MCI project First Milestone Report

Team number: 19
Project Title: Development of a Network Forensic Analytic System

Milestone 1	Activities	Planned Outputs	Achieved Outputs
Restate the milestone from your Draft plan.	Restate the key activities from your draft plan.	Restate the planned outputs from your draft work plan.	Outline the actual outputs compared to what was projected (or type "same as planned")
Implement and test the analytic system build by C4.5 decision tree algorithm.	Implement the system.	The system will be built on Hadoop platform. The algorithm to be used is C4.5.	Same as planned. In addition, the system can run with and without Hadoop.
	Train the system with training data to get the detection model.	The system will generate a decision tree model used for detecting the network attacks.	It is not specified what format the model should be. We use the XML file to store the model. The size of the model is small enough to be loaded into the main memory while the testing part is running.
	Test the system with testing data.	The testing component will print off the accuracy and response time to the testing data set. The satisfying accuracy will be over 90% and the response time should under 20 seconds.	The accuracy is 90.69%. The response time varies from 7 to 9 seconds. The false positive rate is also recorded which is 0.58%.
Team reflection on progress	Provide some comments below regarding the completion of this milestone specifically around: 1. How is the project progressing? 2. Are there any differences between projected and actual outputs/outcomes?		

The progress of the project:
The milestone 1 of this project has been completed. As the goal of implementation, we have built the system by using C4.5 decision tree algorithm on Hadoop platform and achieved the accuracy over 90% and response time below 10 seconds. We are now working on designing the second system. The milestone 2 is the implementation and test of the second one.
In the research perspective, we will determine which algorithm is more efficient for this system. We are going to use one more library, xgboost, to implement another system and compare the outcome of decision tree algorithm such as accuracy, detection rate, training time and response time.
The comparison of two systems will be the final part of this project. To eliminate disturbances, the same training and testing data will be applied to two systems that will both run on Hadoop platform.

Difference between projected and actual outcomes:
1. The false positive rate is introduced
In the draft plan, we would only record the accuracy and response time. The false positives indicate the number of normal vectors detected as attacks. The false positive rate is the proportion of false positives amongst all normal vectors. Our false positive rate is 0.58% which is satisfying. If the false positive rate is too high, say over 10%, it may mean this system is not reliable. The reasons could be in the implementation, the algorithms, the data sets, etc.
2. Detection time is much shorter than expected.
Our goal of response time was below 20 seconds, but our outcome shows the actual time is much shorter as 7 to 9 seconds. It is because we don't have previous knowledge of how short the response time could be. We will set 10 seconds as a baseline for the milestone 2.
3. Training time is longer with map-reduce
Training time with map-reduce is about 4 times longer than without it. The map-reduce class will store data on disk eventually while we can keep it in memory without it. There is a huge percent of time spent in reading and writing files on disk. Another reason is we run the program in Standalone mode instead of Fully-distributed mode so that there is no much improvement in computing time.

Team reflection on managing problems	Have you encountered any problems to date? If so, how have you managed them?
1. No proper source code available on the internet. At the very beginning, we wanted to find the source code implementing C4.5 on Hadoop from the internet, but none of the existing code is applicable. They are either not on Hadoop or cannot handle continuous data (They regard all data as discrete). So, we decided to implement the code ourselves. *2. Not familiar with java and map-reduce libraries.* When implementing the code, because we have little experience in Java language and map-reduce libraries. We spent quite a long time in learning them. We also got the book *Hadoop: The Definitive* Guide for help	*3. Training time is too long* The original C4.5 algorithm needs a huge amount of computation for finding the division point of a continuous attribute. We modified the algorithm to make it check some certain positions first like when the attribute and classification type change together. The training time is just about 1/8 of the original one. The error rate of the model from the modified algorithm is slightly higher about 100 in 500,000 which is 0.02%. *4. Virtual machine produces huge temporary files.* The virtual machine we run Hadoop on generates a few gigabyte temporary files every time that it will not delete automatically. We bought a new hard disk for the virtual machine and reset it regularly.

Supervisor assessment	Please, rate your team (1) effort, (2) project progress and (3) their self-reflection for milestone 1 Rating scale 1-10 as per standard marking scheme, i.e. 5 is a Pass and 7 is a credit. Add some comments to explain your rating
Effort: Progress: Reflection:	