

The University of Texas at San Antonio[™]

The Cyber Center for Security and Analytics



ZEEK INSTRUSION DETECTION SERIES

Lab 12: Developing Machine Learning Classifiers for Anomaly Inference and Classification

Document Version: 03-13-2020



Award 1829698 "CyberTraining CIP: Cyberinfrastructure Expertise on High-throughput Networks for Big Science Data Transfers"

Contents

Dverview	3
Dbjectives	3
ab topology	3
ab settings	3
Lab roadmap	4
Introduction to Weka	4
1.1 Starting Weka	4
Importing a dataset into Weka	6
2.1 Loading the training dataset	7
2.2 Filtering the training dataset	9
2.3 Training a decision table classifier13	3
2.4 Training a decision tree classifier1	5
2.4.1 Updating the decision tree classifier1	7
Reviewing the classifier's predictions on a test dataset	2
3.1 Saving the decision table	2
3.2 Using the classifier to predict labels for the test dataset	4
3.3 Viewing the predicted labels for the testdataset	6
eferences	9

Overview

This lab introduces the application of machine learning in the network security field. The lab explains how to generate a decision table and decision tree to infer scan-related network traffic. The lab is designed to train and test a machine learning classifier using network traffic dataset.

Objectives

By the end of this lab, students should be able to:

- 1. Train a decision table to classify scan-related network traffic.
- 2. Train a decision tree to classify scan-related network traffic.
- 3. Test and modify the trained classifiers and review their output classifications on a test dataset.

Lab topology

Figure 1 shows the lab topology. The topology uses 10.0.0.0/8 which is the default network assigned by Mininet. The *zeek1* and *zeek2* virtual machines will be used to generate and collect network traffic.



Lab settings

The information (case-sensitive) in the table below provides the credentials necessary to access the machines used in this lab.

Device	Account	Password
Client	admin	password

Table 1	Credentials	to access the	Client machine
TUDIC I.	Cicucituais		cheffe machine

Variable Name	Absolute Path
\$ZEEK_INSTALL	/usr/local/zeek
\$ZEEK_TESTING_TRACES	/home/zeek/zeek/testing/btest/Traces
\$ZEEK_PROTOCOLS_SCRIPT	/home/zeek/zeek/scripts/policy/protocols

Lab roadmap

This lab is organized as follows:

- 1. Section 1: Introduction to Weka.
- 2. Section 2: Building a decision classifier with Weka.
- 3. Section 3: Reviewing the classifier's predictions on a test dataset.

1 Introduction to Weka

After formatting Zeek output logs into the ARFF files, Weka is now able to process them. Weka contains the algorithms necessary to develop a number of machine learning classifiers. More information on the Weka software can be found on their documentation pages. To access the following link, users must have access to an external computer connected to the Internet, because the Zeek Lab topology does not have an active Internet connection.

https://www.cs.waikato.ac.nz/ml/weka/documentation.html

In the following sections, we train a DecisionTable and a J48 Decision Tree classifier.

1.1 Starting Weka

Step 1. From the top of the screen, click on the *Client* button as shown below to enter the *Client* machine.



Step 2. The *Client* machine will now open, and the desktop will be displayed. On the left side of the screen, click on the LXTerminal icon as shown below.



Step 3. Navigate to the Weka workspace directory.

cd Zeek-Labs/Lab-Tools/weka	
zeek@admin: ~/Zeek-Labs/Lab-Tools/weka	- + ×
File Edit Tabs Help	
zeek@admin:~\$ cd Zeek-Labs/Lab-Tools/weka zeek@admin:~/Zeek-Labs/Lab-Tools/weka\$	

Step 4. Using Java, launch the Weka software.

java -jar weka.jar		
	zeek@admin: ~/Zeek-Labs/Lab-Tools/weka	- + ×
File Edit Tabs Help	he / ah Teale /wakat java jar waka jar	
zeek@admin:~/zeek-La	DS/Lab-Tools/wekas java -jar weka.jar	

Step 5. Once Weka has been loaded, a notification containing Weka related information will be displayed. Select the *OK* button to continue to the *Weka GUI Chooser panel*.



The Weka GUI Chooser panel will look similar to the following image.



