POLITECNICO DI TORINO

Master Degree in Computer and Communication Networks Engineering

Master of Science Thesis

Machine Learning for Network-Based Prediction of Web browsing QoE

Supervisors
Prof. Marco Mellia
Prof. Idilio Drago

Candidate
Shohreh Bashookian
Student ID: S213742

May 2018

ACKNOWLEDGEMENT

I would first like to thank my thesis supervisor Prof. Marco Mellia of the Department of Electronics and Telecommunications at Politecnico di Torino. He consistently allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it. I wish to thank my college supervisor Prof. Idilio Drago for his supervision at various stages during this project and for their kind interest in my work over several month. The door to Prof. Idilio Drago
Office was always open whenever I ran into a trouble spot or had a question about my research or writing.
Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.
Thank you.

ABSTRACT

There are many definitions for Quality of Experience (QoE). One of the simple and readable definitions of Quality of Experience is: A subjective measure of client's experiences. QoE describes the user observation and the resulting fulfillment of service performance in networks. Necessary services used by people for daily goals are being received on the web. Users are reaching these services mainly by web browsing. ISPs and Content Providers are able to serve their users with much better quality as the quality is the important factor for a user to choose among the services. Confirming a better QoE for web services has been a major research subject in these years.

There are many metrics that can be effective in the designation of Web QoE. According to other studies in this area, QoE measurement methods can be classified into two: Subjective Measurement, Objective Measurement. QoE measures the level of end-user satisfaction for a special service, it is a subjective determination, thus changes from user to user. Also, collecting data about QoE from the user is costly and time-consuming. On the other hand, the objective metrics as the basic index of Web browsing experience. Objective measurement method produces a model from the objective quality to the subjective quality. For example, Speed Index and OnLoad time are two common objective metrics. These metrics are measured by the web browser and are accessible only on the user side. The Speed Index is the average time at which visible portions of the page are displayed. Also, the OnLoad time is when your site is done loading everything local to your site (HTML, CSS, JavaScript code, images).

More recently people started proposing network-based metrics. These metrics could allow ISP to observe QoE too. For instance, the PAIN (Passive indicator) as a method to monitor web page performance using passive traffic logs at ISPs. It leverages passive flow-level and DNS transactions which are available in the network notwithstanding the deployment of HTTPS. PAIN automatically builds a model from the timeline of requests published by browsers to render web pages and uses it to analyze the web performance in real-time. They compare PAIN to indicators based on in-browser instrumentation and obtain strong relationships between the methods.

In this study, the purpose is to develop a methodology to create a system for ISPs to estimate the QoE for web pages for users working the web. We use Objective metrics and PAIN metrics to

predict the Speed Index. We design a machine learning to estimate the speed index and OnLoad time. For this, we use the dataset captured on PAIN (a Passive Web Speed Indicator for ISPs) work and extend the methodology to estimate directly the Speed Index from those traces. To create the dataset we visited 10 popular domain in Italy and for each domain download homepage and 9 internal pages, for total 100 URLs and 6948 visits of them. We obtain a HAR (HTTP Archive) file for each visited page. A HAR file recording HTTP requests in a JSON format and includes a variety of info. Several of the recorded info for each HTTP request are the URL, headers, cookies, request data, response, timing (speed index, OnLoad time, etc.). For the Machine Learning experiments, we use the Orange software. Orange is a machine learning application that used for the training set to build a predictor of Web QoE from the dataset. In this step, a regression should be used to estimate the Speed Index based on our features. Regression analysis is a way to realize that, when one of the independent variables is diverse and other independent variables are fixed, how the dependent variable changes. Regression analysis is widely used for prediction and using of regression has a significant overlap with the field of machine learning. To make a good model we installed orange and used different algorithms such as Linear Regression and Random Forest. Also, we set Speed Index as a target (dependent variable) and PAIN metrics, Round Trip Time, Number Of Servers, Average Bytes Out, Number Of Protocol, etc. as features (independent variables) in our prediction model.

We set up three experiments to validate the system: first, we did Optimistic validation that we used same data table (dataset) for both training and testing. Second, we used Cross Validation to see the performance in the case where we have samples of the sites for training, we used some samples of trace (select 90 percent of sites randomly) for training and some other sites (10 percent of the dataset) for testing, And third, we did a hard experiment which we used a sample of the trace for training (50 percent of our dataset include 5 popular Domains and their Subdomains) and a sample of the trace for testing (50 percent of our dataset include other 5 popular Domains and their subdomains). For appropriate predictions, it is important to check first the capability of these models. So we used R-Squared, MSE (Mean Square Error), RMSE (Root Square Error) MAE (Mean Absolute Error) and MAPE (Mean Absolute Percentage Error) to check the capability of models. RMSE shows how close the observed data points are to the model's predicted values. Lower values of RMSE shows a better fit. RMSE is a good measure of how exactly the model predicts the response, and it is the most important standard for fit if the main purpose of the model

is the prediction. Also, The MAPE is often used in practice because of it displays accuracy as a percentage of the error. Because this number is a percentage, it can be easier to understand than the other statistics.

As results, in the first experiment (Optimistic Validation) in the Random Forest model, R-squared is 0.92% and in Linear Regression model R-Squared is 0.79%. Estimated values are close to actual value, R-squared is close to one and both results are pretty good but the Random Forest model results are better. Also, other metrics in Random forest model achieved the smaller scores than linear regression. For example, MSE and RMSE measures for the Random forest (RMSE: 1141.203, MSE: 1302345) have better scores than Linear Regression and (RMSE: 2678.318, MSE: 7173389). Random Forest model has about 14% error and Linear Regression model has about 45% error so we can use Random Forest because its accuracy is more than Linear Regression. In the second experiment (Cross Validation) we use the Random Forest model, So R-squared is 0.83 %, RMSE: 1135.515, MSE: 1289.395 and Random Forest has about 37.91 % error. In the third experiment, in the Random Forest model, R-squared is 0.67 % and in Linear Regression model R-Squared is 0.56 %, the Random forest (RMSE: 3257.06, MSE: 10608437) have better scores than Linear Regression and (RMSE: 3736.244, MSE: 13959370) and Random Forest model has about 45.42 % error and Linear Regression model has about 52.09 % error, so we see that the third experiment result is worse than two previous experiment results but is acceptable.

In conclusion, we designed a Machine Learning model to estimate the QoE for Web Pages. The validation results have shown there is a high correlation between actual values and estimated values and the model that we designed is good for measure and estimate page load time and Speed Index using network traffic that is available in ISPs. As future work, we can use other network performance features to better estimate QoE of users while browsing Web Pages. Moreover, we will study whether large datasets could help improve estimations. Finally, our estimate could be used to reconfigure the network, thus improving QoE.

CONTENT

Tables and Figures

Chapter 1

Introduction

1.1 Scope

The first of all, it is necessary to define what QoE means. There are many definitions for QoE, one of the simple, readable and natural definition of QoE is: Quality of Experience is a subjective measure of client's experiences .Quality of Experience (QoE) describes the user observation and the resulting fulfillment of service performance in networks. Basic services which are used by very people for a daily goal are being received on the web for more accessibility and usability. Users are reaching these services mainly by web browsing.

ISPs and Content Providers are able to serve their users with much better quality which we know the quality can be the important factor for a user to choose among the services. QoE modeling and assessment is increasingly gaining attention among Internet Service Providers (ISPs) and operators. This growing interest can be explained in terms of the increased competition and the need for aggregated-value solutions, as well as by the risk of having churning clients for quality dissatisfaction.

Quality of Experience (QoE) for web pages are based on the HTTP protocol and accessed via a browser. There are many metrics that can be effective in the designation of Web QoE. Confirming a better QoE for web services has been one of the most major research subjects in these years. QoE measures the level of end-user satisfaction for a special service. It is a subjective determination, thus changes from user to user. Also, collecting data about QoE from the user is costly and time-consuming and is difficult to predict due to its subjective nature.

There are many factors that affect QoE, some are from technical character, but there are also environmental conditions that influence the perception. QoE is commonly referred to as a scalar value, mainly for the simplicity reasons.

However, some argue that it can be understood as a multidimensional value consisted of different aspects of quality [1]. There are many efforts that try to determine the aspects that contribute to

the perceived quality and try to develop objective measurements for those aspects. Mainly because of these reasons most of the work in the area of QoE has been focused on developing different objective methodologies for estimation of the quality values [1].

According to need ensuring the fulfillment of the existing users, Internet Service Providers (ISPs) need to make tools and methods that could measure and improve the user's satisfaction. Measurement of existing QoE is an important factor in understanding the current system's capabilities which can help to Internet Service Providers (ISPs) and system forecasting [31].

In this work we present an approach that uses Machine Learning (ML) technique to develop QoE prediction models which do not rely on training data from subjective studies, but is based on objective metrics and PAIN metrics. The objective metrics as the principal indicator of Web browsing experience. For example, Speed index and OnLoad time are two most common objective metrics.

As service conditions vary from one stream to the other our QoE prediction model learns more and becomes more complete and it's the accuracy improves. In addition this methodology provides for models that adapt to changes in the user preferences as well as to the introduction of new conditions in the environment such as new content and new terminal devices.

1.2 RELATED WORK

Measuring QoE is one of the more interesting subjects in these years.

According to other studies in this area, QoE measurement methods can be classified into three classes: Subjective Measurement, Objective Measurement, Network measurement

1.2.1 Subjective Measurement

The subjective measurement method is based on observation experiments that very reliable but is very difficult and costly method of measuring user' QoE. It has been studied for several years, providing researchers deeper perceptions of QoE subjective dimension. Most of the result of the subjective measurement analysis is the opinion score when the user is being served or has been served, and these scores are finally averaged into Mean Opinion Score (MOS) [3].

Due to a direct gain of data from the users, subjective measurement method results are very accurate, but this method is expensive and cannot be used to automation and real-time situation.

1.2.2 Objective Measurement

The objective measurement method is defined as using separately the measurement of objective quality to evaluate the subjective quality [4]. In other words, Objective measurement method produces a model from the objective quality to the subjective quality.

A variety of objective quality measurement and prediction models have been analyzed. Each model has its suitable scenarios and corresponding constraints. Convenient and tractability are an advantage of objective measurement methods also this method has the disadvantage of inaccuracy, i.e., the QoE received is only an estimation rather than an exact value for any user.

1.2.3 Network measurement (metrics)

This method includes Active and Passive Network Measurements. One of the important things for network operators is to know how well their network fulfills so that they know what kinds of services they are capable to present to their clients. ISPs are interested to transfer most amount of data at minimum amount of data at least costs. On the other hand, users generally wants the low delay and very low packet loss in end-to-end connections, also they prefer to have a contract with ISP that contain continuous connections with full bandwidth. For measuring efficiency, network operators use active or passive measurements to troubleshoot their network. The goal of network measurement is to see and measure what is happening in the network with different methods, techniques, and tools. These metrics are based on traffic metrics to estimate the objective metrics that have presented the subjective metrics. according to other studies, In passive network measurements, data is gathered by passively listening to network traffic for example by using(optical) link splitters or hubs to duplicate a link's traffic or by monitoring buffers in routers [28]. Based on the results, passive measurements have some advantages than active measurements. For example, they do not create extra traffic so they do not disturb the network and they can an exact presentation of the network traffic. One of the methods in this area that I used in my study is

the PAIN (Passive indicator) as a method to monitor web page performance using passive traffic logs [2] at ISPs.

Chapter 2
Background

The Internet Service Providers (ISPs) can use the QoE metrics to know how to improve their services and set the adequate pricing levels to optimize their economic returns as most users prefer affordable services that are priced fairly.

In practice, the QoE is measured with either the subjective or objective metrics. The goal of this studies to design a machine learning to predict the speed index which is one of important objective metrics to predict web performance in ISPS. So first we need to understand the meaning of Objective measurement and Objective metrics and then the concept of the PAIN Metrics obtained by other students which use in this studies.

2.1 Objective measurement

Unlike the subjective QoE metrics that directly evaluate the human perception, the objective QoE metrics utilize data, algorithms, and models to infer the user satisfactions. The data may be provided by applications or by the network protocol layers including the AQoS and NQoS measurements [8].

The objective modeling of system quality is attractive for its low implementation requirements, adaptively, and ability to operate in real-time settings, and it is used extensively by the network operators, codec engineers and the application developers.

Objective quality Evaluation aims to apply an automatic and reliable way to estimate a user's perception of a service. Its goal is to have a good correlation with subjective quality evaluation methods.

The objective QoE metrics can be divided into three classes:

1) Full Reference, which presents the highest accuracy, but it increases the non-data load [8]. In this method, both processed and reference data are available for detailed objective /subjective comparison.

2) No Reference, which may give low accuracy because network condition may affect its quality, however, it has no effect on networking load [8]. In this method, only processed data is used for objective/subjective comparison.

3)Reduced Reference, which promises a benefit over the first and second method as it represents the combination of advantages from first two methods such as higher accuracy but less non-data load.in this method, some features are extracted from reference and processed data are available to derive and compare objective and subjective correlation[8].

We know Metrics can help us find chances to improve performance. There are several metrics related to determining web pages performance and used to measure user experience like server time, render time, Onload Time… But there are some other metrics which help to achieve the more understanding of how users see web pages when they use different devices, browsers, and networks (3g, DSL, Cable). One of important Objective metrics is Speed Index

2.2 Speed Index Metric

WebpageTest is one of the most popular and free tools for measuring webpage performance [10]. Google attached Speed Index to its Webpage test for measuring the performance of different web pages in April 2012. The Speed Index is the average time at which visible parts of the page are displayed. It is expressed in milliseconds and dependent on size of the view port [9]. Speed Index measures how fast the user receives viewable content. WebpageTest captures video of the page loading Figure 1 [9]. Then start to check each frame to understand how many contents have been loaded. (10 frames per second in the current implementation and only works for tests where video capture is enabled [9]).

Speed Index is based on the percentage of the viewport and it is possible to evaluate websites between many devices. Therefore the Speed Index metric is one of important metrics for measuring a user's experience.

