

Article



# An Innovative Approach for the Evaluation of the Web Page Impact Combining User Experience and Neural Network Score

Alessandro Massaro \*<sup>®</sup>, Daniele Giannone, Vitangelo Birardi and Angelo Maurizio Galiano

Dyrecta Lab, Research Institute, 70014 Conversano, Italy; daniele.giannone@dyrecta.com (D.G.); vitangelo.birardi@dyrecta.com (V.B.); angelo.galiano@dyrecta.com (A.M.G.)

\* Correspondence: alessandro.massaro@dyrecta.com

**Abstract:** The proposed paper introduces an innovative methodology useful to assign intelligent scores to web pages. The approach is based on the simultaneous use of User eXperience (UX), Artificial Neural Network (ANN), and Long Short-Term Memory (LSTM) algorithms, providing the web page scoring and taking into account outlier conditions to construct the training dataset. Specifically, the UX tool analyses different parameters addressing the score, such as navigation time, number of clicks, and mouse movements for page, finding possible outliers, the ANN are able to predict outliers, and the LSTM processes the web pages tags together with UX and user scores. The final web page score is assigned by the LSTM model corrected by the UX output and improved by the navigation user score. This final score is useful for the designer by suggesting the tags typologies structuring a new web page layout of a specific topic. By using the proposed methodology, the web designer is addressed to allocate contents in the web page layout. The work has been developed within a framework of an industry project oriented on the formulation of an innovative AI interface for web designers.

Keywords: user experience; artificial intelligence; LSTM; ANN; web designer interfaces

# 1. Introduction

Many approaches about the analysis of the usability of an internet site are based on the check of usage patterns through log files or on the analysis of static pages according to criteria defined by the web designer. Some examples are the Automatic Navigability Testing System (ANTS) [1], and the tool TEA supporting the user tests by means of automated tasks [2]. Other tools as WebTango [3] evaluates a website according to quantitative attributes empirically validated. The ISEtool [4] performs an automated analysis to calculate information for all links of a web page. Usability is one of the main factors determining the success of a website. It is, therefore, important to have methods for evaluating the usability of websites. Such methods could be used to help web designers make their websites more usable. In [5], the studies are focused on the usability problems of a website, implementing an evaluation tool based on web questionaries for user feedback. The User eXperience (UX), is surely a tool suitable for the usability check of specific web pages. In [6], some researchers adopted Graphical User Interfaces (GUIs) for the UX analysis of web maps. The UX can also be improved by Support Vector Machine (SVM) algorithms integrated into E-Commerce platforms [7]. Alternative solutions are the user action monitoring on the web pages [8] and the web mining discovering navigation patterns [9,10]. User personality can be an important factor for product design [11]. In this direction, an idea can be to use the user feedback in order to improve the web page design. The web page design approach by UX is a concept adopted in different application fields [12,13], and it is a topic of the actual research. The web page classification is a way to select the impact pages. In this direction, the Recurrent Neural Network (RNN) could provide important information about the best web page features using tags [14]. The use of the score is useful in order to weight the web page classification algorithms such as Support Vector Machine (SVM) [15]. The Long Short



Citation: Massaro, A.; Giannone, D.; Birardi, V.; Galiano, A.M. An Innovative Approach for the Evaluation of the Web Page Impact Combining User Experience and Neural Network Score. *Future Internet* 2021, *13*, 145. https://doi.org/ 10.3390/fi13060145

Academic Editor: Ivan Serina

Received: 9 April 2021 Accepted: 27 May 2021 Published: 31 May 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/).

Term Memory (LSTM) networks have been applied for the web page depth viewability prediction [16]. Moreover, a key factor influencing the effectiveness of a user interface is the usability resulting from its design, thus confirming the importance of the UX about the design of software interfaces [17]. Concerning E-Commerce pages, Artificial Intelligence (AI) algorithms play an important role in shaping consumer demand [18,19] and defining factors influencing the related use [20]. Other methods such as the UX click rates [20] are also interesting to formulate recommendation models in E-Commerce. Actually, there is no generic model based on AI, usable for web page designers, including all the aspects found in state of the art such as UX, scoring, and AI. Some concepts analyzed for the E-commerce systems are also suitable for the impact of generic web pages. Web interaction metrics are considered fundamental to provide a framework according to the application field. In the case of Digital Marketing (DM) [21], some parameters are based on Web Analytics (WA) [22], Search Engine Optimization (SEO) [23], Search Engine Marketing (SEM) [24], and Click-Through Rate (CTR) prediction model based [25]. Specifically, concerning WA, some important parameters that are considered for the metrics are traffic type, time on site, keywords, new visitors, and returning visitors [21,26–28]. Other metrics and methods suitable for web analysis are discussed in [29]. Following this direction, has been formulated a platform, which automatically provides the best web page template to consider addressing the design. The paper describes the architecture of an innovative platform based on the concepts previously discussed and suggesting a model to construct and design a web page. This architecture has been developed and tested within the framework of an industry project by proving the functionality of the proposed model. The above state of the art is indicative of generic approaches and tools applicable to find possible solutions to assign a score concerning the web page navigation by suggesting some basic approaches to follow for the design of a user interface integrating a web page scoring system.

