**Machine Learning Techniques Applied To Detect Cyber Attacks On Web Applications**

In this paper author is describing concept to detect attack perform on Web Applications using Graph-based approach and Estimating dissimilarities between two components Needleman–Wunsch algorithm.

In graph based approach a graph will form using vertex (circle in graph) and edges are the line connection between two vertexes. Vertex will contains http request data which is coming from client to server, this http data will contains normal or attack data and by analyzing such data we can detect whether request is normal or attack.

All requests which are normal will have similarity and will be goes into same group by adding edges between those two similar http request and attacker will modify request data to perform some malicious behaviour and there will be not much similarity left (due to request data modification) which can indicate us that this request contains attack.

We can check similarity between two request data using Needleman–Wunsch algorithm.

To implement this project author is using CSIC dataset and below is the dataset example

**GET http://localhost:8080/tienda1/index.jsp HTTP/1.1**

**User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5; Linux) KHTML/3.5.8 (like Gecko)**

**Pragma: no-cache**

**Cache-control: no-cache**

**Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,\*/\*;q=0.5**

**Accept-Encoding: x-gzip, x-deflate, gzip, deflate**

**Accept-Charset: utf-8, utf-8;q=0.5, \*;q=0.5**

**Accept-Language: en**

**Host: localhost:8080**

**Cookie: JSESSIONID=1F767F17239C9B670A39E9B10C3825F4Connection: close**

Above is the normal request data in bold format and from above dataset just we need to look for http data (**GET** [**http://localhost:8080/tienda1/index.jsp HTTP/1.1**](http://localhost:8080/tienda1/index.jsp%20HTTP/1.1)**)** and we extract only http data from above dataset using REGULAR EXPRESSION concept. Below is request data which contains SQL Injection Attack

**GET http://localhost:8080/tienda1/publico/anadir.jsp?id=2&nombre=Jam%F3n+Ib%E9rico&precio=85&cantidad=%27%3B+DROP+TABLE+usuarios%3B+SELECT+\*+FROM+datos+WHERE+nombre+LIKE+%27%25&B1=A%F1adir+al+carrito HTTP/1.1**

**User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5; Linux) KHTML/3.5.8 (like Gecko)**

**Pragma: no-cache**

**Cache-control: no-cache**

**Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,\*/\*;q=0.5**

**Accept-Encoding: x-gzip, x-deflate, gzip, deflate**

**Accept-Charset: utf-8, utf-8;q=0.5, \*;q=0.5**

**Accept-Language: en**

**Host: localhost:8080**

**Cookie: JSESSIONID=B92A8B48B9008CD29F622A994E0F650D**

**Connection: close**

In above http request data we can see attacker is performing SQL Injection attack