Step 6. For this lab, we will be using the *Explorer* application. Click the *Explorer* button to launch the application.

plic	ations
	Explorer
Ex	perimenter
Kno	owledgeFlow
۷	Vorkbench
9	Simple CLI

Weka has been successfully launched and we can proceed to the next section.

2 Importing a dataset into Weka

Once the *Explorer* application opens, a new GUI window will be displayed. Initially, this window has all options greyed out, indicating that we have not yet opened or loaded a dataset.

The *Explorer* panel contains a *Menu Bar* located at the top of the GUI window. There is a total of 6 additional panels, which contain related information necessary to train, test and visualize classifiers developed while using Weka. By default, the *Preprocess* panel will be selected.

The *Preprocess* panel is used to import a training dataset to be used for training a machine learning classifier. Features can be removed, randomized or appended within this panel.

0	Weka E	xplorer		- + ×
Preprocess Classify Clus	ster Associate Select attributes	Visualize		
Open file Open U	JRL Open DB Gene	rate Undo	Edit	Save
Filter				
Choose None				Apply Stop
Current relation		Selected attribute	e	
Relation: None Instances: None	Attributes: None Sum of weights: None	Name: None Missing: None	Weight: None Distinct: None	Type: None Unique: None
Attributes				
All None	Invert Pattern			Visualize All
Status)			
Welcome to the Weka Exp	lorer			Log 💉 x 0

2.1 Loading the training dataset

Step 1. On the top left of the *Preprocess* window the *Open file* button can be found. Click the *Open file* button to load the training dataset.

Preprocess	Classify	Cluster	Associate	Select at	tributes	Visualize
Open file	. 0	pen URL	. Oper	n DB	Gene	erate

Step 2. Enter the path to the *trainset.arff* file. Alternatively, use the GUI to navigate to the lab workspace directory to select the file. Use the *Open* button to load the *trainset.arff* file into Weka.

/home/zeek/Zeek-Labs/Sample-PCAP/trainset.arff

0	Open	- + ×
Look In:	Sample-PCAP	
tests	et.arff	Invoke options dialog
		+ = ↑ + b 6
File <u>N</u> ame	: /home/zeek/Zeek-Labs/Sample-PCAP/trainset.arff	
Files of Ty	pe: Arff data files (*.arff)	
		Open Cancel

After click the *Open* button, the *Preprocess* panel will be updated to contain the *trainset.arff* file statistics.

Each section header has been highlighted with a red box. We can see that the *Current* relation, Attributes and Selected attribute sections have been updated to contain trainset.arff file data.

Step 3. Within the *Attributes* section, click the *class* feature to change the active attribute.

Q Wel	ka Explorer - +
Preprocess Classify Cluster Associate Select attribu	ites Visualize
Open file Open URL Open DB (Generate Undo Edit Save
Filter	
Choose None	Apply Stop
Current relation	Selected attribute
Relation: networktrafficAttributes: 8Instances: 1401Sum of weights: 1402	Name: time Type: Numeric 1 Missi 0 (0%) Distinct: 1401 Unique: 1401 (100%
Attributes	Statistic Value
All None Invert Pattern	Minimum 1295981542484409 Maximum 1561919070053261 Mean 1453721842608295.5 StdDev 130687564832435.69
1 time 2 sourceip	Class: class (Nom)
3 destip 4 sourceport 5 destport 6 protocol 7 duration	570
Remove	0 0 0 0 0
Status	1295981542484409 1428950306268835 156191907005
ок	Log 🛷

By selecting the *class* feature within the *Attributes* section, the *Explorer* panel will be updated to display the active feature.

Within the *Current relation* section, our dataset's name, *networktraffic*, is displayed. Additionally, it is shown that the dataset contains 1401 unique data objects (instances).

Within the *Selected attribute* section, the *class* labels added to the dataset in the previous lab are counted. The *trainset.arff* dataset contains 831 data objects labeled with a 1, belonging to the malicious class, while 570 data objects are labeled with a 0, belonging to the benign class.

At this point, *trainset.arff* dataset has been successfully loaded into Weka and we can begin filtering the data before training a machine learning classifier.

2.2 Filtering the training dataset

The majority of machine learning classifiers are unable to handle string attributes. For network analysts, source and destination IP addresses are valuable features that are often necessary for traffic analysis. However, these IP addresses are unable to be stored as string values when training a classifier.