Summarized in Table 1 are some traditional metrics and methods of website analysis. All the mentioned metrics are based on a specific empiric estimation of the parameters or on strategies to increase the web page visibility based on a single approach. The proposed approach is an alternative method because it takes into account the impact of a specific web page by simultaneously processing the related tags, the score assigned by the user, and the navigation parameters. The methodology suggests to the designer, the tags typologies structuring a new web page layout of a specific topic. The total scoring of the web page defines the "web page impact": if higher is the total score, more suitable will be the page to contain specific topics. The score is assigned by a selected user knowing a full scenario of the web page contents (for example, a researcher is chosen to score a web page containing the research and development (R&D) topics). The best page will be the one that obtains the highest average ratings from users having the same basic knowledge about the viewed topic.

Metric/Method	Reference	Category	Description	
Visitor Type (method)	[29]	Site Usage	User accessing the web page	
Visit Length (metric)	[29]	Site Usage Total amount of time a visitor spends on the websi		
Visitor Location (method)	[29]	Site Usage Location of visitors accessing the website		
Number of Visitors (metric)	[29]	Site Usage	Number of visitors/users visiting a web page	
Click-Through Rate (metric)	[25]	Site Usage	Ratio between the number of clicks generated by a web page and the number of times that web page itself has been viewed.	
UX analysis (method)	[17,20]	Site Usage	UX adopted for web page usability	
Visitor Path (method)	[29]	Site Content Analysis	Navigation path	
Top Pages (metric)	[29]	Site Content Analysis	Web pages receiving the most traffic	
Tags classification using neural network (method)	[14]	Site Content Analysis	Classification of a web page by tag and meta tag information	
Keyword (method)	[29]	Referres	Keywords selected by visitors	
Errors (metric)	[29]	Quality Assurance	Errors occurred attempting to retrieve the page	

#### 3 of 21

# 2. Material and Methods

In this section is discussed the project framework within which the research has been developed methods applied about web page impact evaluation.

## 2.1. Main "ANDUIA" Project Framework: Specifications and Platform Architecture Design

The research project involves the development of an intelligent software interface for the design of web pages and web applications, using AI and UX simultaneously to create a designer interface. This interface is linked to a back-end system executing data processing. The user, by means of the front-end, navigates in the web page passing information to the AI algorithm, which, by the tracking analysis of the user navigability, identifies the anomalous behaviors and collects feedback. The main functions of the AI engine are embedded in the platform and are included in the following 2 modules:

- Approval module form, which, by analyzing the data collected on all types of users (user experience), allows to identify those behaviors that slow down the use of web platform features (user behavior);
- Optimal site design module, which, taking into account the results processed by the approval module, allows the design optimization of the web page.

As shown in the block diagram of Figure 1, the user's experience is the starting point to acquire details about webpage design. The platform, in addition to allowing access to a registered designer user or the registering a new user, allows training the neural network for the calculation of a score indicating the best classes of web page templates to consider for the layout design. In the proposed platform, the UX analysis deriving from navigation, together with the web page features extraction, allows defining the best web page layout.



Figure 1. Front-end and back-end architecture of the platform "ANDUIA".

The architecture of Figure 1 integrates a system for analyzing a certain number of web pages and the LSTM algorithm, trained on a number of pages evaluated by the users and by UX. The actors of the whole system were indicated in the Unified Modelling Language (UML) diagram of Figure 2a, and were:

- Designer/developer (user);
- Backend ('ANDUIA' platform);
- AI algorithm (LSTM model listed in Appendix A).

The platform collected the user feedback with a preliminary score ranging from 1 to 10, and UX data such as the click number, navigation time, and page movements. All

data were stored in a SQL database. An Artificial Neural Network (ANN) implementing Multilayer Perceptron (MLP) was adopted to optimize the construction of the training dataset (see Appendix B) by neglecting UX outlier conditions by analyzing anomalous user conditions about click number, navigation time, and mouse movements. At the end of the training phase, the LSTM algorithm was able to evaluate new web pages in order to provide a suggestion about the best layout to follow to develop a new similar page (similarity with a page having a high score and reporting the same topic). Figure 2a were indicated all the main functions of the platform, such as LSTM training, LSTM testing, UX (user behavior), navigation feedback storage, and data saving. Figure 2b was illustrated the UML sequence diagram defining the data flow involving all the system actors.