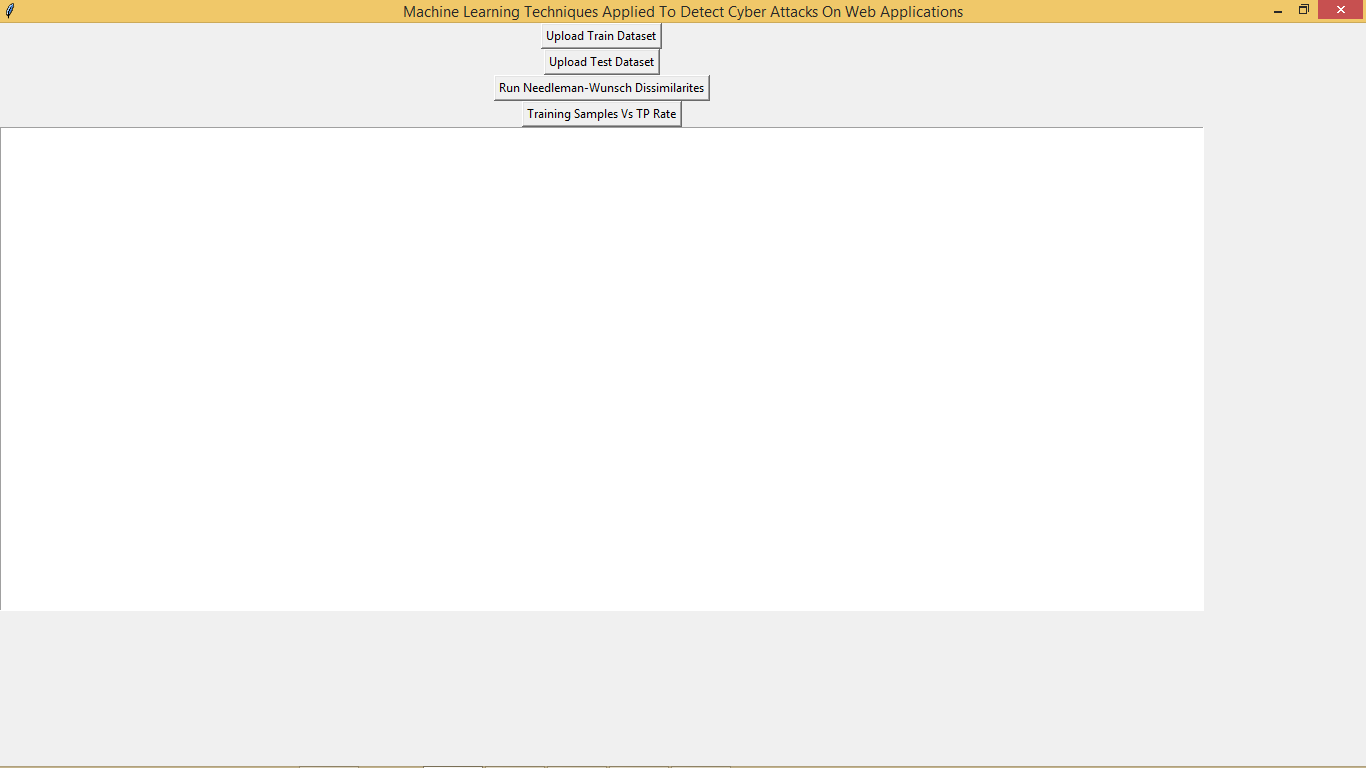
**http://localhost:8080/tienda1/publico/anadir.jsp?id=2&nombre=Jam%F3n+Ib%E9rico&precio=85&cantidad=%27%3B+DROP+TABLE+usuarios%3B+SELECT+\*+FROM+datos+WHERE+nombre+LIKE+%27%25&B1=A%F1adir+al+carrito HTTP/1.1**

For clarity you can see above request data in underline text attacker is trying to execute SQL Drop