There is a number of ways to address this issue. If a network analyst were to know all of the unique IP addresses, when generating their ARFF dataset, they can create the nominal values similar to how we created the nominal protocol values.

Because Internet-scale traffic contains a very large number of unique IP addresses, the aforementioned process may not be feasible. Therefore, in the previous lab, we converted our source and destination IP addresses into numerical values. In this section, we will be using all unique iterations of the numerical values to generate a nominal list. By reformatting the IP addresses into numeric values using Terminal utilities, the Weka software will be able to select all unique IP addresses and convert them into a nominal feature set.

Step 1. Within the *Preprocess* tab, under the *Filter* section, click the *Choose* button.

Preprocess	Classify C	Cluster Assoc	iate Select at	ttributes Visua	alize		
Open file	Oper	n URL	Open DB	Generate	Undo	Edit	S
Choose	None					Appl	y

Step 2. Under the *unsupervised* option, select the *attribute* option to display a list of attribute-based filters.



Step 3. Scroll to the *NumericToNominal* filter and double click to select it.



Step 4. Within the *Filter* section, click the *first-last* description to modify the filter.

Filter	
Choose NumericToNominal -R first-last	Apply Stop

Step 5. Update the *Indexes of the Attributes* to be filtered. Click the *Apply* button to edit the indexes.

Jour		
A filter for turning n	umeric attributes into nominal ones.	More
		Capabilities
debug	False	
NotCheckCapabilities	False	
	G	

Step 6. On the right side of the *Filter* section, click the *Apply* button to apply the modified filter.

Filter	
Choose NumericToNominal -R 2-3	Apply Stop

The source and destination IP addresses will now be converted to the Nominal feature type.

Step 7. Within the *Attributes* section, click the *sourceip* feature to change the active attribute.

By selecting the *sourceip* feature within the *Attributes* section, the *Explorer* panel will be updated to display the active feature.

Within the *Selected attribute* section, the *sourceip* feature will now display the Nominal data objects. In the following image, the highlighted *sourceip* related to the scanning machine's IP address (192.168.1.3), displays 831 unique instances being recorded.

0	Weka E	xplorer			-	+ ×
Preprocess Classify Cluster Associa	te Select attributes	Visualize				
Open file Open URL	Open DB Gene	rate	Undo	Edit.	Save	ə
Filter						
Choose NumericToNominal -R 2-3					Apply	Stop
Current relation		Selected	attribute			
Relation: networktraffic-wek Instances: 1401 Sur	Attributes: 8 m of weights: 1401	Name: Missing:	: sourceip : 0 (0%) E	Distinct: 8	Type: Nomir Unique: 4 (0%	nal .)
Attributes		No.	Label	Count	Weight	
All None Inver	t Pattern	1 2 3 4	10022 100215 19216813 65552560	1 76 831 1	1.0 76.0 831.0 1.0	
2 🗹 sourceip 3 🛄 destip		Class: class	s (Nom)		Visu	alize All
4 sourceport 5 destport 6 protocol 7 duration 8 class	J		831			
Remove		1 76	1	161	329	
Status						
ок					Log	× 0

Additionally, the *Selected attribute* section will be updated to show new statistics for each feature. The updated *Selected attribute* section is displayed in the previous image.

2.3 Training a decision table classifier

Step 1. Within the *Explorer* panel, click the *Classify* tab located at the top of the *Explorer* panel to switch to the *Classify* panel.

(3				Weka E	xplorer	-	+	×
	Preprocess	Classify	Cluster	Associate	Select attributes	Visualize			
-	Classifier								

Step 2. Once the *Classify* panel has loaded, click the *Choose* button within the *Classifier* section to select which machine learning classifier we are developing.

0				Weka B	Explorer
Preprocess	Classify	Cluster	Associate	Select attributes	Visualize
Classifier					
Choose	ZeroR				

Step 3. Under the *rules* collection, double-click with your mouse to select the *DecisionTable* classifier.

CI	assifier
Te	 weka classifiers bayes functions lazy meta misc rules
(1	DecisionTable JRip M5Rules OneR PART ZeroR trees

Step 4. Under the *Test options* section, click the *Start* button to begin training the classifier. Notice the *Classifier* section has been updated to display the *DecisionTable* classifier.

