Malware capture facility project. https://mcfp.weebly.com/, October 2013. [Online; accessed May-2019].

[3] Ctu-13 dataset. https://www.stratosphereips.org/datasets-ctu13/, October 2015. [Online; accessed Jun-2018].

[4] Stratosphere ips project. https://stratosphereips.org/, October 2015. [Online; accessed Jun-2018].

[5] S. Abt and H. Baier. Are we missing labels? a study of the availability of ground-truth in network security research. In *2014 Third International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)*, pp. 40–55, Sep. 2014. doi: 10.1109/BADGERS.2014.11

[6] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, December 2014.

[7] C. and Catania and C. Garcia Garino. Automatic network intrusion detection: Current techniques and open issues. *Computer and Electrical Engineering*, 7(11):1063 – 1073, 2012.

[8] A. Beaugnon, P. Chifflier, and F. Bach. ILAB: An Interactive Labelling Strategy for Intrusion Detection. 7462:120–140, 2012. doi: 10.1007/978-3-642-33338-5

[9] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita. Towards generating real-life datasets for network intrusion detection. *International Journal of Network Security*, 17(6):683–701, 2015.

[10] S. Garcia. *Identifying, Modeling and Detecting Botnet Behaviors in the Network*. PhD thesis, UNICEN University, 2014. doi: 10.13140/2.1.3488.8006

[11] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani. An evaluation framework for intrusion detection dataset. In *2016 International Conference on Information Science and Security (ICISS)*, pp. 1–6, Dec 2016. doi: 10.1109/ICISSEC.2016.7885840

[12] N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld. Active Learning for Network Intrusion Detection. 2009. doi: 10.1145/1654988.1655002

[13] S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In P. A. Hancock and N. Meshkati, eds., *Human Mental Workload*, vol. 52 of *Advances in Psychology*, pp. 139 – 183. North-Holland, 1988. doi: 10.1016/S0166-4115(08)62386-9

[14] S. K. Mukkavilli, S. Shetty, and L. Hong. Generation of Labelled Datasets to Quantify the Impact of Security Threats to Cloud Data Centers. (April):172–184, 2016. doi: 10.4236/jis.2016.73013

[15] D. Pelleg and A. Moore. Active learning for anomaly and rare-category detection. *Advances in Neural Information Processing Systems*, 18(2):1073–1080, 2004. doi: 10.1.1.64.9664

[16] A. Pryke, S. Mostaghim, and A. Nazemi. Heatmap Visualization of Population Based Multi Objective Algorithms\rEvolutionary Multi-Criterion Optimization. 4403:361–375, 2007. doi: 10.1007/978-3-540-70928-2_29

[17] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. (Cic):108–116, 2018. doi: 10.5220/0006639801080116

[18] B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *The Craft of Information Visualization*, pp. 364–371, 2003. doi: 10.1016/B978-155860915-0/50046-9

[19] R. Sommer and V. Paxson. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. *IEEE Symposium on Security and Privacy*, 0(May):305–316, 2010. doi: 10.1109/SP.2010.25

[20] A. Soule and J. Rexford. WebClass: Adding Rigor To Manual Labeling of Traffic Anomalies. *Computer Communication Review*, 38(1):35–38, 2008. doi: 10.1145/1341431.1341437

[21] A. Sperotto, R. Sadre, F. Van Vliet, and A. Pras. A labeled data set for flow-based intrusion detection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5843 LNCS:39–50, 2009. doi: 10.1007/978-3-642-04968-2_4

[22] K. R. Varshney, P. Khanduri, P. Sharma, S. Zhang, and P. K. Varshney. Why interpretability in machine learning? an answer using distributed detection and data fusion theory. *CoRR*, abs/1806.09710, 2018.