query. All normal requests may not contain that drop query due to which dissimilarity will occur to detect it as attack.

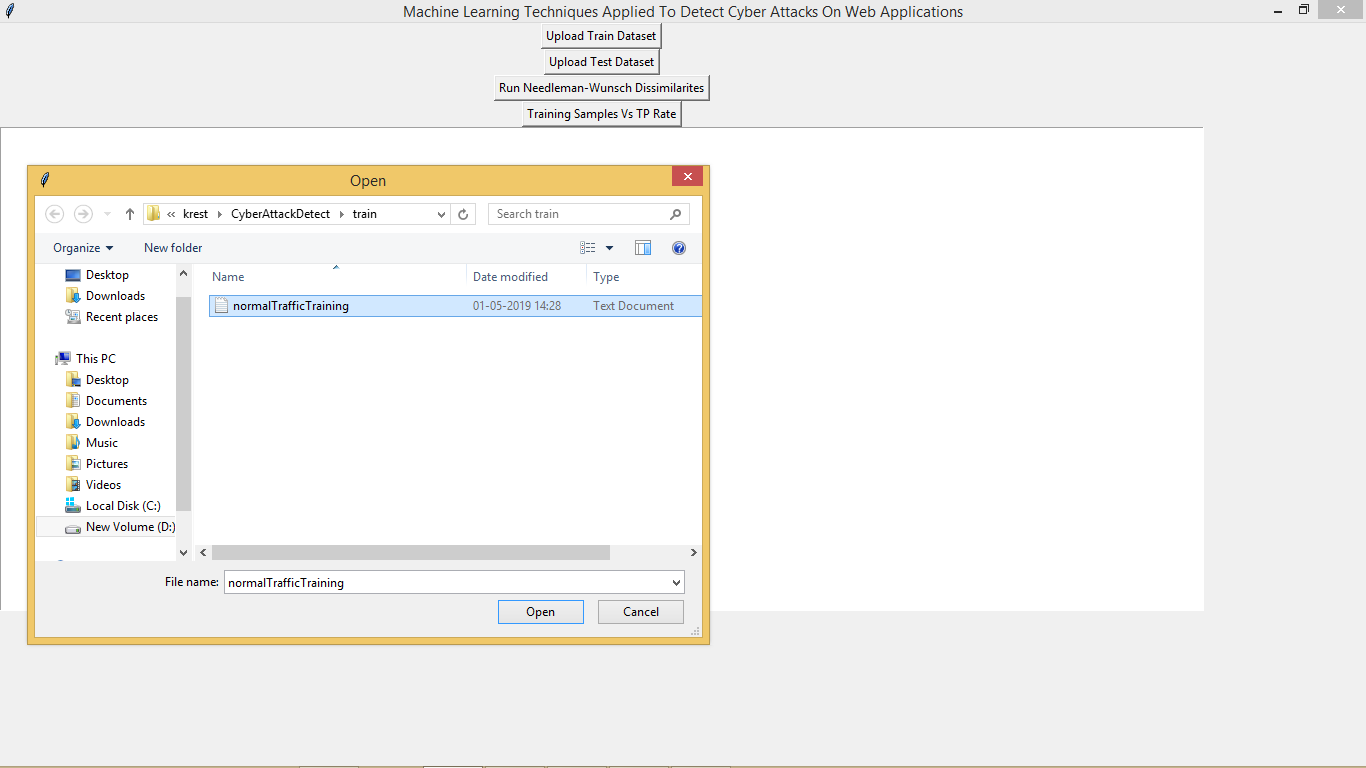
CSIC dataset comes in training and test data, training dataset contains all normal possible request data which can check with test (newly arrived request) data to detect it as normal or attack.