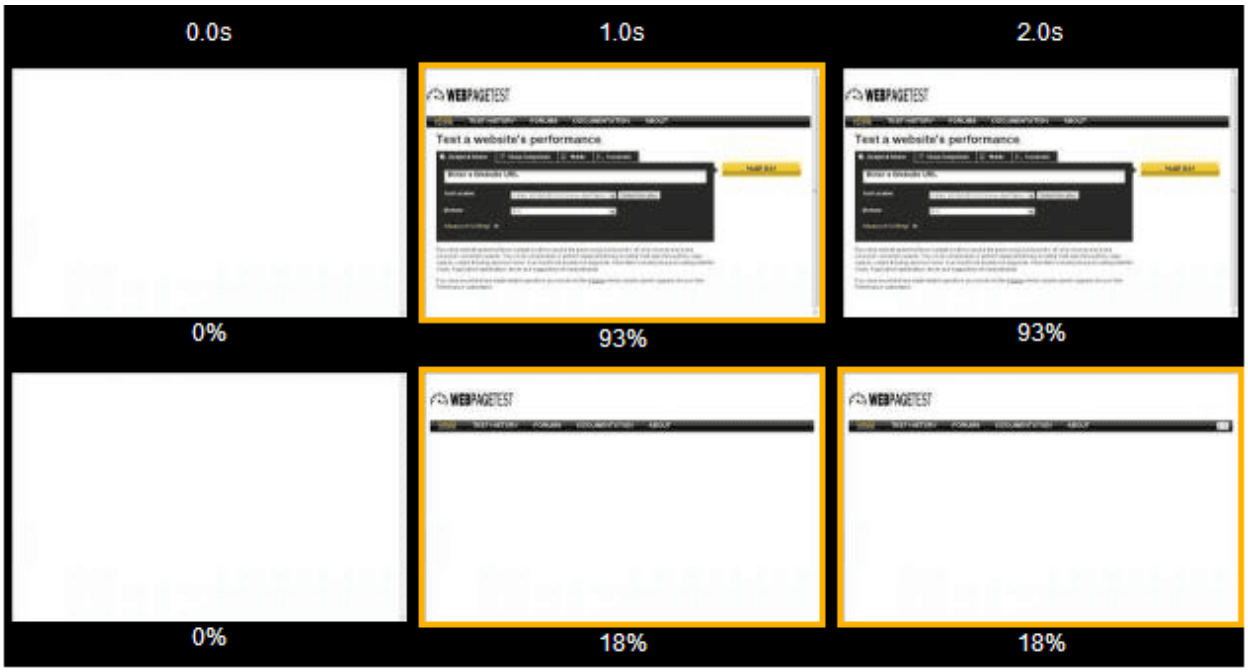

Fig1: WebpageTest captures video of the page loading

A lower Speed index is ideal as it means that large parts of a page render quickly. According to Figure 2 [9], pages with earlier render large visible elements (left picture of Fig2) receive better scores than pages with slowly render elements (right picture of Fig2) even when those pages have an equal visually complete measure.

Left Picture of Fig2 shows page starts rendering earlier, so large visible areas are completed soon and there is a good user experience but Right picture show page renders very late so the user sees an empty page and there is bad user experience.

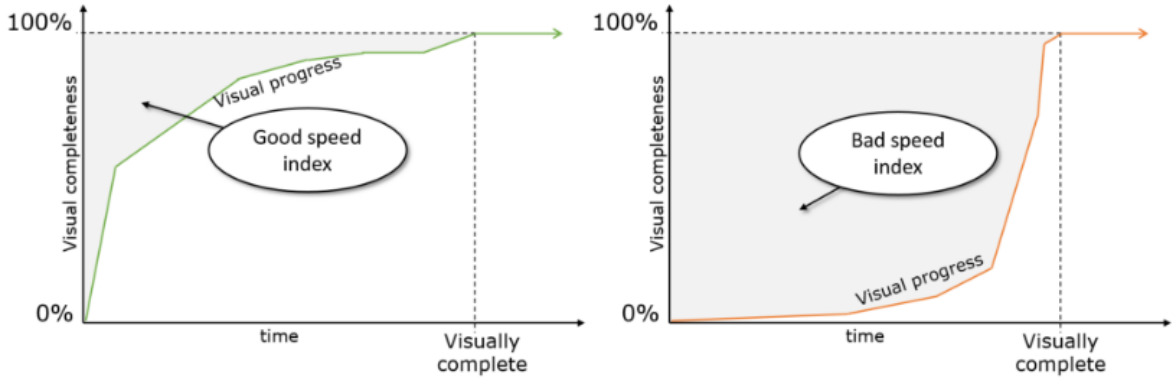

Fig2: example of two page rendering time

Calculate Speed Index:

Each frame is specified a score for visual incompleteness above the fold. The score is 0 percent for a blank screen and 100 percent for a visually complete page.

For calculating the score of each frame we can use this formula:

Interval Score = Interval time * (1.0 - (Completeness/100)

Where Completeness is the % visually complete for that frame and Interval is the elapsed time for that video frame in ms [9] .Finally add the score of frames together and final result is Speed Index score of the web page.

The example in below shows how we can calculate Speed Index. In this example, we reduce the number of frames for the model. In actual fact, we have to examine ten frames per second. Figure 3 shows how quickly the page in this example becomes visually complete [27].

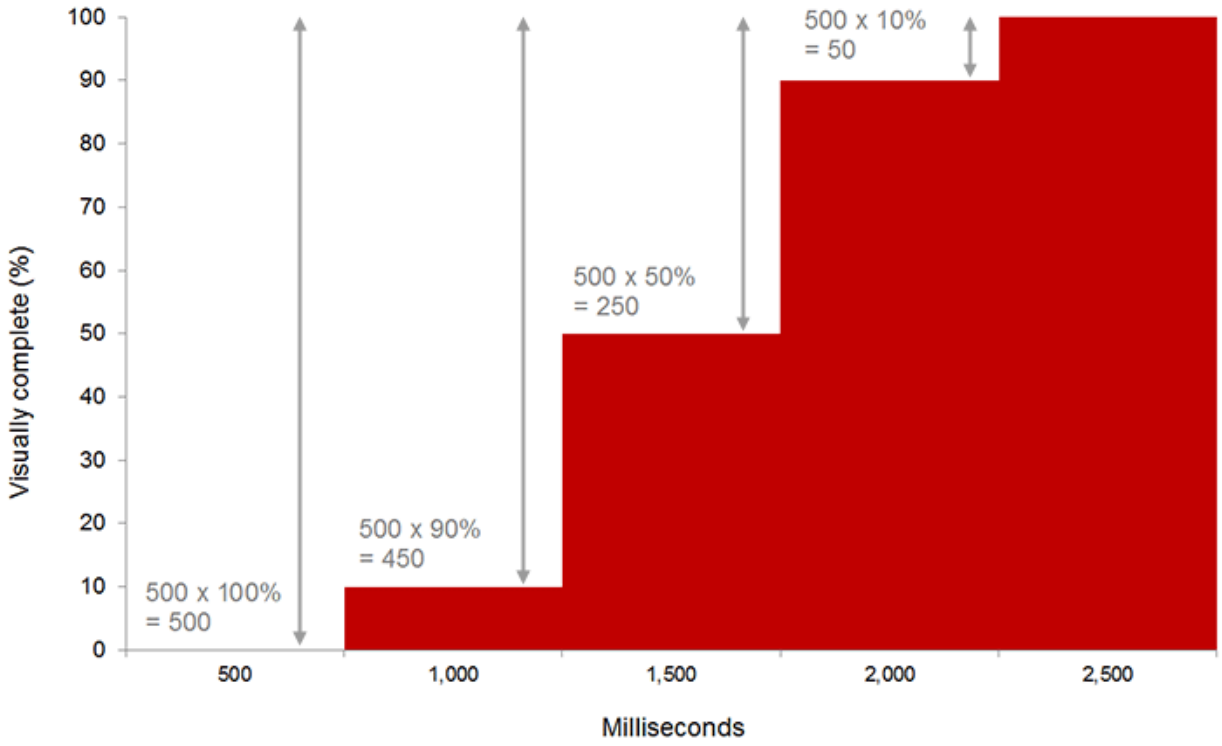

Fig3: example of how quickly completing the visible part of the page.

Frame 1 – 500 milliseconds

When the first frame is captured the page is blank at 500 milliseconds.

Since this is a blank page, it's 100 percent incomplete.

500 * 100% = 500

Frame 2 – 1,000 milliseconds

Alternative 500 milliseconds pass, and we capture another frame.

There is some content on the page now. We see 10 percent visually complete. It's 90 percent incomplete.

500 * 90% = 450

Frame 3 – 1,500 milliseconds

In the subsequent frame, after added 500 milliseconds, the page is 50 percent complete (So 50 percent incomplete):

500 * 50% = 250

Frame 4 – 2,000 milliseconds

The page is virtually done now, at 90 percent complete (10 percent incomplete):

500 x 10% = 50

Frame 5 – 2,500 milliseconds

The page is visually complete, so we do not add anything to the total score (500 *0% = 0).

Finally add the score of frames together and final result is Speed Index score of the web page:

500 + 450 + 250 + 50 = 1,250

2.3 PAIN (Passive Indicator) Metric

In this studies, we use PAIN metrics to build our datasets because pain metrics are strongly correlated with Objective metrics which we used to measure web page QoE. In this section, we describe PAIN Metrics to better understand their concept and their relationship with Objective Metrics.

PAIN (Passive indicator) is a system to monitor web page performance using passive traffic logs [2] at ISPs. PAIN relies only on L4-level statistics (source and destination IP address and TCP port numbers), annotated with DNS information to compute a synthetic indicator of the web page rendering time [2]. In simple words we can say, when clients open a website, to fetch data like HTML objects, media content, and scripts, the browser opens very flows to several servers. So we have Core Domain (for the first contacted server) and Support Domain (for other contacted servers).

The PAIN is a method that evaluates a performance index from the passive measurement. With given Core Domains of interest, PAIN automatically learns contacted Support Domains and builds models describing the typical order in which such flows as a performance indicator [2].

To build models of the website traffic, compute a performance index applying the models to new traffic, recognize checkpoints that model the download process and calculate the delay to transit checkpoints, PAIN uses visits the website from all clients.

PAIN is the unsupervised system that receives only the list of Core Domains (10 popular Domains which we use in this studies) to be monitored and use flow level measurements. It builds a model from traffic, flows automatically opened by browsers to regain images, videos, scripts etc., and does not need user's intervention or get some data from the user's side.

The other students calculated the PAIN metrics include 4 PAIN checkpoint times for all 10 popular Domains and their Sub Domains (totally 6948 visits) that we use in this study. Therefore, we used PAIN output which is a set of checkpoint times for each website visit [2] as one of the input files to create our dataset which we need to use in this study. Moreover, checkpoints representing Support domains that are usually contacted a long time after the Core visit [2]. They used 4

checkpoints for remaining experiments and they always take the arrival time of the last flow in each group as a checkpoint [2].

According to previous studies in this area, Pain checkpoints are strongly correlated with Objective metrics for different sites [2]. Results show that Pain acts as a proxy to quality monitoring and providing strong indications without user side instrumentation [2]. Therefore, as we said before we use the PAIN output to create our dataset which we need.

Chapter 3
Methodology

3.1 DATASETS

Our goal is to produce a model able to predict the quality of experience for web pages in ISPs. For this, we need to create CSV (comma-separated values) file as an input file of our machine learning. To make an input file, Webpage Test visit 10 popular domains in Italy and for each domain, Webpage Test downloads the homepage and 10 internal pages, so in total visits 100 URLs and export 6948 HAR files of these 100 URLs which we used to make our CSV file.

To simulate realistic network conditions and clients who are browsing web pages, we use Chrome and Firefox as our browser, PCs, Tablets and Smart Phone as our devices and 3G, DSL, Cable, e.g. for network emulation. Webpage Test publishes the HAR (HTTP Archive) file for each page which we visited, also Webpage Test contains several objective metrics which are used for QoE like Speed Index and Onload Time.

3.1.1 HAR File

HAR (HTTP Archive Viewer) is a JSON file that contains a record of the network traffic between client and server (Figure 4). It contains all the end to end HTTP requests/responses which are sent and received between the two network components. Some of the recorded information for each HTTP request are:

- The URL
- Headers
- Cookies
- Request data
- Response
- Timing

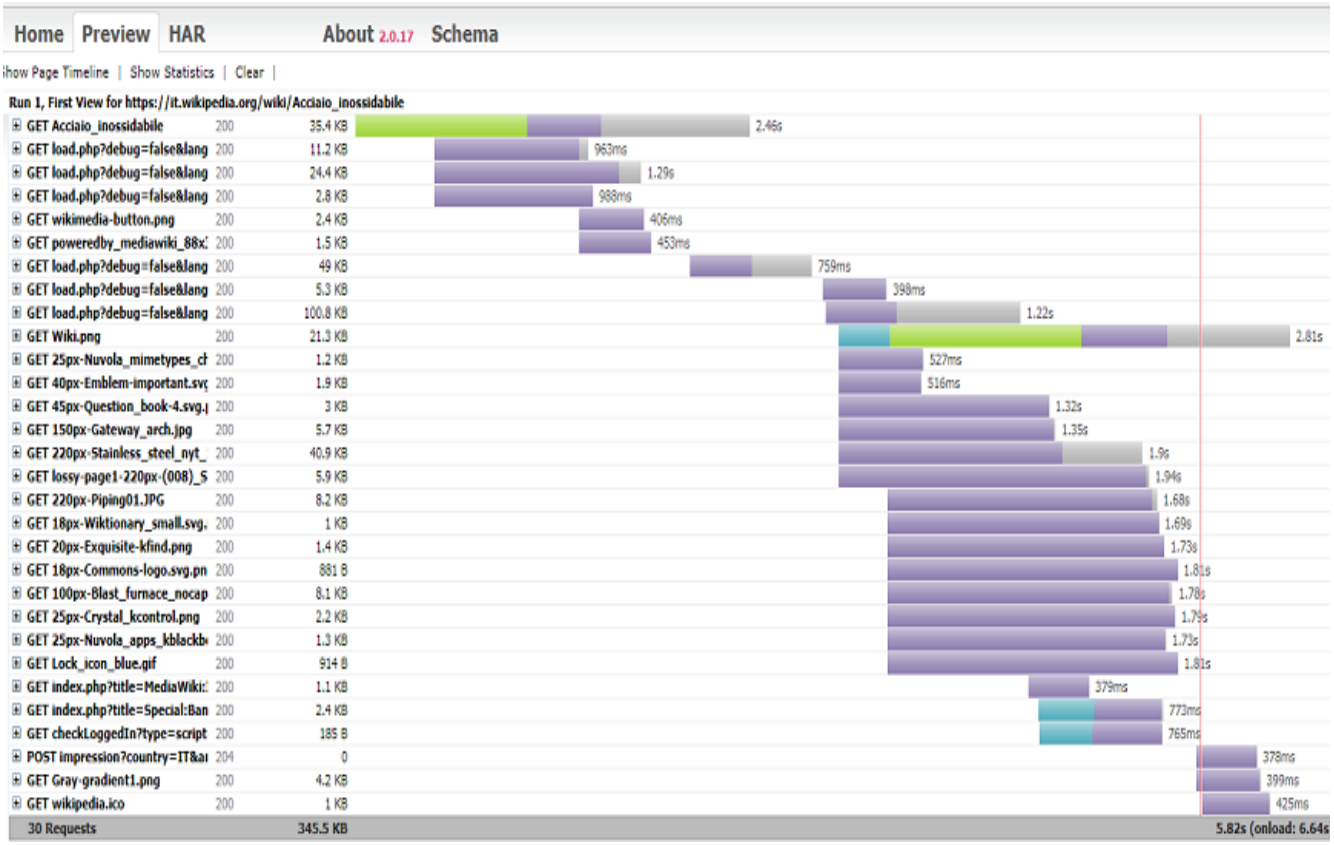

Fig4: **HAR (HTTP Archive Viewer) for one visited web page**

To build a dataset WebPageTest visits 11 popular Domain in Italy. The list of this domain shows in Table1 and we visit each web page in the typical network and artificial network condition and finally, we achieve to 6948 HAR file to build our Input File which used in machine learning.