Contraction of the second seco		Y	- Y	-
Preprocess	Classify	Cluste	er Ass	0
Classifier				
Choose	Decisior	Table	-X1-S	
Test options	1			
🔘 Use train	ing set			
O Supplied	test set	Set		
Cross-val	idation	Folds	10	
O Percenta	ge split	%	66	
Mo	ore option	IS		
(Nom) class			Ļ	No. 1
	_			-

Step 5. See the *Decision Table* classifier's results.

Inconfectly ch	aborited th	Jeances	v		0	
Kappa statisti	C		1			
Mean absolute	error		0.00	16		
Root mean squa	red error		0.00	16		
Relative absolu	ute error		0.32	81 %		
Root relative	squared err	or	0.32	81 %		
Total Number o	f Instances		1401			
Weighted Avg.	1.000	0.000	1.000	1.000	1.000	1.000
=== Confusion I a b <	Matrix === classified = l	as				

Within the *Result list* section we can see our new *Decision Table* that has been trained with the *transet.arff* dataset. Within the *Classifier output* section, we can see the prediction results for the *Decision Table* classifier. The *Confusion Matrix* depicts that the classifier had a 100% accuracy when predicting labels after being trained.

2.4 Training a decision tree classifier

Step 1. Click the *Choose* button within the *Classifier* section to select which machine learning classifier we are developing.

0				Weka E	xplorer
Preprocess	Classify	Cluster	Associate	Select attributes	Visualize
Classifier					
Choose	Decision	Table -X	1 -S "weka.	attributeSelection.	BestFirst -D 1 -N 5"

Step 2. Under the *trees* collection, double-click with your mouse to select the *J48* decision tree classifier.

CL	assifier	
	🔻 📄 weka	
-	Classifiers	
Te	bayes	
	►	
	► 📄 lazy	
	🕨 🚞 meta	ļ.
	► 📄 misc	
	rules	
	🔻 🚞 trees	
	DecisionStump	
	HoeffdingTree	
	¹ J48	
()	LMT	
-	M5P	
L	🕒 RandomForest	
R	RandomTree	
1	EPTree	

Step 3. Under the *Test options* section, click the *Start* button to begin training the classifier. Notice the *Classifier* section has been updated to display the *J48* classifier.

assifier		
Choose] J48 -C 0.2	25 -M 2	
st options		
🔿 Use training set		
O Supplied test set	Set	
Cross-validation	Folds	10
O Percentage split	%	66
More option	ns	
om) class		
Start	Stor	

Step 4. See the *J48 Decision Tree* classifier's results.

Incorrectly classified instances			-		0.0714	0
Kappa statistic	0.99	85				
Mean absolute e	0.00	07				
Root mean squar	ed error		0.02	67		
Relative absolu	te error		0.14	79 %		
Root relative s	quared err	or	5.43	85 %		
Total Number of	Instances		1401			
Weighted Avg.	0.998	0.002	1.000	0.998	0.999	0.999
=== Confusion M	latrix ===					
a b <	classified	as				
	= 1					
831 0 a						

Within the *Result list* section we can see our new *J48 Decision Tree* that has been trained with the *transet.arff* dataset. Within the *Classifier output* section, we can view the prediction results for the *Decision Tree* classifier. The *Confusion Matrix* depicts that the classifier did not have a 100% accuracy when predicting labels after being trained and misclassified a single malicious data packet as benign.

2.4.1 Updating the decision tree classifier

Because our *J48 Decision Tree* has made an error in predicted a label, we can attempt to remove or add additional features to increase the classifier's accuracy.

Step 1. Right click the *J48 Decision Tree* under the *Result list* section to display more options. Click to *Visualize* the *J48 Decision Tree*.

	TP Rate FP Rat				
(Nom) class	View in main window View in separate window				
Start Stop	Save result buffer				
Result list (right-click for optio	Delete result buffer(s)				
16:18:36 - rules.DecisionTable 16:21:45 - trees.J48	Load model Save model Re-evaluate model on current test set Re-apply this model's configuration				
	Visualize classifier errors				
-	Visualize tree				
Status	Visualize margin curve				
ок	Visualize threshold curve				
	Cost/Benefit analysis				
🕋 🛅 🔚 💽 🔤 [zeek@a	Visualize cost curve				

Step 2. View the Visualized J48 Decision Tree.



