For Research Question 1   
  
  
a) (Replication Study)  
  
  
Steps to be performed   
  
Replication the existing ( Analysis) Do the same for 4 Datasets..   
  
1) Transfer the existing files (both the input yaml file and Api responses and their output decl file and data trace file) into a folder called existing..  
2) Now Run the beet and again copy the api response file , yanl file , newly generated trace file and decl file into a folder called current..   
3)Compare both the decl file and data trace files of current folder and existing   
4) Generate the invariants for the exisinting data trace files and paste in the existing folder   
5) Generate the invariants for the current data trace files and paste in the current folder   
6) Comapare the invariant that are generated ..   
7)Check to see if there are any difference..  
8) If there are any difference , report the findings,,

b) Extension of the dataset   
  
2) use the RESTest framework to generate API reponse for any particular API. (prefereable the same API chosen for the above step)  
  
Choose the API while coming to this point..   
  
1) Read the RESTest git hub api documentation.   
2) Change the configuration of the RESTest to set to API response as 15,000 responses   
3) paste the 15000 responses into a new folder .  
4) Run the beet .. save the output in the same folder   
5) Run the daikon   
6) Analyze the daikon output (invariants) with that of the both 10 k and 50 responses ..  
7) If there are differences in the detection of the invarints , please analyse the difference   
8) Report the finding.

Results   
  
For the Resaearch question 1   
  
Here are the findings   
  
  
Original Result   
  
Comparison Of the Existing Data With the Current data to monitor the accuracy of the results mentioned in the paper. For this We have taken 3 API ( Yelp,Marvel,Youtube) with 100, 1000, 10k api responses   
  
Here are the Results

|  |  |  |  |
| --- | --- | --- | --- |
| Existing\_Daikon\_modified\_Results | | | |
|  | 100 | 1000 | 10000 |
| Yelp | 147 | 147 | 142 |
| Marvel | 133 | 112 | 102 |
| Youtube | 254 | 254 | 252 |

|  |  |  |
| --- | --- | --- |
| Current\_Modified\_version\_Results | | |
| 100 | 1000 | 10000 |
| 55 | 42 | 146 |
| 109 | 95 | 94 |
| 120 | 128 | 131 |

The difference in these invariants between the current and existing is due to the fact , the difference in operating environments.  
  
Difference in configuration of the Daikon, Beet or any other tool used in the process would have result in different set of invariant.  
  
  
Extension of the Dataset Size   
  
We have taken the OMDB API with ByIdOrTitle as an endpoint for the extension of the dataset. We have used the customized RESTest framwork along with python code to generate **15000** API responses for the existing swagger file for the OMDB APIendpoint .  
  
Here is the table which represents the OMDB ByIdOrTitle invariants of different dataset sizes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 50 | 100 | 1000 | 10000 | 15000 |
| 11 | 10 | 9 | 12 | 15 |

The dataset was carefully produced , so as to increase a higher range of values including edge cases   
An example IMDb ID could be for the movie "Blade Runner," which has several versions, including the original 1982 release. Like this several edge case was considered for the data extension.

Similarly The youTube data api Yaml file and their corresponding responses are also tweaked so as to check if there are any increase In the invaraints. The test was carried out for 10k dataset size. Here is the response   
  
Here are the results.

|  |  |  |
| --- | --- | --- |
|  | Current\_Invariant | Modified\_Invariant |
| YouTube | 10000 | 10000 |
|  | 132 | 139 |

As part of extending the API coverage, a new API, named WeatherAPI, was introduced specifically to assess the arithmetic aspects of the API responses. A set of 150 API response records was generated for this purpose. From these records, AGORA identified 8 invariants. This testing phase aimed to scrutinize the generation of false positives by AGORA. Remarkably, during this evaluation, AGORA did not produce any false positive invariants.