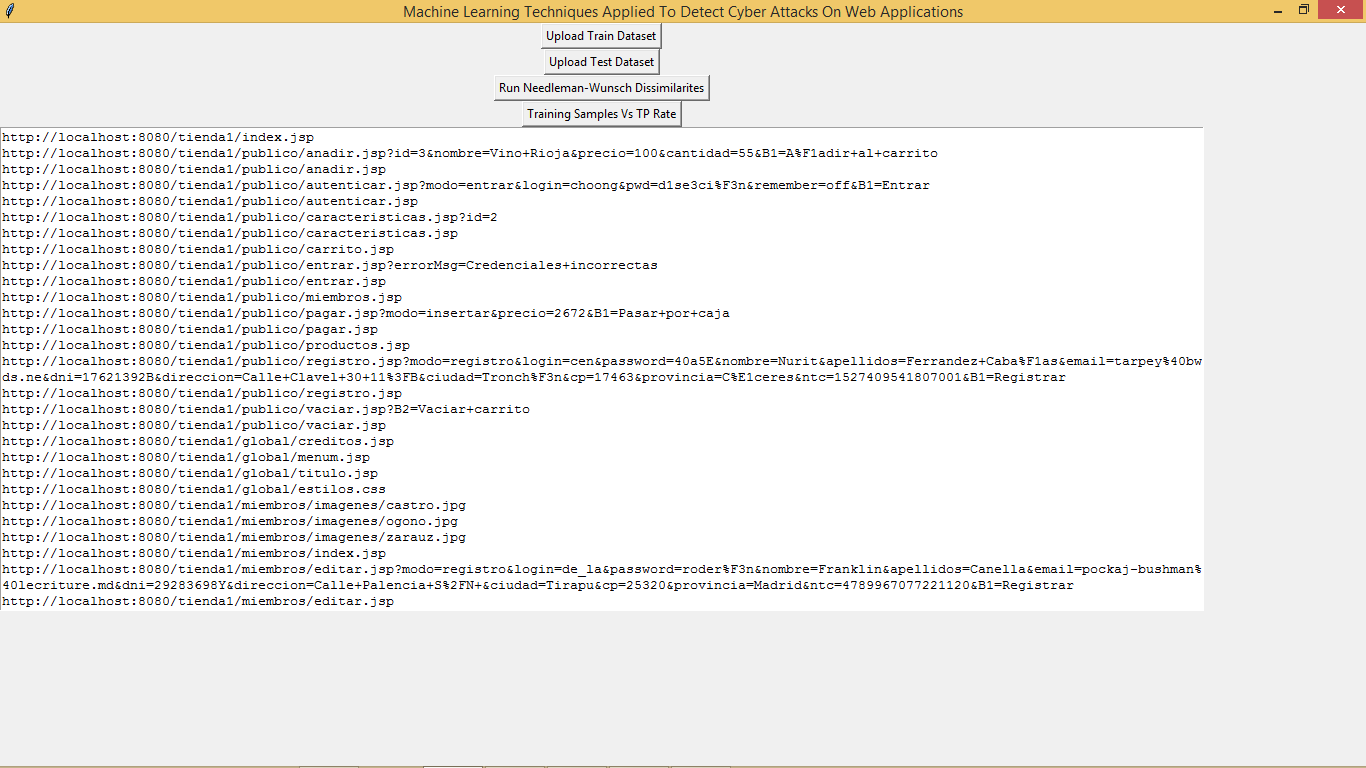
Screen shots

Double click on ‘run.bat’ file to get below screen

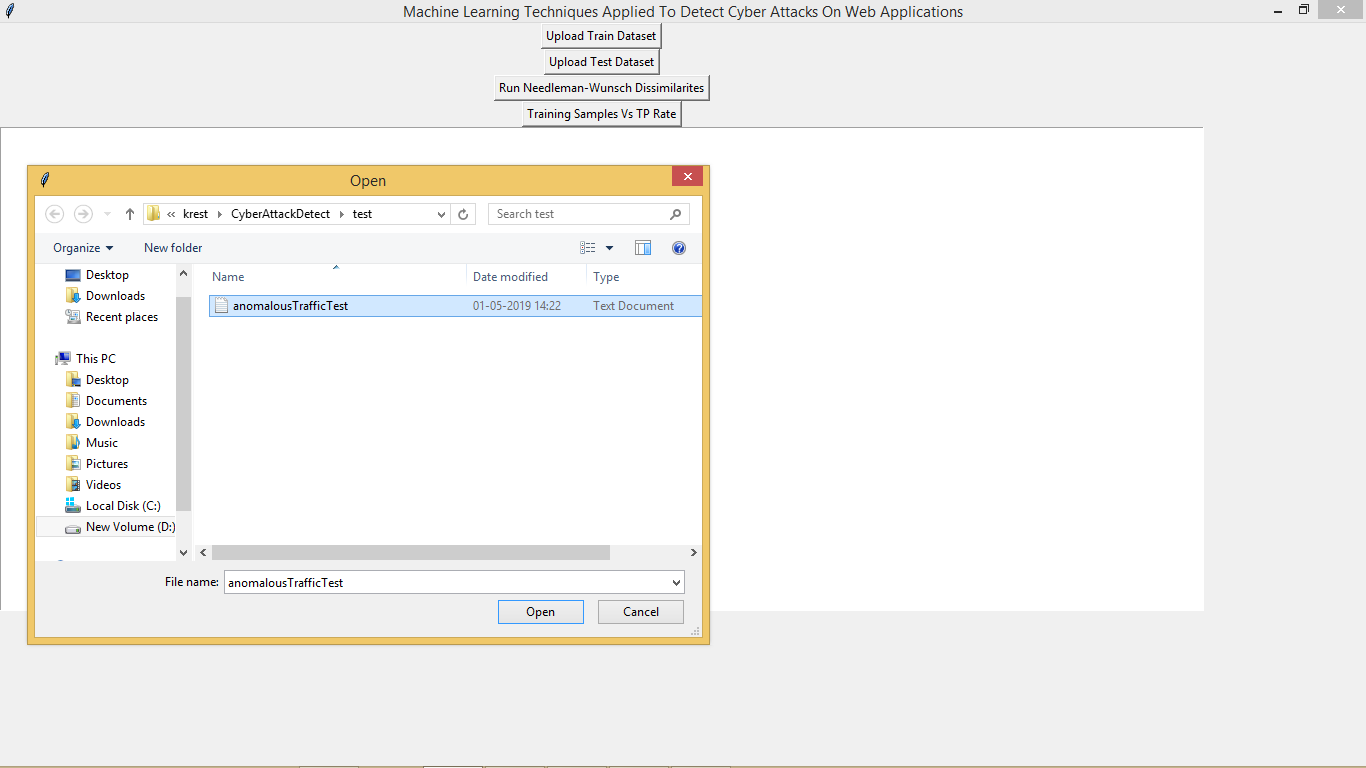