We can see the *time* feature column was the only decision node within the tree. For the purposes of this lab, the datasets were collected at varying times; therefore, the decision tree had an over reliance on the *time* feature to determine when the malicious and benign events took place.

Step 3. Click the top right \boxtimes button to close the Tree Visualizer window then, within the *Explorer* panel, click the *Preprocess* tab located at the top of the *Explorer* panel to switch to the *Preprocess* panel.

)				1	Weka B	Explore
Preprocess	Classify	Cluster	Associate	Select at	tributes	Visualize
Open file	.] [0]	oen URL	.) (Ope	n DB	Gene	erate

Step 4. Once the *Preprocess* tab has loaded, click the *time* feature within the *Attributes* section and select the *Remove* button.

All	None Invert Pattern
No.	Name
1	. 🗹 time
2	sourceip
3	destip
4	sourceport
5	destport
6	protocol
7	duration

With the *time* feature removed, we can retrain our decision tree to view the new accuracy.

Step 5. Within the *Explorer* panel, click the *Classify* tab located at the top of the *Explorer* panel to switch to the *Classify* panel.

1	0	Weka Explorer		- +	×			
	Preprocess	Classify	Cluster	Associate	Select attributes	Visualize		
-	Classifier		-					

Step 6. The *J48 Decision Tree* should still be selected. Under the *Test options* section, click the *Start* button to begin training the classifier. Notice the *Classifier* section has been updated to display the new *J48* classifier.

O Use training set		
O Supplied test set	Set	
Cross-validation	Folds	10
O Percentage split	%	66
More option	าร	
(Nom) class		

Step 7. See the J48 Decision Tree classifier's results.

Kappa statistic Mean absolute er Root mean square Relative absolut Root relative sq	ror d error e error		0.99	26 3		
Mean absolute er Root mean square Relative absolut Root relative sq	ror d error e error		0.00	3		
Root mean square Relative absolut Root relative sq	d error e error		0 04			
Relative absolut Root relative sq	e error		0.04	15		
Root relative sq			0.61	31 %		
	uared err	or	8.44	11 %		
Total Number of	Instances		1401			
	TP Pate	ED Pate	Procision	Pocal1		MCC
becarco nee	aracy by	ctubb				
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC
	0.999	0.007	0.995	0.999	0.997	0.993
	0.993	0.001	0.998	0.993	0.996	0.993
Weighted Avg.	0.996	0.005	0.996	0.996	0.996	0.993
	1041000					10000000000000
=== Confusion Ma	trix ===					
10 10 100 1000 000		and a second				
a b < c	lassified	as				
830 1 a =	1					
4 566 b =	0	- 12				

Within the *Result list* section we can see our new *J48 Decision Tree* that has been trained with the *transet.arff* dataset. Within the *Classifier output* section, we can view the prediction results for the *Decision Tree* classifier. The *Confusion Matrix* depicts that the classifier actually had a worse accuracy than the previously trained *J48 Decision Tree*.

In this example, we highlight the importance of choosing the best fit features when training a classifier. In a real-time network environment, it may take multiple tests before discovering which features are necessary for classifying a specific anomaly.

Step 8. Right click the newest *J48 Decision Tree* under the *Result list* section to display more options. Click to *Visualize* the *J48 Decision Tree*.

-	IP Rate
(Nom) class	View in main window
Start Result list (right-click fo	View in separate window Save result buffer Delete result buffer(s)
16:18:36 - rules.Decisior 16:21:45 - trees.J48 16:29:46 - trees.J48	Load model Save model Re-evaluate model on current test set Re-apply this model's configuration
	Visualize classifier errors
	Visualize tree
Status	Visualize margin curve Visualize threshold curve
	Cost/Benefit analysis Visualize cost curve

Step 9. View the Visualized J48 Decision Tree.



Because the *Decision Tree* has a larger number of nodes, we are unable to see some of the decision thresholds. The following steps will explain how to scale the *Visualized* tree.

Step 10. Right click on the Visualized J48 Decision Tree and select the Auto Scale option.



Step 11. Click the top right \square button to close the Tree Visualizer window then, open the *Visualized J48 Decision Tree*.



Here we can see the new *J48 Decision Tree* has multiple layers and decision nodes. The *duration* feature has replaced the *time* feature as the root node, and the *sourceip* feature is used to further classify the dataset. However, because this tree has reduced accuracy, we will be continuing the lab by using the *Decision Table* created initially.