	Domain
1	it.wikipedia.org
2	www.corriere.it
3	www.ebay.it
4	www.gazzetta.it
5	www.ilmeteo.it
6	www.lastampa.it
7	www.meteo.it
8	www.mymovies.it

9	www.repubblica.it
10	www.subito.it
11	www.wordreference.com

Table1:11 popular URLs used to build dataset

For typical network condition (typical dataset), the testbed connected through 1 Gbps Ethernet cable to the Politecnico di Torino network and use WebPageTest to visit 10 popular domain in Italy, for each domain WebPageTest download homepage and 9 internal page (Sub Domain), so in this part, we download details of 100 URL. We used 4 combinations for browsers and devices in our test which shows in Table2 and 8 network technologies which show in Table3. The important point is we visit each page two times for each step, one time with browser cache, and one time after few seconds for benefiting from caching.

Browser	Device	Operating System
Mozilla Firefox	PC	Windows 10
Google Chrome	PC	Windows 10
Google Chrome	Nexus 7	Android
Google Chrome	iPad Mini	IOS

Table2: Browsers and devices combination used in typical dataset

For artificial network condition (artificial dataset), We simulate the model that users open the web pages with the delay due to bad network conditions, so we should increase link delay or bandwidth limit on the testbed [11]. We simulate 10 cases: (1) adding from 100 ms to 500 ms extra delay and (2) imposing a limit from 5 Mbit/s down to 312.5 kbit/s on bandwidth [11].similar to a typical dataset in this part we visit each page two times for each step, one time with browser cache, and one time after few seconds for benefiting from caching again. Now we have 6948 HAR file as first data set which use to build final data set as input of our machine learning.

Name	Down Link	Up Link	RTT
Native	-	-	-
FIOS	20 Mbit/s	5 Mbit/s	4 ms
Cable	5 Mbit/s	1 Mbit/s	28 ms
DSL	1.5 Mbit/s	1 Mbit/s	50 ms
LTE	12 Mbit/s	12 Mbit/s	70 ms
3G Fast	1.6 Mbit/s	768 Kbit/s	150 ms
3G	1.6 Mbit/s	768 Kbit/s	200 ms
3G Slow	780 Kbit/s	330 Kbit/s	200 ms

Table3: 8Network technologies used in the typical dataset.
In Native case webPageTest enforces no shaping.

3.2 Feature Extraction

As it has been discussed in previous part we captured a HAR file for each web page and we know each HAR file contains several features (for example Figure 5 and Figure 6 shows features of a HAR file) which not all of them can help us to create a model to predict web pages QoE. After studying the features, selected some important features to be extract from HAR files.


```
{
    "log": {
        "browser": {
            "version": "56.0.2924.87",
            "name": "Google Chrome"
        },
        "version": "1.1",
        "entries": [
            {
                "_score_cache": "-1",
                "_socket": "28",
                "_request_id": "8",
                "startedDateTime": "2017-02-14T21:52:28.374+00:00",
                "_download_start": 1818,
                "_contentType": "text/html",
                "_index": 0,
                "_score_cookies": "-1",
                "_ip_addr": "91.198.174.192",
                "_is_secure": "1",
                "_load_ms": "762",
                "_server_rtt": "374",
                "_jpeg_scan_count": "0",
                "_host": "it.wikipedia.org",
                "_initiator_type": "other",
                "_cache_time": "-1",
                "_full_url": "https://it.wikipedia.org/wiki/Eriopsis",
                "_all_end": 2136,
                "_gzip_total": "10941",
                "_was_pushed": "0",
                "_server_count": "1",
                "_all_ms": 1825,
                "_contentEncoding": "gzip",
                "_dns_end": "304",
                "cache": {},
                "_ssl_ms": 684,
                "_connect_ms": 379,
                "_dns_ms": "-1",
                "_initiator_detail": "{\"type\":\"other\"}",
                "_http2_stream_exclusive": "1",
                "_score_compress": "-1",
                "_bytesOut": "237",
                "_minify_save": "0",
                "_image_save": "0",
                "_number": 1,
                "_score_etags": "-1",
                "_type": "3",
                "_minify_total": "0",
                "_url": "/wiki/Eriopsis",
                "_dns_start": "0",
                "_score_gzip": "100",
                "_score_keep-alive": "100",
                "_load_start": "1374",
```

```
"_load_start": "1374",
"_client_port": "55085",
"_connect_end": "683",
"_http2_stream_dependency": "0",
"_score_progressive_jpeg": -1,
"_responseCode": "200",
"_score_cdn": "-1",
"_protocol": "HTTP/2",
"_image_total": "0",
"_certificate_bytes": "3129",
"_cacheControl": "private, s-maxage=0, max-age=0, must-revalidate",
"response": {
    "status": 200,
    "cookies": [],
    "statusText": "",
    "content": {
        "mimeType": "text/html",
        "size": 10941
    },
    "headers": [
        {
            "name": "date",
            "value": "Tue, 14 Feb 2017 21:52:29 GMT"
        },
        {
            "name": "content-type",
            "value": "text/html; charset=UTF-8"
        },
        {
            "name": "content-length",
            "value": "10941"
        },
        {
            "name": "server",
            "value": "mw1239.eqiad.wmnet"
        },
        {
            "name": "x-powered-by",
            "value": "HHVM/3.12.7"
        },
        {
            "name": "vary",
            "value": "Accept-Encoding,Cookie,Authorization"
        },
        {
            "name": "x-ua-compatible",
            "value": "IE=Edge"
        },
        {
            "name": "content-language",
            "value": "it"
        },
```


Fig5: part of a HAR file


```
            "value": "bytes"
        }
    ],
    "headerssize": 1331,
    "redirectURL": "",
    "bodySize": 10941,
    "httpversion": "us:"
},
"_method": "GET",
"_download_end": 2136,
"_http2_stream_id": "1",
"_load_end": 2136,
"_score_minify": "-1",
"_http2_stream_weight": "256",
"_download_ms": 318,
"pageref": "page_1_0",
"request": {
    "cookies": [],
    "url": "https://it.wikipedia.org/wiki/Eriopsis",
    "queryString": [],
    "headers": [
        {
            "name": "upgrade-insecure-requests",
            "value": "1"
        },
        {
            "name": "user-agent",
            "value": "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36"
        },
        {
            "name": "accept",
            "value": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8"
        },
        {
            "name": "accept-encoding",
            "value": "gzip, deflate, sdch, br"
        },
        {
            "name": "accept-language",
            "value": "en-US,en;q=0.8"
        }
    ],
    "headerssize": 395,
    "httpversion": ": /",
    "method": "GET",
    "bodysize": -1
},
"_priority": "VeryHigh",
"timings": {
    "receive": 318,
    "send": 0,
    "ssl": 684,
```

```
    }
],
"pages": [
    {
        "_result": 0,
        "startedDateTime": "2017-02-14T21:52:27.000+00:00",
        "_minify_savings": 0,
        "_URL": "https://it.wikipedia.org/wiki/Eriopsis",
        "_server_rtt": 374,
        "_browser_other_private_memory_kb": 83264,
        "title": "Run 1, First View for https://it.wikipedia.org/wiki/Eriopsis",
        "_SpeedIndex": 3491,
        "pageTimings": {
            "onLoad": 5775,
            "_startRender": 3016,
            "onContentLoad": -1
        },
        "_responses_404": 0,
        "_score_compress": 100,
        "_effectiveBps": 71038,
        "_bytesOut": 7293,
        "_connections": 2,
        "_eventName": "Step 1",
        "_optimization_checked": 1,
        "_requests": 22,
        "_score_progressive_jpeg": -1,
        "_isResponsive": -1,
        "_effectiveBpsDoc": 76718,
        "_browser_name": "Google Chrome",
        "_requestsDoc": 22,
        "_render": 3016,
        "_domContentLoadedEventEnd": 2796,
        "_score_cache": 19,
        "_responses_other": 2,
        "_profile": "3G",
        "_docTime": 5775,
        "_score_minify": -1,
        "_gzip_total": 213898,
        "_run": 1,
        "_minify_total": 0,
        "_userTime.mwLoadStart": 2617,
        "_test_id": "170214_4D_2C",
        "_docCPUpct": 28,
        "_pageSpeedVersion": "1.9",
        "_browser_process_count": 10,
        "_gzip_savings": 0,
        "_browser_main_memory_kb": 71264,
        "_base_page_ttfb": 1818,
        "_loadEventEnd": 5748,
        "_bytesInDoc": 303575,
        "_score_keep-alive": 100,
        "_firstPaint": 2953,
```


Fig6: part of a HAR file

There is the list of features that have been selected in this study, in below:

1. Test ID
2. Speed Index
3. Onload Time
4. Number Of Objects
5. Protocol
6. Number Of Protocol
7. Total RTT (Round Trip Time)
8. Average ByteOut
9. Total ByteOut
10. Number Of Servers

OnLoad Time: It is the time when all elements of the web page (like images, videos, stylesheets, and scripts) have been downloaded.

Round-trip time (RTT): is the duration, measured in milliseconds, from when a browser sends a request to when it receives a response from a server [29]. It is one of the performance metrics when measuring page load time, Speed Index and network latency.

Protocol: The Internet relies on a number of protocols in order to function properly, a protocol is a standard for allowing the connection, communication, and data transfer between two places on a network [30].

1	ID	Speedindex	Onload	Number of Objects	Avg # of RTT	Avg ByteOuts	Total ByteOuts	Protocol	# Of Protocol	# Of Server
2	first_170210_05_1BN	2309	3819	22	187	170	3750	HTTP/2	22	13
3	first_170210_10_1BQ	1848	3701	18	188	186	3352	HTTP/2	18	14
4	first_170210_28_1BJ	2782	4179	30	239	143	4318	HTTP/2	30	14
5	first_170210_5S_1BP	1840	3678	17	187	191	3255	HTTP/2	17	14
6	first_170210_GH_1BG	2125	3428	13	187	199	2587	HTTP/2	13	12
7	first_170210_HW_1BK	2328	4069	69	206	109	7571	HTTP/2	69	14
8	first_170210_J1_1BF	2487	4321	55	188	115	6329	HTTP/2	55	13
9	first_170210_K0_1BH	2900	5605	51	188	121	6190	HTTP/2	51	14
10	first_170210_NH_1BM	2058	3888	22	188	165	3645	HTTP/2	22	14
11	first_170210_YK_1BR	1809	4225	51	189	117	6007	HTTP/2	51	14
12	first_170211_36_5R	3742	4039	17	187	206	3515	HTTP/2	17	10
13	first_170211_4W_5N	3379	4919	25	257	170	4251	HTTP/2	25	12
14	first_170211_DP_5S	6640	8207	23	189	171	3940	HTTP/2	23	10
15	first_170211_E3_5V	3328	4568	22	187	193	4249	HTTP/2	22	14
16	first_170211_E4_5X	3238	5262	19	187	205	3908	HTTP/2	19	14
17	first_170211_FM_5T	3477	4938	24	188	181	4354	HTTP/2	24	14
18	first_170211_GT_5Y	3655	4188	22	188	174	3837	HTTP/2	22	11
19	first_170211_M3_5W	2452	4680	17	188	220	3750	HTTP/2	17	13
20	first_170211_SF_5Q	2840	5461	44	190	130	5736	HTTP/2	44	10
21	first_170211_SJ_5P	3063	4568	9	187	247	2231	HTTP/2	9	8
22	second_170211_36_5R	1822	2183	5	188	338	1694	HTTP/2	5	3
23	second_170211_4W_5N	1515	2243	7	187	230	1611	HTTP/2	7	5
24	second_170211_DP_5S	1680	1469	4	188	265	1062	HTTP/2	4	3
25	second_170211_E3_5V	2061	2042	5	187	282	1411	HTTP/2	5	4
26	second_170211_E4_5X	1519	1448	5	191	265	1327	HTTP/2	5	4
27	second_170211_FM_5T	1704	1566	5	188	262	1314	HTTP/2	5	4

Fig 7: an example of CSV file obtained from 6948 HAR files

As we know HAR files can be exported by Firefox or Chrome. They include a JSON description of a sequence of HTTP requests including headers and request body. The code that was written first would take a HAR file and will extract Test ID of HAR file from the name of given HAR file. Then according to the body of HAR file which includes two part [log] [entries] and [log] [pages], and each parts are contained different features, we wrote a loop for each part to extract our features. For each HAR file, [log] [entries] part would calculate the average of Round Trip Time, total of ByteOut and average of ByteOute, the number of protocols, servers, and objects and extract the protocol that used. After [log] [pages] part would use a loop to extract Speed Index time and Onload time in each HAR file related to a web page. At the end, wrote a command to create a CSV file for saving each extract data on that.

