Group 8 - Black Box Testing Summary

Project Name		POSTER: Vulnerability Discovery with Function Representation Learning from Unlabeled Projects				Project No.		G8-01	
Unit Name		Performance Testing							
Testing No.	Function Name	Testing Procedure	Input Size	Expect Output	Actual Output	Result	Tester	Comment	Date
1	ProcessCFilesWithCodeSensor.py	1. Set up a .c file input dataset and store it in one directory. 2. Set up a file directory for storing the output of function 3. Set up the PATH in the code and start executing the code	1514 .c files	1514 preprocess AST files	NO OUTPUT	FAILED	TIANLEI QI	Too many files will result Codesensor failed running. During executing, the memory will full and execution stop. The error message shows that the memory is too small for this operation to complete	2021/6/1
2	ProcessCFilesWithCodeSensor.py	1. Set up a .c file input dataset and store it in one directory. 2. Set up a file directory for storing the output of function 3. Set up the PATH in the code and start executing the code	6000 .c files	6000 preprocess AST files	6000 preprocess AST files	PASSED	FAN ZHANG	A larger memory size machine will finish the code without error. 16GB memory will be recommanded for the function.	2021/6/2
3	ProcessRawASTs_DFT.py	1. Set up the PATH in the code 2.Set up a file directory for storing the output of function 3.Start executing the code	6000 preprocess AST files	One .txt file contains processed DST. The DST are come from the 6000 source codes' ASTs	One .txt file contains processed DST. The DST are come from the 6000 source codes' ASTs	PASSED	TIANLEI QI	Finished without error. Only take few seconds to finish executing.	2021/6/3
4	BlurProjectSpecific.py	1. Set up the PATH in the code 2.Set up a file directory for storing the output of function 3.Start executing the code	One .txt file contains processed DST. The DST are come from the 6000 source codes' ASTs	One .csv file contains the function name and numerical represetations	One .csv file contains the function names and numerical represetations	PASSED	TIANLEI QI	Finished without error. Only take few seconds to finish executing.	2021/6/3
5	LSTM.py	1. Set up the PATH in the code 2.Set up a file directory for storing the output of function 3.Start executing the code	One .csv file contains the function names and numerical represetations	One Result_probabilities.csv file. One test_id.csv file. One test_label.csv file	One Result_probabilities.csv file. One test_id.csv file. One test_label.csv file	PASSED	TIANLEI QI	Finished without error. Total runtime 3408 seconds. CPU utilization rate 50-60% in average.	2021/6/3
6	RandomForest.py	1. Set up the PATH in the code 2.Set up a file directory for storing the output of function 3.Start executing the code	One test_cm.csv contains testing samples code metric extract by Understand. One test_id.csv contains test samples function names. One train_cm.csv contains training samples code metric extract by Understand. One train_id.csv contains training samples function names.	One Result_probabilities.csv file. One test_id.csv file. One test_label.csv file	One Result_probabilities.csv file. One test_id.csv file. One test_label.csv file	PASSED	TIANLEI QI	Finished without error. Total runtime 4330 seconds. CPU utilization rate 100% during executing.	2021/6/3
7	LSTM.py	1. Set up the PATH in the code 2.Set up a file directory for storing the output of function 3.Start executing the code	One .csv file contains the function names and numerical represetations	One Result_probabilities.csv file. One test_id.csv file. One test_label.csv file	One Result_probabilities.csv file. One test_id.csv file. One test_label.csv file	PASSED	FAN ZHANG	Finished without error. Total runtime 3173 seconds. CPU utilization rate 50-60% in average.	2021/6/4
8	RandomForest.py	1. Set up the PATH in the code 2.Set up a file directory for storing the output of function 3.Start executing the code	One test_cm.csv contains testing samples code metric extract by Understand. One test_id.csv contains test samples function names. One train_cm.csv contains training samples code metric extract by Understand. One train_id.csv contains training samples function names.	One Result_probabilities.csv file. One test_id.csv file. One test_label.csv file	One Result_probabilities.csv file. One test_id.csv file. One test_label.csv file	PASSED	FAN ZHANG	Finished without error. Total runtime 3678 seconds. CPU utilization rate 100% during executing.	2021/6/4

Enviornment Testing

Project Name		POSTER: Vulnerability Discovery with Function Representation Learning from Unlabeled Projects				Project No.		G8-01	
Unit Name		Enviornment Testing							
Testing No.	Function Name	Testing Procedure	Input Size	Expect Output	Actual Output	Result	Tester	Comment	Date
1	Windows	1. Set up a Windows enviornment 2. Clone our GitHub Repository on Windows Machine 3. Run ProcessCFilesWithCodeSensor.py , ProcessRawASTs_DFT.py , BlurProjectSpecific.py , LSTM.py , RandomForest.py in sequence	6000 .c files for LSTM .csv files code metrics from Understand for Random Rorest	Finish executing without error.	Finished executing without error.	PASSED	TIANLEI QI	All codes and functions can operate on Windows	2021/6/4
2	MAC OS	1. Set up a Windows enviornment 2. Clone our GitHub Repository on Windows Machine 3. Run ProcessCFilesWithCodeSensor.py , ProcessRawASTs_DFT.py , BlurProjectSpecific.py , LSTM.py , RandomForest.py in sequence	6000 .c files for LSTM .csv files code metrics from Understand for Random Rorest	Finish executing without error.	Finished executing without error.	PASSED	FAN ZHANG	All codes and functions can operate on MAC OS	2021/6/4