3 Reviewing the classifier's predictions on a test dataset

Now that we have determined that the *Decision Table* was a more accurate classifier, we can begin testing the classifier's accuracy using the test dataset.

3.1 Saving the decision table

It is possible to save a trained classifier to be reused in future instances of testing and classification. This section will introduce how to save a trained classifier.

Step 1. Under the *Result list* section, right click on the *Decision Table* and select *Save model*.

(Nom) class	View in main window
Start Stop Result list (right-click for option	View in separate window Save result buffer Delete result buffer(s)
16:18:36 - rules.DecisionTable	Load model Save model
16:21:45 - trees.J48 16:29:46 - trees.J <mark>4</mark> 8	Re-evaluate model on current test set Re-apply this model's configuration
	Visualize classifier errors Visualize tree
Status	Visualize margin curve
ОК	Visualize threshold curve
	Cost/Benefit analysis
🕋 🥅 🎦 🕞 🔤 [zeek@a	Visualize cost curve

Step 2. Navigate to the Lab workspace directory and save the *Decision Table*. Alternatively, use the GUI to navigate to the lab workspace directory to select the file. Use the *Save* button to save the new *DecisionTable* file into Weka.

/home/zee	k/Zeek-Labs/Sample-PCAP/DecisionTable	
0	Save	- + ×
Look In:	Sample-PCAP	
File <u>N</u> ame:	/home/zeek/Zeek-Labs/Sample-PCAP/DecisionTable	
Files of Type:	Model object files	
		Save Cancel

Once saved, we can proceed to testing the classifier's accuracy on predicting labels for the test dataset.

3.2 Using the classifier to predict labels for the test dataset

Step 1. Under the Test options section, select the Supplied test set then, select *Set* button to load the select the *Set* button to load the test dataset. Within the *Test Instances* window, click the *Open file* button.

Test options	Classifier output	
O Use training set	C Test Ir	nstances – + ×
Supplied test set Set	Relation: None	Attributes: None
O Cross-validation Folds 10	Instances: None	Sum of weights: None
O Percentage split % 60	Open file Open URL	
More options	Class No class	
(Nom) class		Close

Step 2. Select the *testset.arff* file and click the *Open* button to load the test dataset then, click the Close button.

0		Open		- + ×
Look <u>I</u> n:	Sample-PCAP			
tests trains	et.arff set.arff			ce options dialog
File <u>N</u> ame Files of <u>T</u> y	: testset.arff pe: Arff data files (*.arff)		- + -	
			Op	en Cancel

Step 3. Under the *Test options* section, select the *More options* button to configure the classifier to match the following image then, click on *OK*.



Step 4. Under the *Result list* section, right click on the *Decision Table* and select *Re-evaluate model on current test set*.

(Nom) class	View in main window
Start Stop Result list (right-click for option	View in separate window Save result buffer Delete result buffer(s)
16:18:36 - rules.DecisionTable	Load model Save model
16:21:45 - trees.J48 16:29:46 - trees.J48	Re-evaluate model on current test set Re-apply this model's configuration
	Visualize classifier errors
Status	Visualize margin curve
ок	Visualize threshold curve
	Cost/Benefit analysis

Step 5. After filtering the *sourceip* and *destip* features into Nominal attributes, the *testset.arff* file will not be properly formatted. Weka will need to update the *testset.arff*

dataset to be used by the classifier. Select the Yes button on the ClassifierPanel pop-up panel.



The classifier will generate new predictions, which can be viewed by saving the resulting *.arff* file.

3.3 Viewing the predicted labels for the testdataset

To save the resulting .arff file,

Step 1. Within the *Explorer* panel, click the *Visualize* tab located at the top of the *Explorer* panel to switch to the *Visualize* panel.

0			Weka E	xplore	r			-	+ ×
Preprocess	Classify Cluster	Associate Select	attributes	Visualize					
Plot Matrix	sourceip	destip	sourcep	oort	destport	proto	col	duration	
class	u un u		1000			6 B			
	14	8	8	a (=		10	12	ć	
	1					1			7
duration									
protocol			e v r			e.			7
PointSize: [1]	·			Update					
Jitter:	·			Select At	tributes				
Class Colou	r ·····								
Status							_		
ОК								Log	P X 0

Displayed will be resulting graphs from attribute correlations solved by the machine learning classifier.