3.2.1 Python code and scripts

A HAR file contains much valuable information for defining where you can improve website performance. HAR file information is stored in JSON format which means in order to visualize the information easier, tools such as the HAR Viewer can be used [12]. Inside the HAR file, there will be many timing components. To extract the defined features from the HAR file there is a need to parse and extract component of results from achieved HAR file. A python code wrote to do this on the way that results is split into parts, and features are extracted. The parsed file save as CSV (Comma separated values) file will have a column for each feature and a row for each HAR file.

```
import glob
import argparse
import json
import re
import sys
import csv
import os
filenames = glob.glob('test_*.har')
sys.stdout=open("out1.csv","a")
csvwriter = csv.writer(sys.stdout, delimiter=',', lineterminator='\n')
fields = ['ID', 'Speedindex', 'Onload' , 'Number of Objects' , 'Avg # of RTT' ,'Avg ByteOuts' , 'Total ByteOuts' , 'Protocol' ,'# Of Protocol' ,'# Of Server']
csvwriter.writerow(fields)
for file_name in filenames:
  filename, file_ext =os.path.splitext(file_name)
  splittedFileName = filename.split('_')
  if len(splittedFileName) <= 8:
      splittedFileNamelastPart = splittedFileName[7].split('.')
      splittedFileNameFull = splittedFileName[4] +  '_' + splittedFileName[5] + '_' + splittedFileName[6] + '_' + splittedFileNamelastPart[0]
  else:
      splittedFileNamelastPart = splittedFileName[7].split('.')
      splittedFileNameFull = splittedFileName[5] + '_' + splittedFileName[6] + '_' + splittedFileNamelastPart[0] + '_' + splittedFileName[8]
  FI = open(file_name)
  harfile_json = json.loads(FI.read())
  i = 0
  Objects= 0
  p = 0
  p1=0
  x =[]
  y =[]
  sum=0
```

Fig 8: Python script part1

As we said the outcome file is a CSV file which will be used for an input file of our machine learning model. A CSV is a comma separated values file, which allows data to be saved in a tabular format (numbers and text) [13]. The idea is that you can send out complex data from one application to a CSV file, and then import the data in that CSV file to another application. [13]. an example of the outcome of a parsed file as a CSV file can be seen in figure 7.

The python code is shown in the Figures 8 till 10.

```
b=0
sumbyteout=0
pl=0
serverlist=[]
for entry in harfile_json['log']['entries']:
    i = i + 1
    x = entry
    y = (len(x))
    Objects = Objects+1
    if (entry.get('_server_rtt','unknown') == 'unknown') or (entry.get('_server_rtt','unknown') == 'n') or (entry.get('_server_rtt','unknown') == ''):
        rtt = ''
    else:
        rtt =int(entry.get('_server_rtt','0'))
        p=p+1
        sum=sum+rtt
    if (entry.get('_bytesOut','unknown') == 'unknown') or (entry.get('_bytesOut','unknown') == 'n') or (entry.get('_bytesOut','unknown') == ''):
        byteout = ''
    else:
        byteout =int(entry.get('_bytesOut' ,'0'))
        b=b+1
        sumbyteout=sumbyteout+byteout
    if (entry.get('_protocol','unknown') == 'unknown') or (entry.get('_protocol','unknown') == 'n') or (entry.get('_protocol','unknown') == ''):
       protocol = ''
    else:
        protocol1 =(entry.get('_protocol' ,'0'))
        pl=pl+1
for entry in harfile_json['log']['entries']:
   i = i + 1
   items =(entry ['response']['headers'])
```

Fig 9: Python script part 2

```
    items =(entry ['response']['headers'])
    for x in items:
      i =i+1
      if (x['name'] == "server") or (x['name'] == "Server"):
          servername= x['value']
          if servername not in serverlist:
              serverlist.append(servername)
for entry in harfile_json['log']['pages']:
    i = i + 1
    speedindex =entry.get('_SpeedIndex', '')
    onloadTemp = entry.get( 'pageTimings','')
    onload = onloadTemp.get('onLoad', '')
    csvwriter.writerow([splittedFileNameFull, speedindex ,onload,Objects,int(sum/p),int(sumbyteout/b),sumbyteout ,protocol1 ,pl ,len(serverlist)])
FI.close()
```

Fig10: Python script part 3

3.3 Machine Learning

In statistical modeling, regression analysis is a set of statistical processes for estimating the relationships among variables, it includes many techniques for modeling and analyzing several variables when the focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors') [14]. Regression analysis is a way to realize that, when one of the independent variables is diverse and other independent variables are fixed, how the dependent variable changes. Regression analysis is generally worked on prediction and using of regression has a large overlap with the machine learning.

In this step, a regression should be used to predict the Speed Index based on our features because never used a machine learning to predicting Speed Index and we know that Speed Index is an important metric which helps to predict QoE of web pages in ISPs. To make a good model we installed orange and used a different algorithms such as Linear Regression and Random Forest as will be discussed below.

3.3.1 Linear Regression

Linear regression is a linear approach that generates an equation that describes the relationship between one or more predictor variables and the response variable. The case of one explanatory

variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression [14]. Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications, this is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine [14].

In simple linear regression, we predict scores on one variable from the scores on a second variable. The variable we are predicting is called the criterion variable and the variable we are basing our predictions on is called the predictor variable. Multiple linear regression attempts to model the relationship between two or more independent variables and a dependent variable by fitting a linear equation to observed data. Every value of the independent variable x is associated with a value of the dependent variable y.

Linear regression includes finding the best-fitting straight line through the points. The best-fitting line is called a regression line. The black diagonal line in Figure 11 is the regression line and includes the predicted score on a dependent variable (Y) for each possible value of the independent variable (X) [14]. The vertical lines from the points to the regression line represent the errors of prediction. As you can see in the figure 11 [14], the red point is very near the regression line so its error of prediction is small. Against, the orange point is much higher than the regression line and therefore its error of prediction is large.

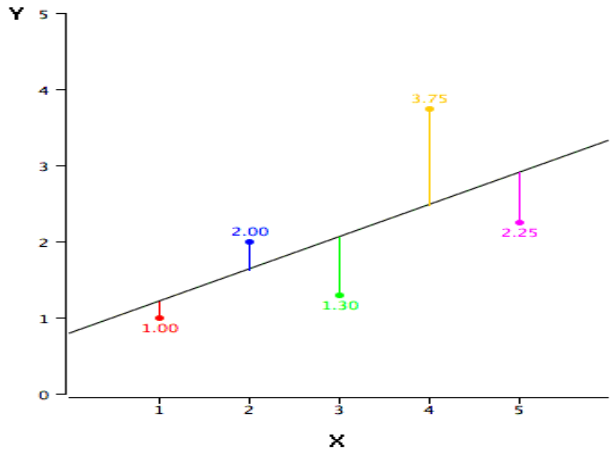

Fig 11: In the scatter plot, the black line consists of the predictions,
The points are the actual data, and the vertical lines between the points and the black line represent errors of prediction [14]

The model for multiple linear regression that relates a y-variable to p-1 x-variables is written as

$$Y_i=\beta_0 +\beta_1 X_{i,1}+\beta_2 X_{i,2}+\ldots+\beta_{p-1} X_{i,p-1}+ \varepsilon_i \text{ [18]}$$

We assume that the ε_i have a normal distribution with mean 0 and constant variance σ^2. The subscript $\boldsymbol{i}$ refers to the i^{th} individual or unit in the multiple linear regression. In the symbolization for the *x*-variables, the subscript following $\boldsymbol{i}$ simply represents which *x*-variable it is. The name "linear" in "multiple linear regression" refers to the fact that the model is linear in the parameters, β_0 , $\beta_1 \ldots \beta_{p-1}$. This just means that each parameter increases an *x*-variable, while the regression function is a sum of these "parameter times *x*-variable" terms [18].

3.3.2 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees[19].

Decision trees are non-parametric models that perform a sequence of simple tests for each instance, traversing a binary tree data structure until a leaf node (decision) is reached [19].

Decision trees have some benefits:

- They are effective in both computation and memory usage during training and prediction.
- They can represent non-linear decision boundaries.
- They perform combined feature selection and classification and are strong in the presence of noisy features.

In a standard classification tree, the idea is to split the dataset based on similarity of data. A decision tree is made top-down from a root node and includes splitting the data into subsets that contain instances with similar values, Then again, in a regression tree, as the target variable is a real-valued number, we suitable a regression model to the target variable using each of the independent variables. Then for each independent variable, the data is split at several split points. We calculate Sum of Squared Error (SSE) at each split point between the predicted value and the actual values, the variable resulting in minimum SSE is selected for the node, then this process is recursively continued till the entire data is covered [19].

In the other words, we can use a Random forest for regression analysis and are in fact called Regression Forests. They are group of different regression trees. Each leaf contains a distribution for the continuous output variable/s. This regression model consists of an ensemble of decision trees, each tree in a regression decision forest outputs a Gaussian distribution as a prediction, an aggregation is performed over the ensemble of trees to find a Gaussian distribution closest to the combined distribution for all trees in the model [20].

3.4 Validation Metrics

We set up three experiments to validate the system: first, we did Optimistic validation that we used same data table (dataset) for both training and testing. Second, we used Cross Validation to see the performance in the case where we have samples of the sites for training, we used some samples of trace (select 90 percent of sites randomly) for training and some other sites (10 percent of the dataset) for testing, And third, we did a hard experiment which we used a sample of the trace for training (50 percent of our dataset include 5 popular Domains and their Subdomains) and a sample of the trace for testing (50 percent of our dataset include other 5 popular Domains and their subdomains).

We know that regression models are used for predictions. For appropriate predictions, it is important to check first the capability of these models. So we used R Squared , MSE (Mean Square error) , RMSE(Root Square Error) MAE(Mean Absolute Error) and MAPE(Mean Absolute Percentage Error) methods are used to check the capability of models.

3.4.1 R-squared

Is a statistical measure of how close the data are to the fitted regression line, it is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression [12].

High values of R-Squared represent a strong relationship between response and predictor variables while low values mean that developed regression model is not appropriate for required predictions.

The value of R is between 0 and 1, that 0 means no relationship between sample data and 1 mean exact linear relationship. (Figure 12) shows how we can calculate the R-Squared.

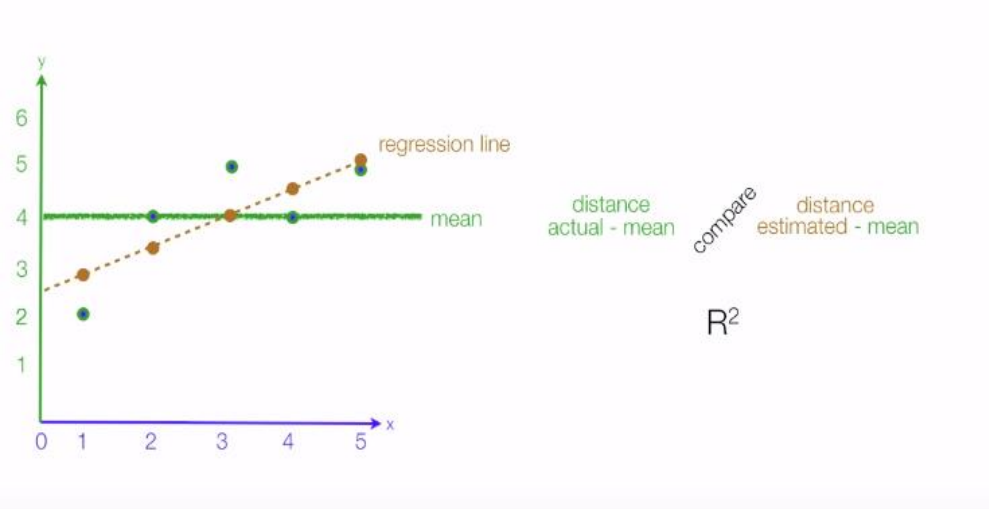

Fig 12: R-Squared Details

For calculate R Squared we used the following formula:

$$R^2 = \frac{SSR}{SST}$$

In this formula respectively SST (Sum of Squares of Total) and SSR (Sum of Squares of Regression) are the total sums of the squares and measures how far the data are from the mean and the sum of squares of errors and measures how far the data are from the model's predicted values.

SST = (actual value – mean actual value) ^2

SSR= (predicted value – mean actual value) ^2

According to our goal which is predict Speed Index we calculated SST and SSR based on Speed Index variable.

SST = (Speed Index value – Speed Index Mean) ^2

SSR = (predicted value – Speed index Mean) ^2

3.4.2 MSE (Mean Squared Error)

Measures the average of the squares of the errors, that is, the average squared difference between the estimated values and what is estimated [22]. The mean squared error tells us how close a regression line is to a set of points. It does this by taking the distances from the points to the

regression line (these distances are the "errors") and squaring them, the squaring is necessary to remove any negative signs.

The measure of mean squared error requires a target of prediction along with a predictor which is said to be the function of the given data. The mean squared error can be referred to the second moment of the error measured about the origin. It incorporates both the variance and bias of the estimator, if an estimator is an unbiased estimator, then its mean squared error is same as the variance of the estimator [22]. The unit of MSE is the same as the unit of measurement for the quantity which is being estimated [22].

For calculate MSE we used the following formula:

$$\mathbf{MSE} = \frac{\sum_{t=1}^{n}(A_t - F_t)^2}{n}$$

Which AT indicates the actual number, FT indicates the prediction number and n indicates number of observations.

The smaller the means squared error, the closer you are to finding the line of best fit but this number is related to the range of your values. Depending on your data, it may be impossible to get a very small value for the mean squared error because when we have 6948 different values so, the result is close to regression line but it is not small.

3.4.3 RMSE (Root Mean Squared Error)

It is just the square root of the mean square error [23], in the other word, it's the square root of the average of squared differences between prediction and actual observation. RMSE measures how much error there is between two data sets. The RMSE is thus the distance, on average, of a data point from the fitted line, measured along a vertical line. Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors [23]. This means the RMSE is most useful when large errors are particularly undesirable, the range of this metric can between from 0 to ∞.

For calculate RMSE we used the following formula:

$$\mathbf{RMSE} = \sqrt{\frac{\sum_{t=1}^{n}(A_t - F_t)^2}{n}}$$

Which AT indicates the actual number, FT indicates the prediction number and n indicates number of observations.

3.4.4 MAE (the Mean Absolute Error)

It is the sum of absolute differences between the actual value and predicted value, divided by the number of observation. In the other word, a quantity used to measure how close forecasts or predictions are to the eventual outcomes, as the name suggests, the mean absolute error is an average of the absolute errors [25].

For calculate RMSE we used the following formula:

$$\mathbf{MAE} = \frac{\sum_{t=1}^{n} | A_t - F_t |}{n}$$

Which AT indicates the actual number, FT indicates the prediction number and n indicates number of observations.