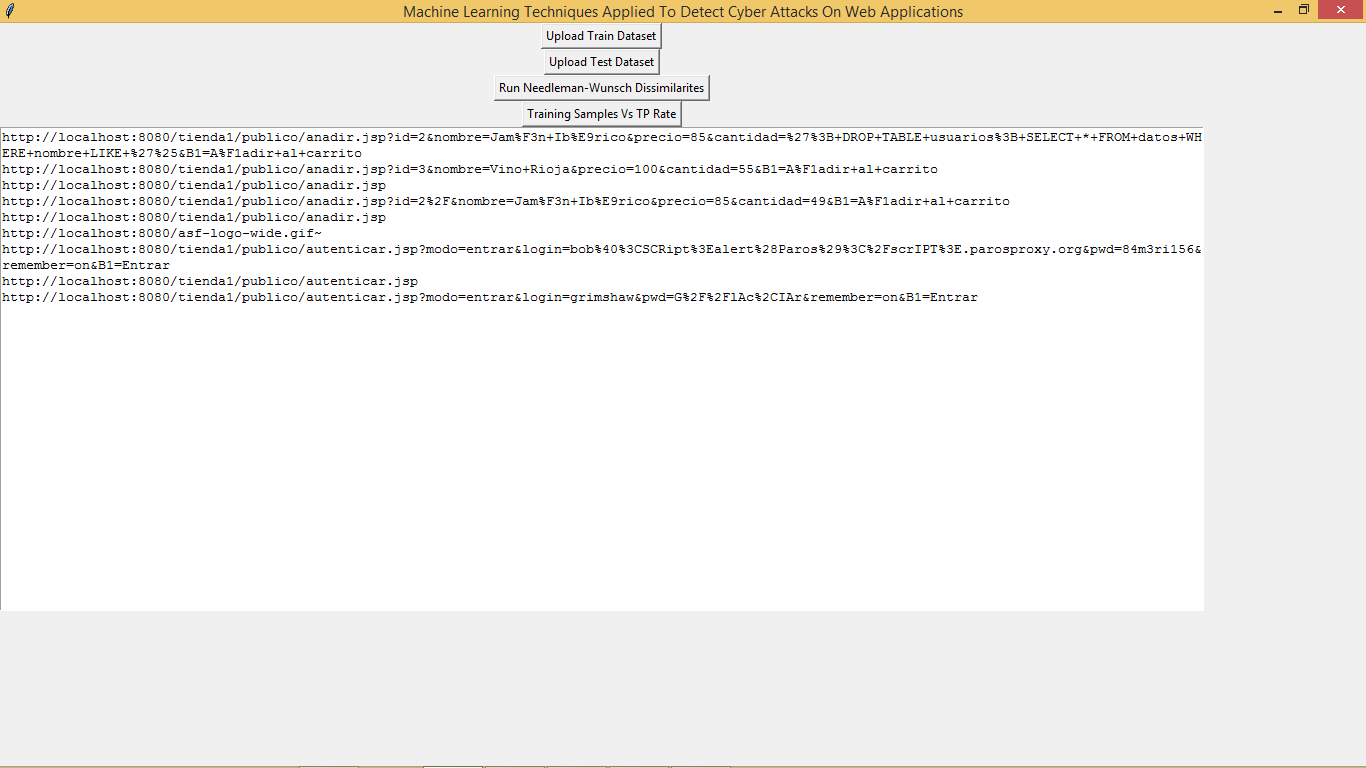
Click on ‘Upload Train Dataset’ button to upload normal training data