Step 2. Select the *duration x duration* graph, found in the sixth column (*duration*) and second row (*duration*).

Plot Matrix	sourceip	destip	sourceport	destport	protocol	duration
class	1 3 5 5		778 E			
	- 18	-	a			1.54
	1				ł	
duration		an Marine Anton				

Step 3. Click the Save button.

X: duration (Num)		-	Y: duration (Num)	
Colour: class (Nom)		•	Select Instance	
Reset Clear	Open Save		Jitter 🔾	
lot: networktraffic-	veka.filters.unsupervised.	attr	ibute.NumericToNominal-R2-3	-weka.filters.unsupervised.attr.
258520601	**** **** *** **** ****	××	×******	B
3	1292603	02	258520601	

Step 4. Navigate to the Lab workspace directory and save the *DecisionTableResults*. Alternatively, use the GUI to navigate to the lab workspace directory to select the file. Use the *Save* button to save the new *DecisionTableResults* file into Weka.

/home/zeek/Zeek-Labs/Sample-PCAP/DecisionTableResults

0	Save		- + ×
Look <u>I</u> n:	Sample-PCAP		
tests 🕒 trains	t.arff et.arff		
		1	
File <u>N</u> ame	/home/zeek/Zeek-Labs/Sample-PCAP/DecisionTableResult	s	
Files of Ty	e: Arff data files		T
			Save Cancel

Step 5. Close all of the Weka tabs with the orange *x* on the top right corner of each panel.

Step 6. Return to the Terminal and navigate to the lab workspace directory.

cd ~/Zeek-Labs/Sa	mple-PCAP/	
1	zeek@admin: ~/Zeek-Labs/Sample-PCAP	- + x
File Edit Tabs Help		
zeek@admin:~\$ <mark>cd</mark> zeek@admin:~/Zeek	Zeek-Labs/Sample-PCAP/ -Labs/Sample-PCAP\$	

Step 7. Using a text editor, view the *DecisionTableResults.arff* file.

nano DecisionTableRe	sults.arff	
	zeek@admin: ~/Zeek-Labs/Sample-PCAP	- + ×
File Edit Tabs Help		
zeek@admin:~/Zeek-La zeek@admin:~/Zeek-La	<pre>bs/Sample-PCAP\$ nano DecisionTableResults.arff bs/Sample-PCAP\$</pre>	

The file will be opened, and each data object will contain a new classification label.

	Ze	eek@admin: ~/Zeek	-Labs/Sample-PCAP	- + ×
File Edit Tab	os Help			
GNU nano	2.9.3	DecisionTab	leResults.arff	
@relation	networktraffic	-weka.filters.un	supervised.attribute	.NumericToNominal\$
@attribute @attribute @attribute @attribute @attribute @attribute @attribute	sourceip {100 destip {10022 sourceport num destport nume protocol {tcp duration nume class {1,0}	22,100215,192168 ,10023,10112,100 meric ric ,udp,icmp} ric	13,65552560,17216255 215,1002255,1721601,	1,655557251,19216\$ 2454514,6443557,6\$
@data 19216813,1 172162551, 19216813,1 19216813,1 19216813,1 172162551, 1921683131	9216822,49526, 18912644128,50 2049163158,106 9216822,49526, 9216822,49526, 9216822,49526, 255255255255,6 ,65549575,5633	1755,tcp,3,1 983,3192,udp,259 30,80,tcp,150006 9002,tcp,3,1 15660,tcp,4,1 2607,tcp,4,1 8,67,udp,?,0 2,80,tcp,6663279	354,0 ,0	

Traffic found in the first row of data was labeled with a 1, as malicious traffic. Traffic found in the second row of data was labeled with a 0, as benign traffic.

Concluding this lab, we have introduced the capabilities of implementing a machine learning classifier to detect specific anomalies or events. Multiple classifiers can be used for training network security classifiers, and the features within each training dataset can have a profound impact on the classifier's accuracy. By removing, modifying or adding new features you can test the accuracy of a classifier. In this lab, we generated a *Decision Table* that was capable of labeling malicious and benign traffic.

References

1. "Attribute-relation file format", The university of waikato, [Online], Available: https://www.cs.waikato.ac.nz/~ml/weka/arff.html