3.4.5 MAPE (Mean Absolute Percentage Error)

It is the average of absolute error divided by actual observation values. Is a measure of prediction accuracy of a forecasting method in statistics [25]. The MAPE is often used in practice because of its very great description in terms of relative error [26]. Displays accuracy as a percentage of the error. Because this number is a percentage, it can be easier to understand than the other statistics, we used MAPE as one of the quality measures for regression models.

For calculate RMSE we used the following formula:

$$\mathbf{MAPE} = \frac{\sum_{t=1}^{n} \left|\frac{A_t - F_t}{A_t}\right|}{n} \times 100$$

Which AT indicates the actual number, FT indicates the prediction number and n indicates number of observations.

3.5 Tools

3.5.1 Orange

Open source machine learning and data visualization for novice and expert, also
Interactive data analysis workflows with a large toolbox [15]. Orange is an open-source Data visualization, machine learning and data mining toolkit. It features a visual programming front-end for explorative data analysis and interactive data visualization, and can also be used as a Python library [16]. Orange is a component-based visual programming software package for data visualization, machine learning, data mining and data analysis [17]. Widgets offer basic functionalities such as reading the data, showing a data table, selecting features, training predictors, comparing learning algorithms, visualizing data elements, etc. [15]. The user can interactively explore visualizations or feed the selected subset into other widgets [16].

Chapter 3
Experimental Results

We want to predict Speed Index metric by using other features which is the main goal of this work, the idea is that we make a model for future which able to predict speed index based on other features that we have in ISPs. This model helps ISPs to predict the QoE for web pages. In Orange application, the first step was to design regression models, after gaining confidence on the models. Then used them on the real captured data.

3.1 Regression Design

To design regression model we used a captured data of visited 10 popular web pages in Italy and their sub domain which we saved these data in a CSV file as an input of our machine learning. At first to design a model we used all metrics in SCV file. As we said in the previous chapter we used Linear Regression and Random Forest models to achieve in our goal. We need to tests learning algorithms, we used a dataset and two learning algorithms and used Cross-validation to estimate performance. As outputs evaluation results we observed their performance in the table inside the Test&Score widget and in the scatter plot**.**

For appropriate predictions, it is important to check first the capability of these models. So we used R Squared, MSE (Mean Square Error), RMSE (Root Square Error) MAE (Mean Absolute Error) and MAPE (Mean Absolute Percentage Error) methods are used to check the capability of models.

The cross-validations (Figure 14) illustrates that the Random forest shows a better result than Linear Regression model. As we know, we have 6948 data and data range are very different and sometimes high so the result of MSE, RMSE, MAE shows us large numbers but they show correct numbers.

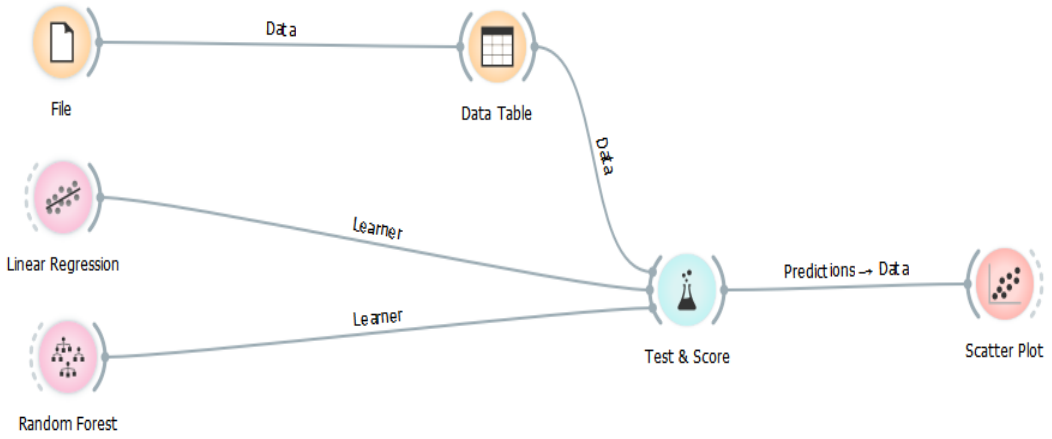

Fig 13: Orange Regression with the Random Forest algorithm and Linear Regression algorithm

Evaluation Results

Method	MSE	RMSE	MAE	R2
Random Forest	5393207.721	2322.328	1261.969	0.656
Linear Regression	7235808.908	2689.946	1591.712	0.539

Fig 14: Evaluation Results in Orange application for Random Forest and Linear Regression

3.2 Training and testing

Considering the fact that both regression models are able to use in prediction Speed Index we want to make a model to use in ISPs, so we need to know which model is better and give us the best result. So the our data table (Datasets) is include Speed Index metrics as Target variable and 10 independent metrics include Round Trip Time, Number Of Objects, Average ByteOuts, Number of Protocols, Number of Servers and four PAIN metrics (Checkpoint1, Checkpoint2, Checkpoint3, Checkpoint4) . To save the predictions result we use a CSV format and use this csv file to calculate MSE, RMSE, R Squared and MAE to use to compare both regression model and find the best one. We calculated all method according to their formula which explained in the previous chapter.

So we set up three experiments to validate the system:

1) Optimistic Validation

In this design we use same data table (dataset) for training and testing, Figure 15 showed our design.

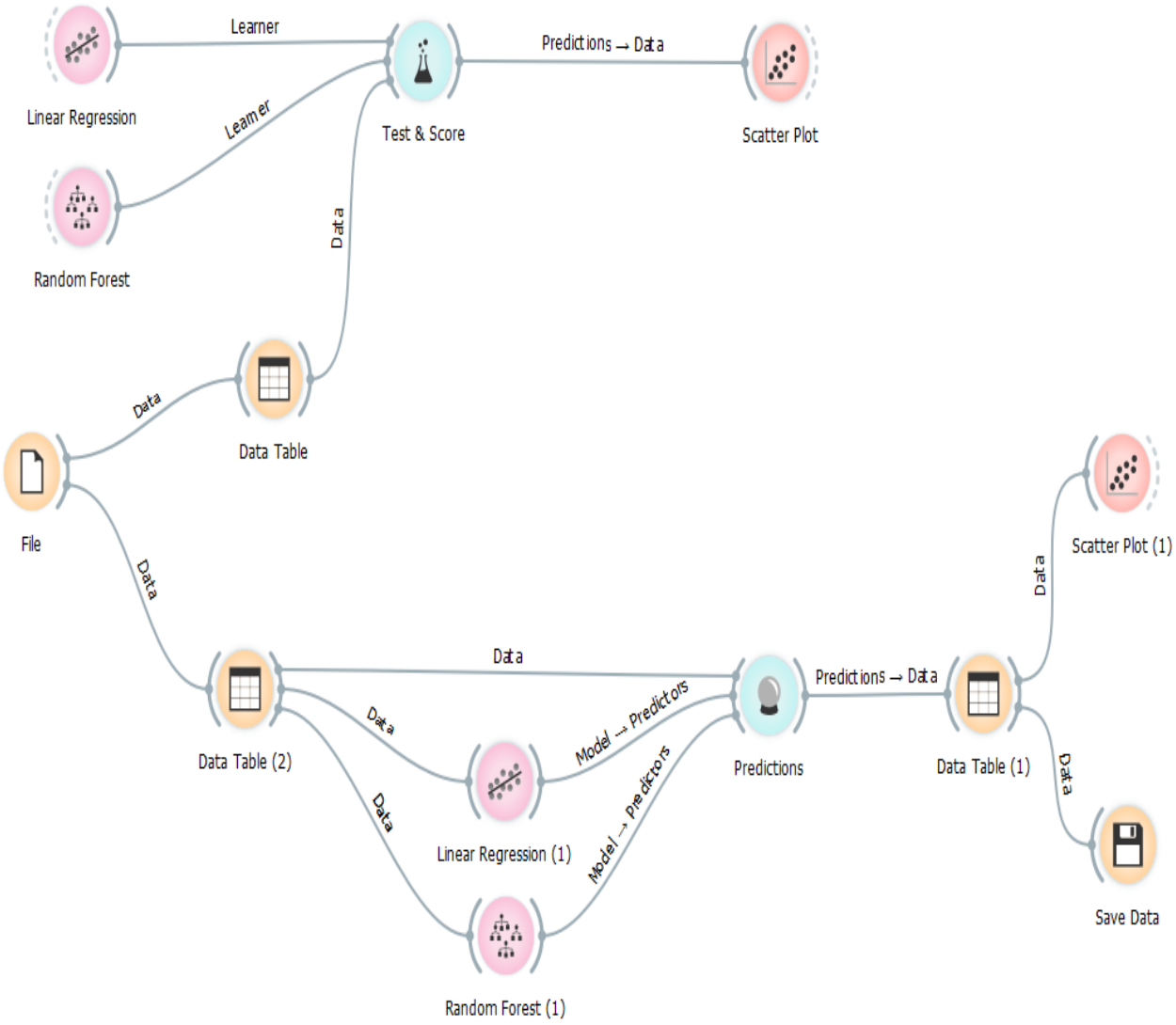

Fig 15: Design a first Prediction model

(Figure 16) shows obtained CSV file include which we have columns for our independent variables and target variable and predicted values by Random Forest and Linear regression models.

	A	M	N	O
1	ID	Speedindex	Linear Regression	Random Forest
2	first_170210_05_1BN	2309	2658.740286	2700.490476
3	first_170210_10_1BQ	1848	2618.688964	1953.348333
4	first_170210_28_1BJ	2782	3032.673935	2986.324167
5	first_170210_5S_1BP	1840	2441.402221	1993.811667
6	first_170210_GH_1BG	2125	2550.641254	2169.942976
7	first_170210_HW_1BK	2328	3085.390764	2376.246667
8	first_170210_J1_1BF	2487	2662.63521	2516.731667
9	first_170210_K0_1BH	2900	2613.280799	2703.668333
10	first_170210_NH_1BM	2058	2578.805519	2530.930476
11	first_170210_YK_1BR	1809	2571.246714	2494.651667
12	first_170211_36_5R	3742	2248.332151	3421.740476
13	first_170211_4W_5N	3379	2882.772106	3400.777976
14	first_170211_DP_5S	6640	2128.591512	5386.536667
15	first_170211_E3_5V	3328	2518.495245	3431.961667
16	first_170211_E4_5X	3238	2106.807887	3354.293333
17	first_170211_FM_5T	3477	2082.110701	3474.439444
18	first_170211_GT_5Y	3655	2271.137104	3390.610476
19	first_170211_M3_5W	2452	2219.473473	2843.21
20	first_170211_SF_5Q	2840	2308.588051	3042.67
21	first_170211_SJ_5P	3063	2224.678779	2873.563214
22	second_170211_36_5F	1822	2316.362661	1720.189286
23	second_170211_4W_5	1515	1942.051763	1527.060714
24	second_170211_DP_5	1680	2233.045826	1607.834351
25	second_170211_E3_5\	2061	2171.378955	1752.869048
26	second_170211_E4_5X	1519	2146.864383	1568.980952
27	second_170211_FM_5	1704	2127.816788	1532.827143

Fig 16: CSV file include actual variables and predicted variables of Speed Index

Figure 17 and 18 shows how we calculated the MSE, RMSE, MAE, MAPE and R-squared for both Random Forest AND Linear Regression and obtained the result. According to the result, it is clear that Random Forest is better than Linear Regression. As we said before because of a large and different range of data the obtained scores are high but are correct. In the Random Forest model, R-squared is 0.92 percent but in Linear Regression model R-Squared is 0.79 percent and as we said before if the estimated value is close to actual value, R-squared is close to one and both results are pretty good but Random Forest model results is better. Also other metrics in Random forest model achieved the smaller scores than linear regression. We can see the MSE and RMSE measures for Random forest have better scores than Linear Regression. RMSE shows the perfect fit of the model to the data, how close the observed data points are to the model's predicted values. Lower values of RMSE shows better fit. RMSE is a good measure of how exactly the model predicts the response, and it is the most important standard for fit if the main purpose of the model is the prediction.

	P	Q	R	S	T	U	V	W
1	Error(RF)	Absolute value of error(RF)	Square of error(RF)	Absolute value of error/actualvalue	Error(LR)	Absolute value of error	Square of error	Absolute value
2	-391.490476	391.490476	153264.7928	0.169549795	-349.7403	349.740286	122318.2677	0.151468292
3	-105.348333	105.348333	11098.27127	0.057006674	-770.689	770.688964	593961.4792	0.417039483
4	-204.324167	204.324167	41748.36522	0.073445064	-250.6739	250.673935	62837.42169	0.090105656
5	-153.811667	153.811667	23658.02891	0.083593297	-601.4022	601.402221	361684.6314	0.326849033
6	-44.942976	44.942976	2019.871092	0.021149636	-425.6413	425.641254	181170.4771	0.200301767
7	-48.246667	48.246667	2327.740877	0.020724513	-757.3908	757.390764	573640.7694	0.325339675
8	-29.731667	29.731667	883.9720226	0.011954832	-175.6352	175.63521	30847.72699	0.070621315
9	196.331667	196.331667	38546.12347	0.067700575	286.7192	286.719201	82207.90022	0.09886869
10	-472.930476	472.930476	223663.2351	0.229801009	-520.8055	520.805519	271238.3886	0.253063906
11	-685.651667	685.651667	470118.2085	0.37902248	-762.2467	762.246714	581020.053	0.421363579
12	320.259524	320.259524	102566.1627	0.085585121	1493.6678	1493.667849	2231043.643	0.399162974
13	-21.777976	21.777976	474.2802387	0.006445095	496.22789	496.227894	246242.1228	0.146856435
14	1253.463333	1253.463333	1571170.327	0.188774598	4511.4085	4511.408488	20352806.55	0.679428989
15	-103.961667	103.961667	10808.02821	0.031238482	809.50476	809.504755	655297.9484	0.243240611
16	-116.293333	116.293333	13524.1393	0.035915174	1131.1921	1131.192113	1279595.597	0.349349016
17	2.560556	2.560556	6.556447029	0.000736427	1394.8893	1394.889299	1945716.156	0.4011761
18	264.389524	264.389524	69901.8204	0.072336395	1383.8629	1383.862896	1915076.515	0.378621859
19	-391.21	391.21	153045.2641	0.159547308	232.52653	232.526527	54068.58576	0.094831373
20	-202.67	202.67	41075.1289	0.071362676	531.41195	531.411949	282398.6595	0.187116883
21	189.436786	189.436786	35886.29589	0.061846812	838.32122	838.321221	702782.4696	0.273692857
22	101.810714	101.810714	10365.42149	0.055878548	-494.3627	494.362661	244394.4406	0.271329671
23	-12.060714	12.060714	145.4608222	0.007960867	-427.0518	427.051763	182373.2083	0.281882352
24	72.165649	72.165649	5207.880896	0.042955743	-553.0458	553.045826	305859.6857	0.329193944
25	308.130952	308.130952	94944.68358	0.149505557	-110.379	110.378955	12183.51371	0.053556019
26	-49.980952	49.980952	2498.095563	0.032903853	-627.8644	627.864383	394213.6834	0.413340608
27	171.172857	171.172857	29300.14697	0.100453555	-423.8168	423.816788	179620.6698	0.248718772
28	-40.08	40.08	1606.4064	0.024453935	-413.6234	413.623403	171084.3195	0.252363272

Fig 17: CSV file shows how calculated MSE, RMSE, MAE, MAPE and R^2

Random forest	
MSE	1302345
MAPE	14.33318
RMSE	1141.203
MAE	583.3631
R^2	0.917789

Linear Regression	
MSE	7173389
MAPE	45.43928
RMSA	2678.318
MAE	1586.916
R^2	0.790203

Fig 18: Prediction Result

In addition, we calculated MAPE measure, because often is effective for purposes of reporting. MAPE is the average of absolute error divided by actual observation values. The MAPE is often used in practice because of its very great description in terms of relative error [26]. Displays accuracy as a percentage of the error. Because this number is a percentage, it can be easier to understand than the other statistics. We can see in the (figure 18) that Random Forest model has

about 14 percent error and Linear Regression model has about 45 percent error so we can use Random Forest because its accuracy is more than Linear Regression.