**Figure 2.** UML diagrams: (**a**) functional diagram representing the main functions of the platform "*ANDUIA*" about automatisms; (**b**) sequence diagram.

#### 2.2. Methods: Web Page Scoring Based on Ccombined UX and LSTM Approaches

The method applied for the platform realization is sketched in Figure 3 where it was possible to distinguish 2 main dashboards:

- Outlier dashboard providing information about anomalous user navigation behaviors to not consider for the final web page scoring;
- LSTM score dashboard indicating the final results by processing UX data, score user (user feedback provided during the navigation), and web page tags.

Concerning web page tags, for the realization of the designer interface, has been created a training dataset consisting of a set of HTML web page source files by ranking each file from 1 to 10. This scoring indicates the quality of the web page source code and UX evaluation. 100 files have been collected for the LSTM training by considering mostly generic website home pages and by storing the related source codes in files named as:

#### <file\_index>\_<vote>.html.

For the experimentation phase have been considered the 100 different pages having different contents to be submitted to the score of users. Each web page is assigned a name and a preliminary score pre-classifying the web page to be evaluated by the LSTM algorithm. Two main topics of the web pages were considered: the research and development (R&D) web page and web pages of companies, by selecting for the experimentation two

main class of users (researchers scoring the first type of web pages, and industry employees for the scoring of the second type ones). The preliminary scoring scale for each web page follows the metrics of Table 2.



**Figure 3.** Flowchart describing the proposed model to estimate scoring for the best web page using UX, ANN, and LSTM algorithms.

Table 2. Web page preliminary scoring.

Score	Description
From 1 to 4	Poor web pages of javascript libraries and css files
From 5 to 7	Old-style web pages that do not use current frameworks
From 8 to 10	Modern web pages, characterized by a strong use of div, javascript libraries and css files

In order to train the LSTM algorithms accurately, html web pages with different scores have been selected, inserting in the training dataset both pages poor and rich in tags contents.

Tokenization was an approach suitable for LSTM training model and sentence segmentation [30]. By applying a tokenizer, the 5000 most used tags were selected from the source codes, including their attributes. For each source code, were considered the first 10,000 characters. For source codes with fewer characters were returned to a length of 10,000, adding zeros at the beginning. The sequence of detected tags was replaced by a sequence of numbers where each number was the identifier of a specific HTML tag: This operation was performed using the Tokenizer operator from the Keras framework; in this way the neural network can analyze past texts in input as they were converted into a format understandable to the network itself:

max\_words = 5000
max\_len = 10,000
tok = Tokenizer(num\_words = max\_words)
tok.fit\_on\_texts(x)
sequences = tok.texts\_to\_sequences(x)
sequences\_matrix = sequence.pad\_sequences(sequences, maxlen = max\_len)

By starting from a folder containing the HTML files, the memory was filled by a list showing the extracted contents from the HTML files. The tags included were extracted from each HTML file of their attributes and, among these, the 5000 most used were selected. In Figure 4a was illustrated an explosion of the "Web page tags" function indicated in Figure 2, where it was possible to distinguish the different data flows involved in the data processing (source code acquisition, source code classification, HTML tag extraction, tag selection, tokenization, code conversion, LSTM model training, and LSTM model adoption). An example of tokenization processing was shown in Figure 4b. It was observed that the tokenization was also applied for other web page tags listed in Table 2, defining the features of the web page layout.



(b)

**Figure 4.** (a) Web page analysis and LSTM training approach. (b) Example of tokenization process "translating" a web page text into a sequence number.

The LSTM model was trained by taking into account the attributes of Table 3 and the sequence of words used in the text.

Table 3.	Main	tags and	l attributes	s of a	HTML	web pag	e.
		()					/

Tag HTML	Use	Attributes
<html></html>	Beginning of document	//
<title></title>	Page title	//
<head></head>	Header	//
<body></body>	Body of the page	//
<h1>-<h6></h6></h1>	Other headers	//
<pre></pre>	Preformatted, as written	//
<center></center>	Centered object	//
<a href="url"></a>	Hyperlink	Id, class
<img src="url"/>	Insert image	Align, alt, id, class, width, height, border
	Paragraph	Align
	Next row	//
<ul><li></li></ul>	Bulleted list and related items	Type, value, align

6 of 21

Table 3. Cont.