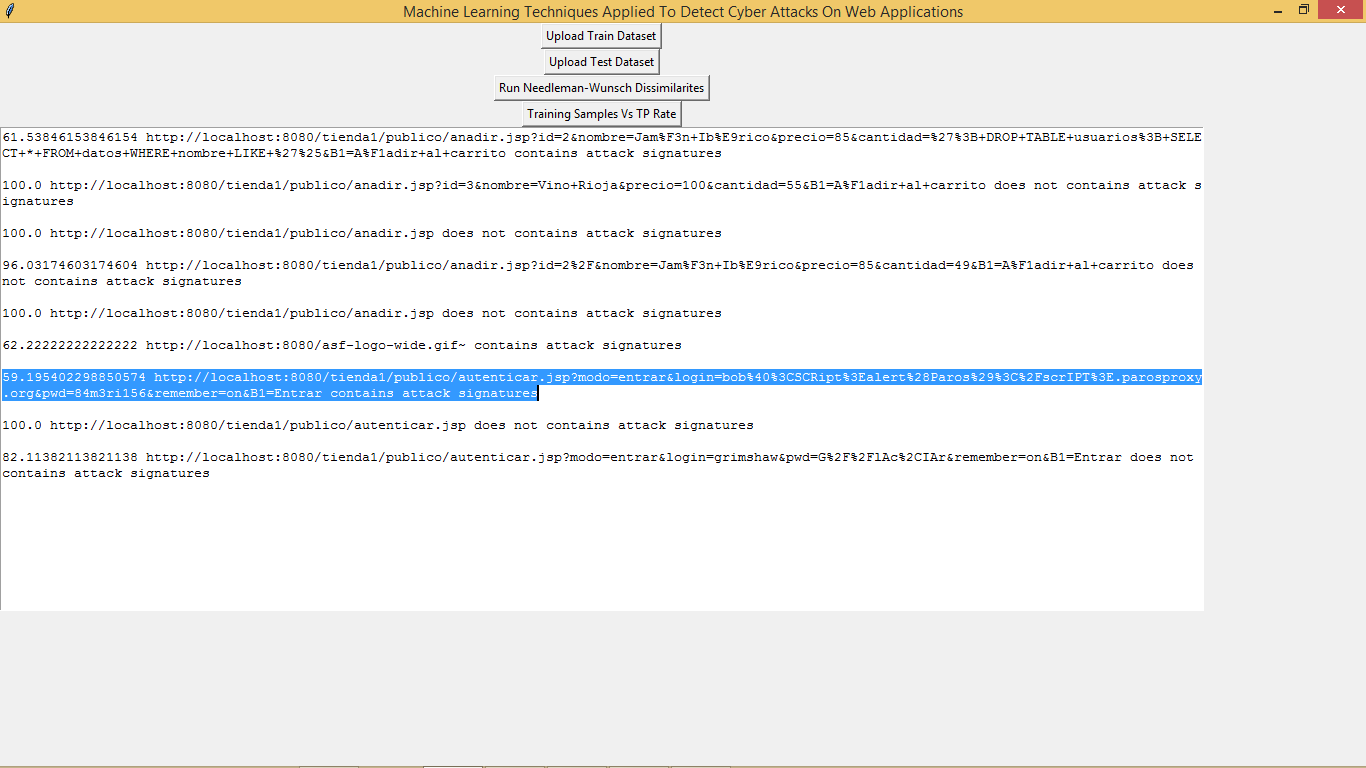


After uploading we can see only http request URL data is extracted using regular expression from training data and this will apply on test data to get result. Now upload test data





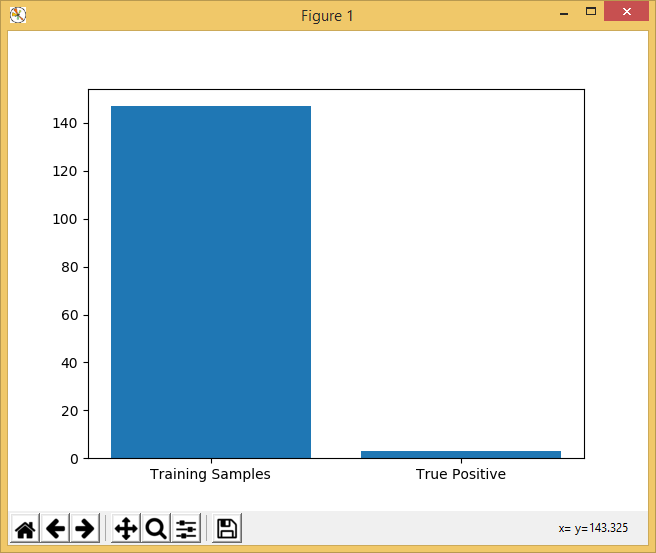
Above are some test request data, now click on ‘Run Needleman-Wunsch Dissimilarites’ button to check similarity between train and test request data



In above screen in selected text u can see first contains similarity score between train request data and test request data and then request data is displaying and then showing whether its normal or contains attack signatures.

**61.53846153846154 http://localhost:8080/tienda1/publico/anadir.jsp?id=2&nombre=Jam%F3n+Ib%E9rico&precio=85&cantidad=%27%3B+DROP+TABLE+usuarios%3B+SELECT+\*+FROM+datos+WHERE+nombre+LIKE+%27%25&B1=A%F1adir+al+carrito contains attack signatures**

For clarity u can in above bold data first 61.53 is the similarity score and then request data which contains SQL injection attack. Now click on ‘Training Samples Vs TP Rate’ button to get graph



In above graph x-axis contains total train dataset size and true positive detection rate and y-axis contains length