In the part of Figure 15 you can see scatter plot object which is another way to use to shows the prediction result. We obtained two scatter plots based on actual values of Speed Index and predicted values in Random Forest model and Regression model. Figures 19 and 20 show the scatter plots of Random Forest and Linear Regression model, we can see that in Random Forest Graph, the points are closer to the line than they are in Linear Regression Graph. Therefore, the predictions in the Random Forest Graph are more accurate than in Linear Regression graph. The regression model on the RF accounts for 92.0% of the R-Squared (variance) while the one on the LR accounts for 74.2%. The more R-Squared (variance) that is accounted for by the regression model the closer the data points will fall to the fitted regression line. Theoretically, if a model could explain 100% of the R-Squared (variance), the fitted values would always equal the observed values and, therefore, all the data points would fall on the fitted regression line.

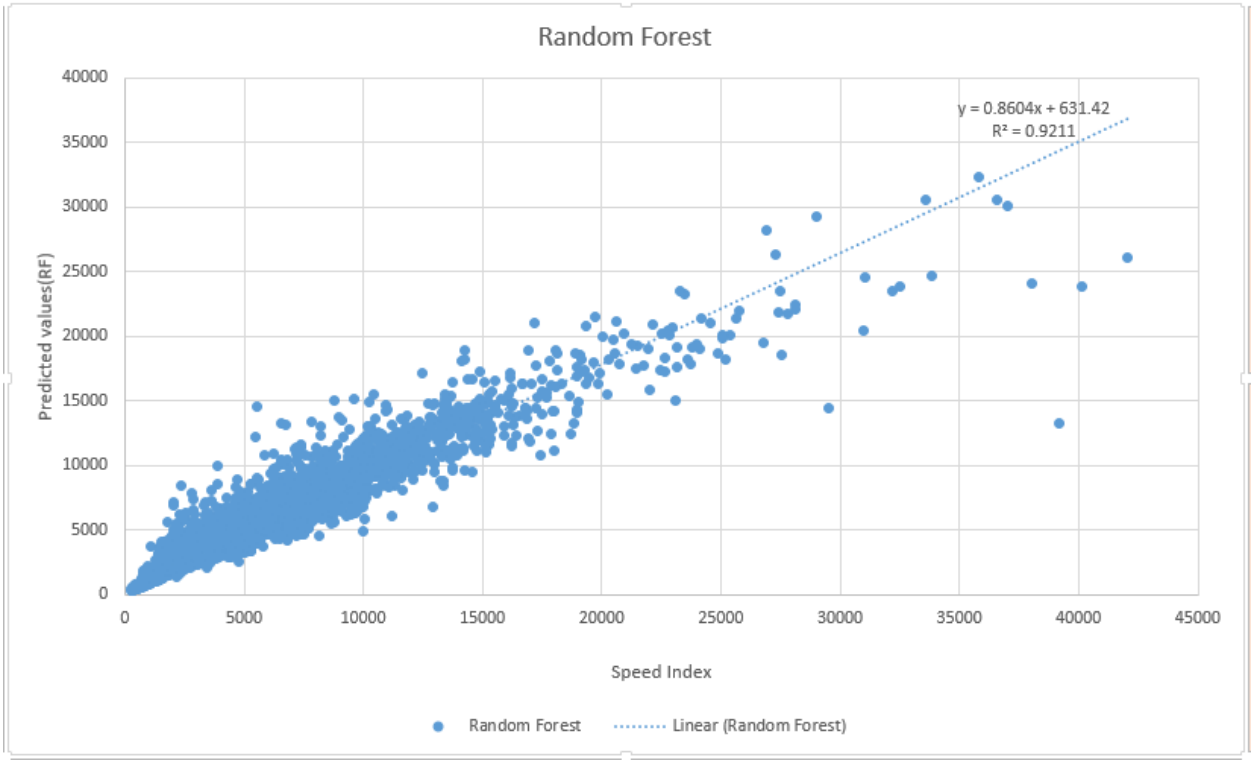

Fig 19: RF Scatter plot

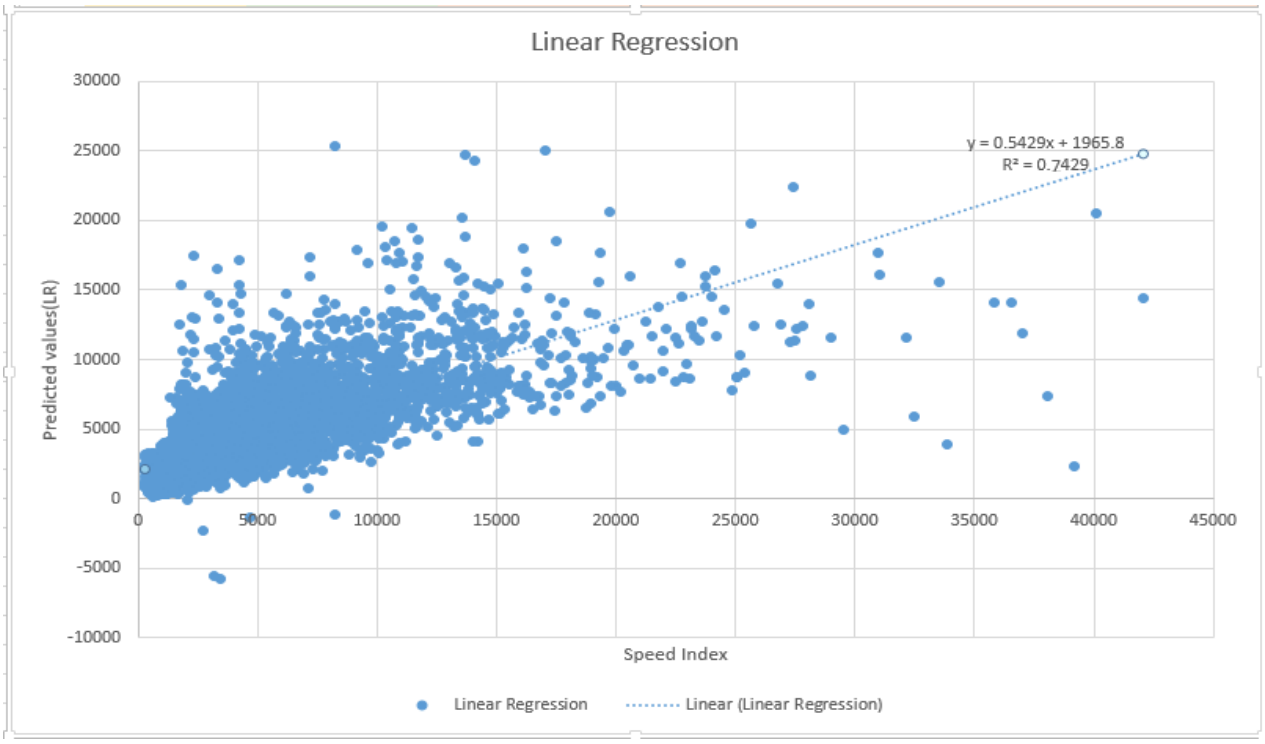

Fig 20: LR Scatter plot

As example Figures 21and 22 show the detail of one point in scatter plot related to Random Forest model.

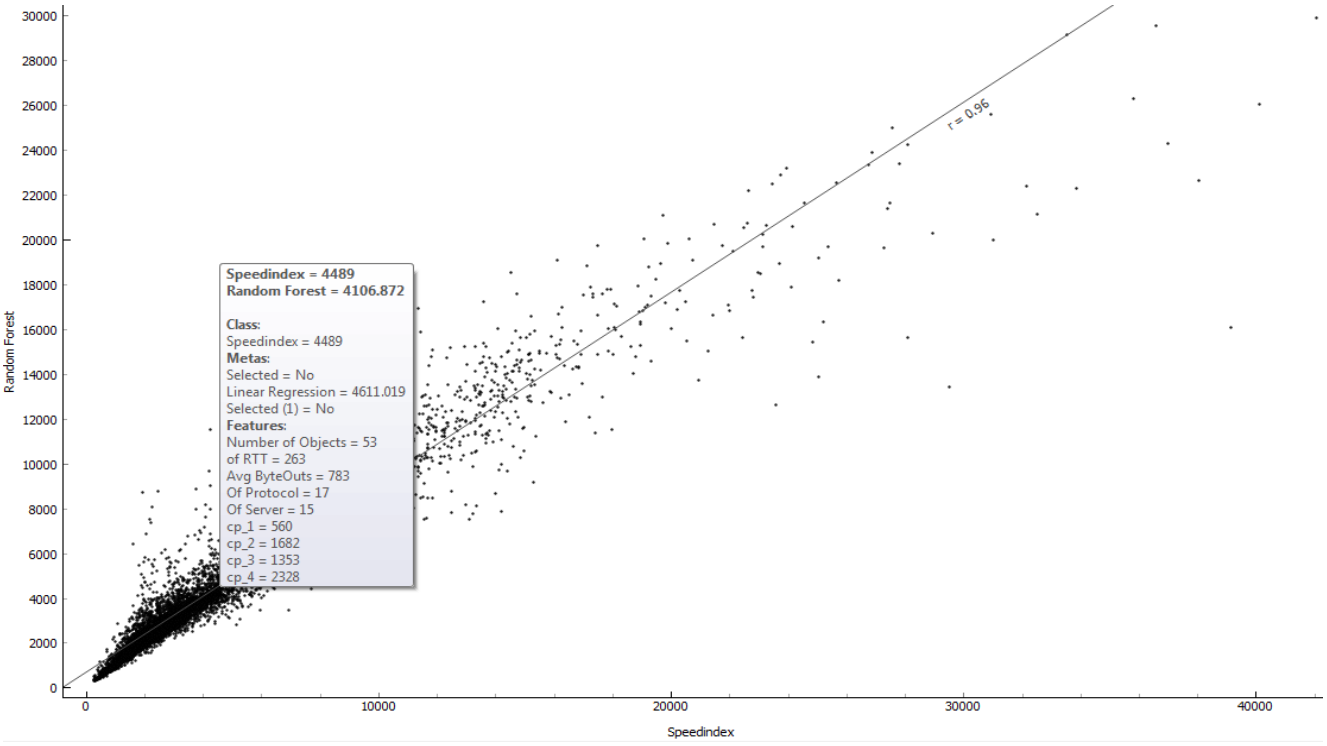

Fig 21: details of one point in RF model

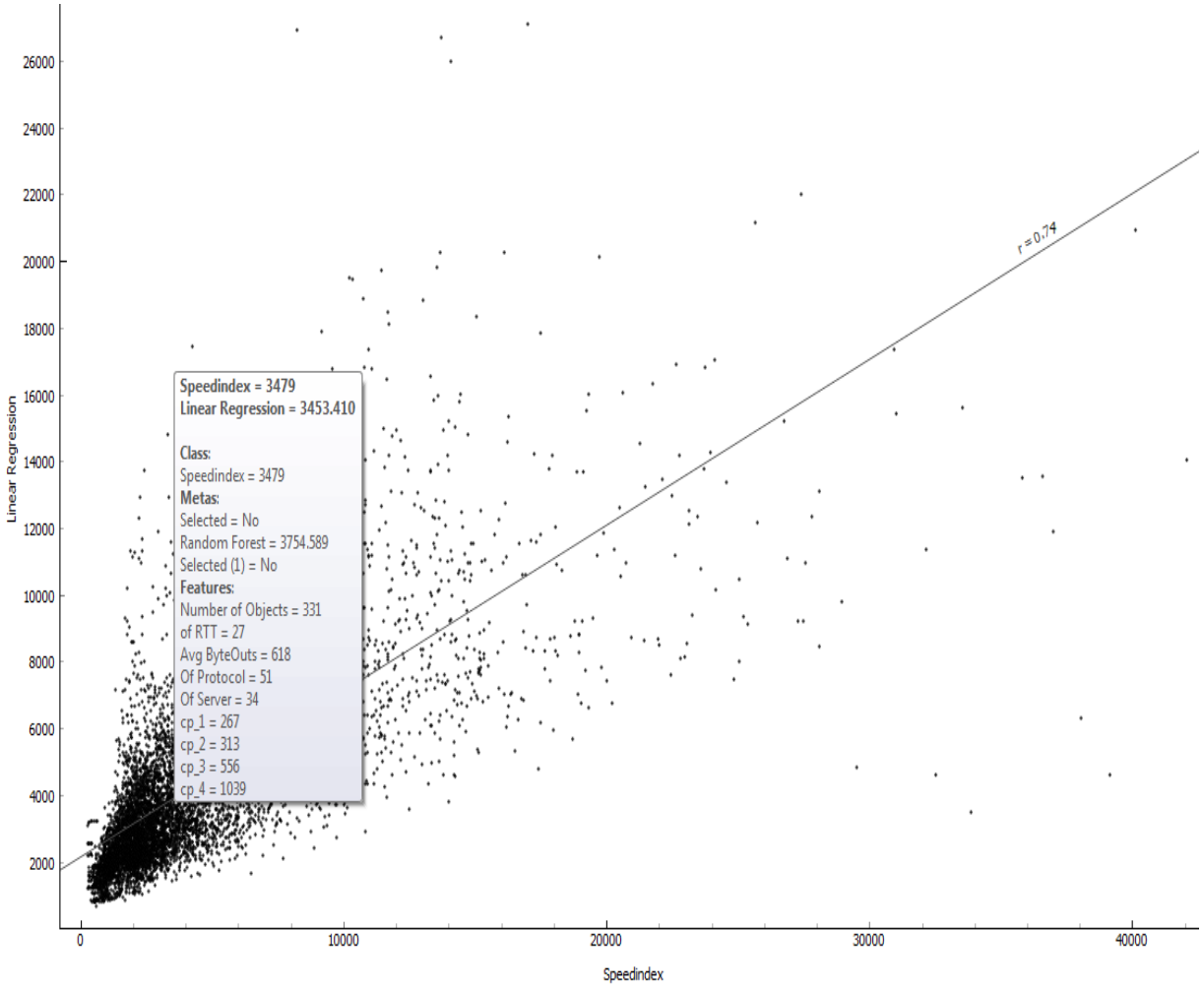

Fig 22: Details of one point in LR model

2) Cross Validation

The goal of this design is answering to a question that if in the future we want to predict the page load time of website that we never saw it before, can we use this model to build another model (result) or no? According to this kind of design, the answer is yes.

In this design, we use some sample of trace (select 90 percent of sites randomly) for training and some other sites (10 percent of the dataset) for testing (Figure 27). So we build the model then check the result with calculating the MSE, RMSE, MAE, MAPE, and R-squared for Random Forest like our first design (Figure 28).

According to two previous results we chose Random Forest to check Cross Validation model. In the Random Forest model, R-squared is 0.83 percent which is lower than First experiment result that we achieved and as we said before if the estimated value is close to actual value, R-squared is close to one and in this model, R-squared is more than 0.50 percent and it is acceptable. Also, other metrics in Random forest model are proper. With checking the result shown in (Figure 28) we can see that the Random Forest model has about 38 percent error. The results of this design is worse than the first experiment and is better than the third experiment and it is acceptable to use in the future.

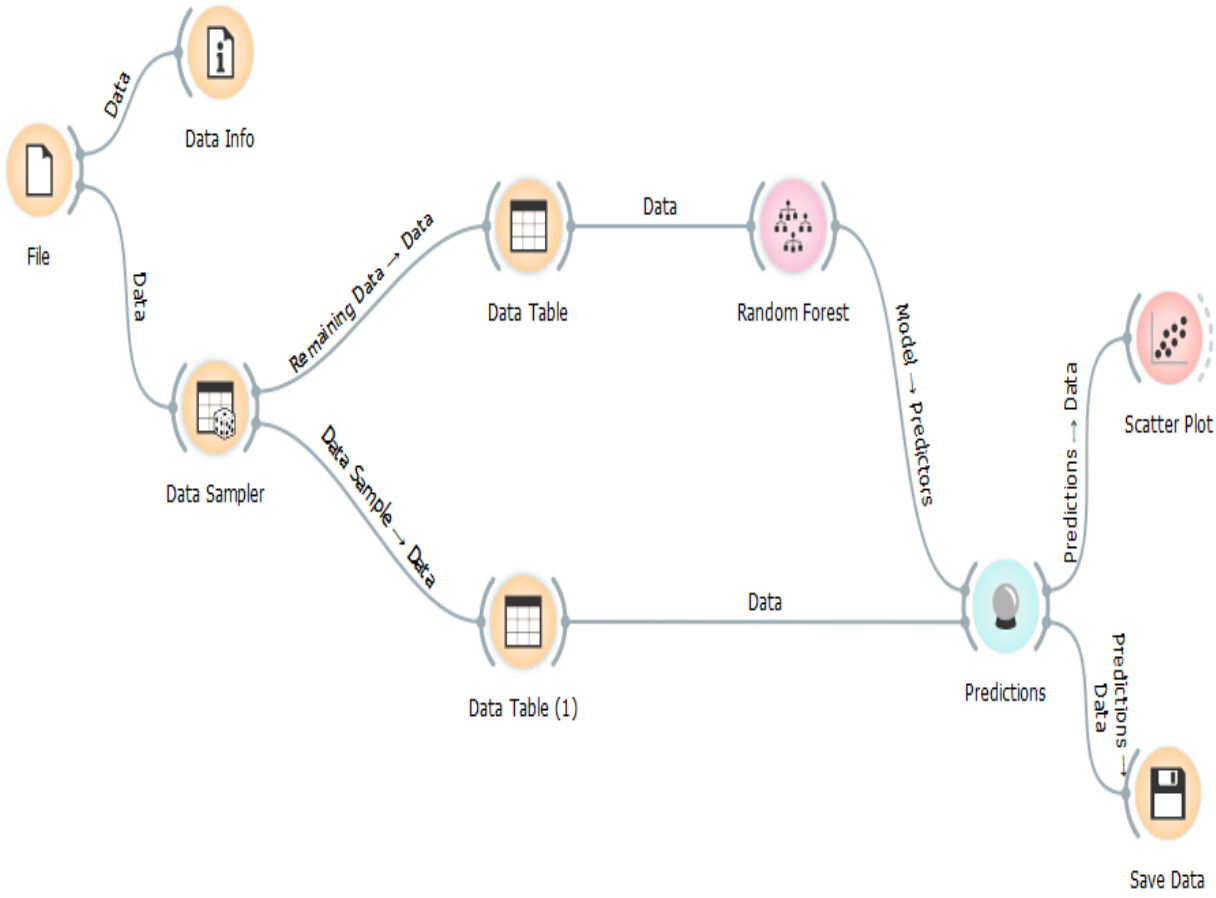

Fig 23: Design a Cross Validation model

Random forest	
MSE	1289395.163
MAPE	37.91591469
RMSE	1135.515373
MAE	589.7386285
R^2	0.835728919

Fig 24: Prediction Result related to Cross Validation design

Also, in this part, we obtained a scatter plots based on actual values of Speed Index and predicted values in Random Forest model (Figure 29)

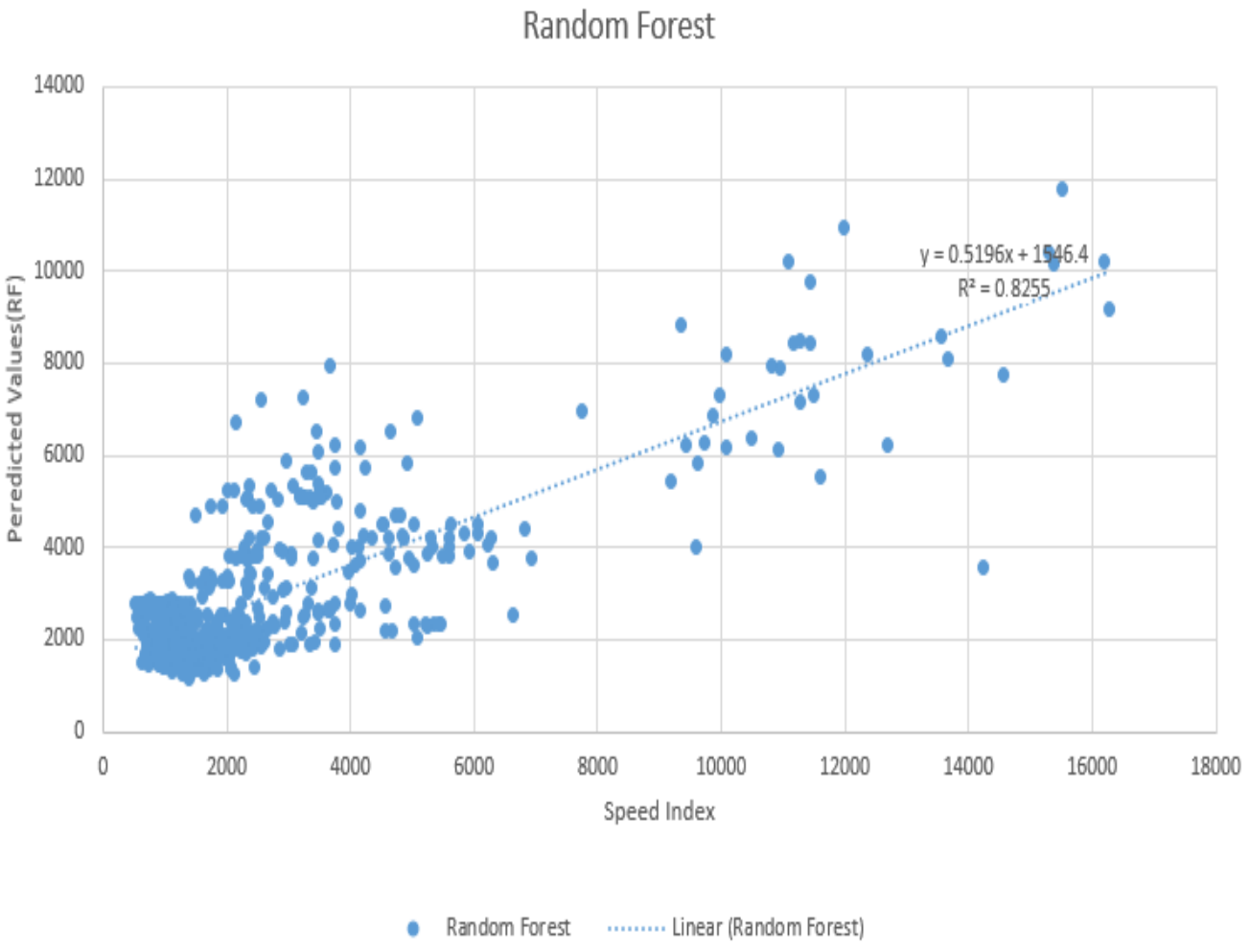

Fig 25: RF Scatter plot related to Cross Validation design

3) Third Experiment

In this experiment, we have 2 different part of things. We used a sample of the trace for training (50 percent of our dataset include 5 popular Domains and their Subdomains) and a sample of the trace for testing (50 percent of our dataset include other 5 popular Domains and their subdomains).

So we have two independent sets and first, we build the model (Figure 23) then check the result with calculating the MSE, RMSE, MAE, MAPE, and R-squared for both Random Forest and Linear Regression like our first design (Figure 24).

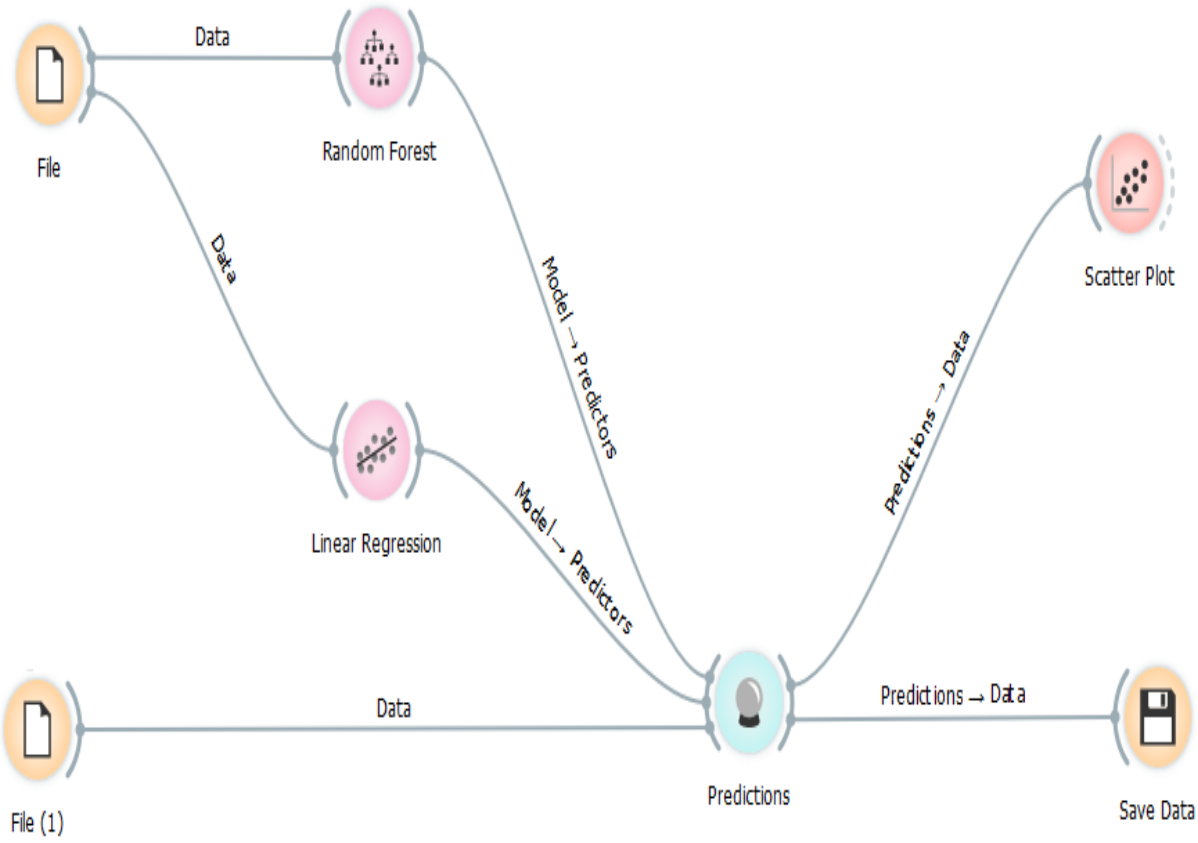

Fig 26: Design a third Prediction model

Random Forest	
MSE	10608437
RMSE	3257.06
MAE	2132.032
MAPE	45.42573
R^2	0.677133

Linear Regression	
MSE	13959370
RMSE	3736.224
MAE	2310.551
MAPE	52.09511
R^2	0.56219

Fig 27: Prediction Result related to third design

According to the result, like the first model, it is clear that Random Forest is better than Linear Regression. As we said before because of a large and different range of data the obtained scores are high but are correct. In the Random Forest model, R-squared is 0.67 percent but in Linear Regression model R-Squared is 0.56 percent and as we said before if the estimated value is close to actual value, R-squared is close to one and both results are pretty good but Linear Regression model results is better.in this model other metrics in Random forest model achieved the smaller scores than linear regression. We can see the MSE and RMSE measures for Random forest have better scores than Linear Regression. With checking the result shows in (Figure 24) we can see that Random Forest model has 45.42 percent error and Linear Regression model has 52.09 percent error so we can use Random Forest because its accuracy is more than Linear Regression. The results of this experiment is worse than 2 previous experiments but is acceptable.

Like Optimistic Validation part, in this part, we obtained two scatter plots based on actual values of Speed Index and predicted values in Random Forest model and Regression model.

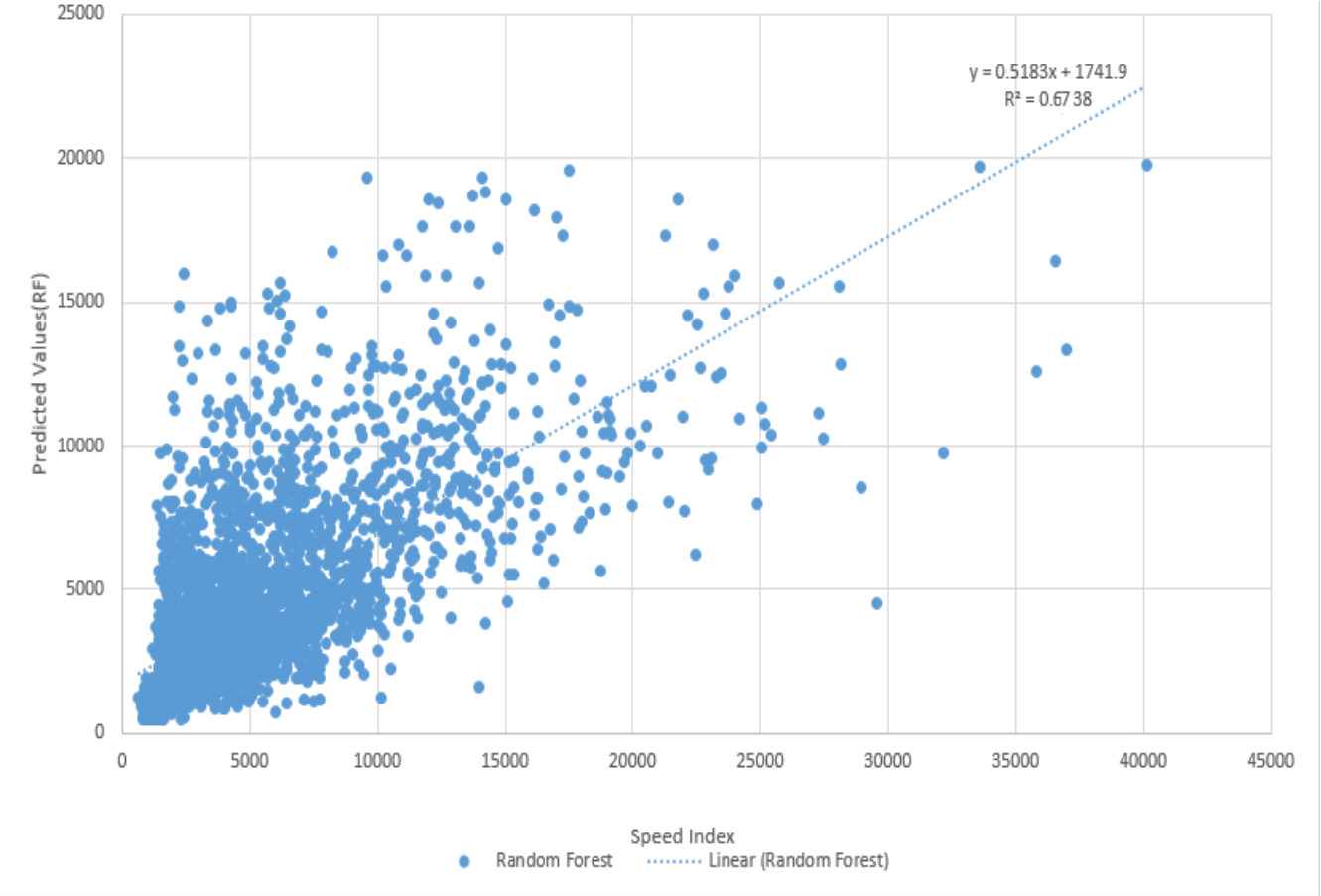

Fig 28: RF Scatter plot related to third design

Figures 25 and 26 show the scatter plots of Random Forest and Linear Regression model, we can see that in Random Forest Graph, the points are closer to the line than they are in Linear Regression Graph. Therefore, like our first design the predictions in the Random Forest Graph are more accurate than in Linear Regression graph. The regression model on the RF accounts for 67.0% of the R-Squared (variance) while the one on the LR accounts for 56.0%. The more R-Squared (variance) that is accounted for by the regression model the closer the data points will fall to the fitted regression line.

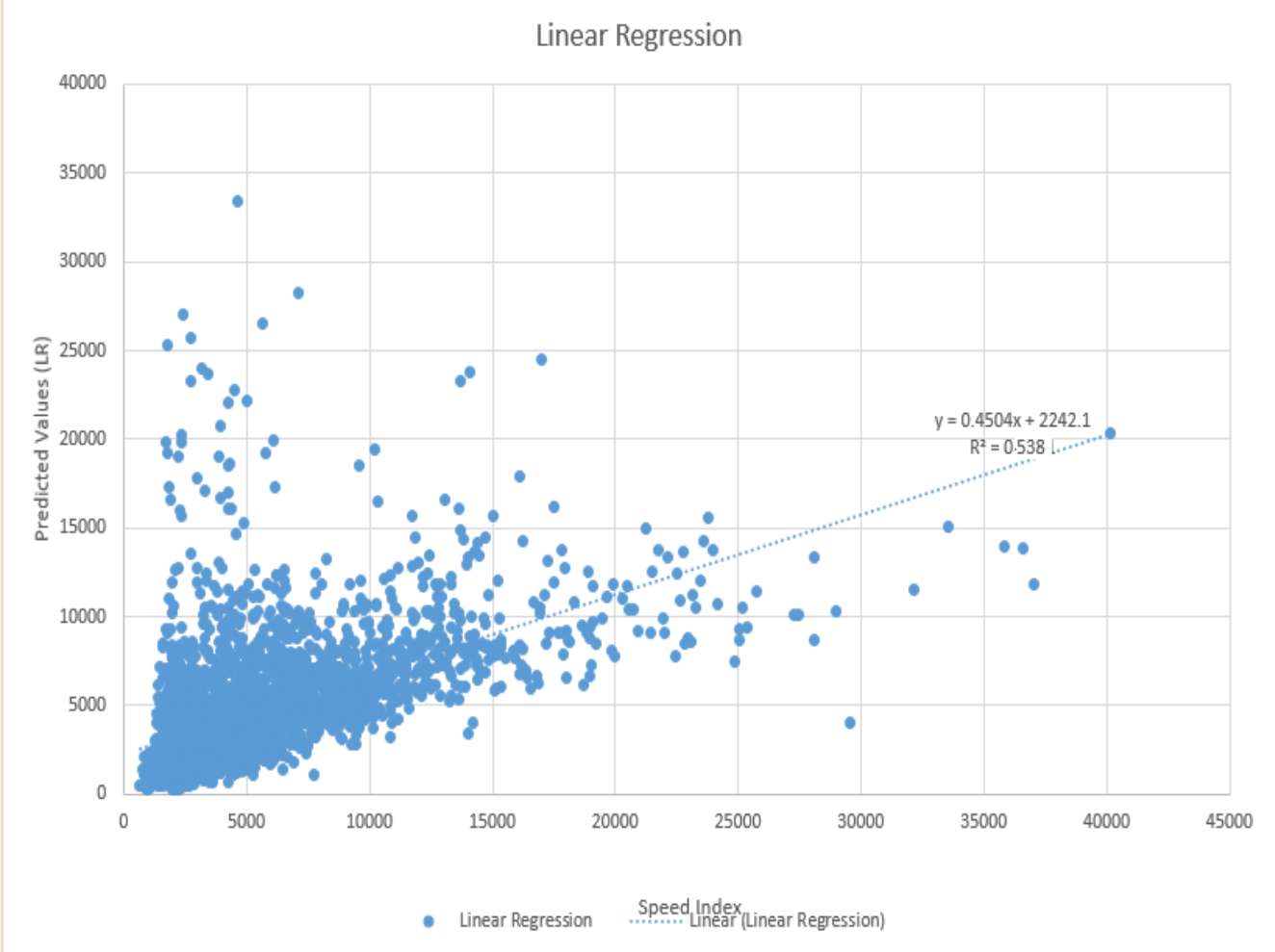

Fig 29: LR Scatter plot related to third design

Chapter 4
Conclusion and future work

4.1 Conclusion

The increasing complexity of web pages and its impact on performance has been well-recognized. So we need to predict QoE for Web Pages since it can help ISPs to have the best performance and give the best service to their client. In this paper, we presented a first attempt at characterizing web page metrics likes Objective and Subjective metrics. then We characterized the objective metrics for 10 popular Domain in Italy and their sub Domains, we used a HAR file to captured all metrics related to each web page visited and finally obtained 6948 HAR file which used to make our dataset, by parsing important metrics from HAR files obtained to good dataset to use as input of our machine Learning. We designed a Machine Learning model to estimate the QoE for Web Pages. Used two Regression model (Random Forest and Linear Regression) for predicting Speed Index metric. Obtained results shows that Random Forest has better behavior than Linear Regression and can be used to predict Speed Index based on other metrics. The validation results have shown there is a high correlation between actual values and estimated values and the model that we designed is good for measure and estimate page load time and Speed Index using network traffic that are available in ISPs.

4.2 Future work

As future work, we can use other network performance features to better estimate quality of experience of users while browsing Web Pages. Moreover, we will study whether large datasets could help improve estimations. Finally, our estimate could be used to reconfigure the network, thus improving QoE.

References

[1] S. Winkler, Video Quality and Beyond, Symmetricom, 2007.

[2] Trevisan, Martino; Drago, Idilio; Mellia, Marco (2017). PAIN: A Passive Web Speed Indicator for ISPs. Trevisan, Martino; Drago, Idilio; Mellia, Marco (2017). PAIN: A Passive Web Speed Indicator for ISPs. In: ACM SIGCOMM Workshop on QoE-based Analysis and Management of Data Communication Networks, Los Angeles, California, USA, August 21 - 25, 2017.Networks, Los Angeles, California, USA, August 21 - 25, 2017.

[3] ITUT Rec. "P. 800.1, mean opinion score (MOS) terminology". International Telecommunication Union, Geneva, 2006

[4] W. Cherif, A. Ksentini, D. Négru, and M. Sidibé. "A_PSQA: Efficient real-time video streaming QoE tool in a future media internet context". In Proceedings of 2011 IEEE International Conference on Multimedia and Expo (ICME), pages 1–6. IEEE, 2011

[5] G. Rubino. "The PSQA project". INRIA Rennes, http://www.irisa.fr/armor/lesmembres/ Rubino/myPages/psqa.html, 2010

[6] M. Ghareeb and C. Viho. "A multiple description coding approach for overlay multipath video streaming based on QoE evaluations". In Proceedings of International Conference on Multimedia Information Networking and Security (MINES), pages 39–43. IEEE, 2010.

[7] P. Reichl, B. Tuffin, and R. Schatz. "Economics of logarithmic quality-of-experience in communication networks". In Proceedings of 2010 9th Conference on Telecommunications Internet and Media Techno Economics (CTTE), pages 1–8. IEEE, 2010.

[8] O. Issa, F. Speranza, and T. H. Falk, "Quality-of-experience perception for video streaming services: Preliminary subjective and objective results," in Proc. APSIPA'12, 2012, pp. 1-9

[9] https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics/speed-index

[10] https://www.webpagetest.org/about

[11] Martino Trevisan, Idilio Drago, and Marco Mellia. 2017. PAIN: A Passive. Web Speed Indicator for ISPs

[12] HAR, Wikipedia: https://en.wikipedia.org/wiki/.har

[13] CSV, Bigcommerce: https://www.bigcommerce.com/ecommerce-answers/ CSV

[14] Statistical Regression, Wikipedia: https://en.wikipedia.org/wiki/Regression_analysis

[15]. Orange Machine Learning Application, Orange: https://orange.biolab.si/

[16]. Orange Machine Learning Application, Wikipedia: https://en.wikipedia.org/wiki/Orange-(Software)

[17] Orange Machine Learning Application, Wikipedia: https://en.wikipedia.org/wiki/

[18] Multiple linear regression, Investopedia: https:// www.investopedia.com/terms/m/mlr.asp

[19] Random Forest, https://en.wikipedia.org/wiki/Random-forest

[20] Random Forest, Microsoft: https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/decision-forest-regression

[21] R-squared, Minitab: http://blog.minitab.com/blog/adventures-in-statistics-2

[22] MSE, Wikipedia: https://en.wikipedia.org/wiki/Mean_squared_error

[23] RMSE, Vernier: https://www.vernier.com/til/1014/

[24] MAE, Wikipedia: https://en.wikipedia.org/wiki/Mean_absolute_error

[25] MAPE, Wikipedia: https://en.wikipedia.org/wiki/Mean_absolute_percentage_error

[26] Mean Absolute Percentage Error for Regression Models Arnaud de Myttenaerea,c, Boris Golden a , B´en´edicte Le Grand b , Fabrice Rossic, ∗ aViadeo, 30 rue de la Victoire, 75009 Paris – France

[27] SpeedIndex, nccgroup: https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2015/june/speed-index--how-it-works-and-what-it-means/

[28] Venkat Mohan. , Y. R. Janardhan Reddy, K. Kaplan. Active and Passive Network Measurements:
https://pdfs.semanticscholar.org/22f5/f25224c515d4fff3c1c1ce6a6c6a03b62c42.pdf

[29] Round Trip time: https://www.incapsula.com/cdn-guide/glossary/round-trip-time-rtt.html

[30] Protocol: https://www.quackit.com/how-websites-work/web-protocols.cfm

[31] Parikshit Julurp, 2015 : Measurement and improvement of quality of experience for online video streaming services.