Tag HTML	Use	Attributes	
<form></form>	Data entry form	Align, action, enctype	
<input/>	Text input box	//	
<select></select>	Multiple choice box	//	
	Table	Border, cellspacing, cellpadding, width, heitgh, align, bgcolor,	
	Cell header	//	
	Cell	//	
	Row	//	

Figure 5 shows a generic web page structuration with tags indicated in Table 2. The web page structuration defines the web page layout to be scored.



Figure 5. Tags structuring the web page layout.

Appendix A is listed the LSTM algorithm adopted for the testing results. The LSTM model was implemented using the open-source library for deep-learning Keras [31]. The parameters analyzed for the UX are:

- User information (device, browser);
- The time of the user remaining on a web page;
- The number of clicks *n*<sub>c</sub>;
- Mouse movements in terms of coordinates.

In order to estimate these parameters has been adopted the library of [32]. The UX model indicated in Figure 2 was implemented by the following equation:

$$s_{\rm UX} = T^2 \cdot n_c, \tag{1}$$

where *T* is the time spent on the web page,  $n_c$  is the number of clicks that the user makes on the page and *s* is the score that will be multiplied with the feedback given by the user.

For the UX score is considered the following normalization, scaling the values between 1 and 10:

$$s_{\rm UX} = \frac{s_i - \min(S)}{\max(S) - \min(S)} * 9 + 1$$
(2)

where  $s_i$  is the *i*-th score and min(*S*) and max(*S*) are respectively the minimum and maximum of the score array. The normalized  $s_{UX}$  score is multiplied by the user feedback w by:

$$w(w_i)_{user} = w \cdot s_{\mathrm{UX}} \tag{3}$$

In Figure 6 is illustrated the graphical representation of the metrics adopted for the UX scoring (the limits values  $T_1$ ,  $T_2$ ,  $T_T$ ,  $n_{cT}$ , are calculated automatically by the algorithm, and are functions of the UX scoring following the normalization process).



**Figure 6.** Criteria of UX scoring  $(w(w_i)_{user})$  level based on the navigation time and on the number of clicks  $n_c$  (UX scoring matrix).

As for the CTR approach [21], Equations (1)–(3) defined that  $n_c$ , if associated with the time spent to visualize a web page, was a key parameter to estimate the web page impact. The visualization of a high-impact web page was characterized tendentially by a high number of clicks correlated to a high time spent to visualize the web page (see Figure 6). In order to measure the degree of attention with which the web page was evaluated, the time spent by the user in viewing the web page was associated with the number of clicks that are made during the evaluation of the page. This gives greater weight to the ratings of users who have reviewed the page more carefully. The relationship associating  $n_c$  and the time spent T to visualize a testing web page was provided by Equation (1): the "weight" of the T parameter is major then the click number one ( $S_{UX}$  quadratic behavior). The time variable indicates the interest and curiosity of the user towards the visualized page in order to ensure that the vote assigned to the page is not hasty. By means of the platform, the designer can train the network on a predefined number of templates. The created training dataset was split into 2 datasets: one to fix for the model training and one to validate it. Specifically, having an initial dataset of 100 samples, the first 5 examples were selected as examples from use to validate the system, while the other 95 to train it. Below was listed the used train and test commands.

sequences\_matrix\_train = sequences\_matrix[5 :]
sequences\_matrix\_test = sequences\_matrix[:5]

While the user is browsing the web pages, the platform records the behavior of the tester: clicks made, mouse movements, the time of the browsing session on the page.

Another approach combined to optimize the training dataset was the estimation of the outliers representing anomalous behaviors that can derive from various factors such as:

- The user has left browsing, thus the session on that page remains open but is not actually browsing;
- The user has a real difficulty in finding what he wants and has a non-optimal user experience.

The outliers were estimated by taking all UX data by checking if its distance from the mean was greater than the standard deviation multiplied by a constant (the constant empirically chosen is 1). Another important aspect was the prediction of the outliers. The estimation of outliers and of predicted outliers allows discarding in advance the templates expected to be of little significance, making the training of the LSTM model more efficient. This operation can be performed when the outliers were continuously predicted. Furthermore, the designer by the outlier analysis checks if there are web page features that slow down navigation. A method to predict the outliers was to use the ANN-MLP by means of Rapid Miner Graphical User Interfaces (GUIs) [33]. In Appendix B was illustrated the workflow executed for the ANN-MLP outlier prediction.

#### 3. Results

The training model was observed until 100 epochs, using the ADAM optimizer [34]. At the end of the hundred periods of training, the accuracy found on the test data exceeded 97%. The source codes of 100 web pages were collected to train the model; each of these was given feedback on the quality of the source code itself, using the 10-value rating scale ranging from 1 (poor quality) to 10 (excellent quality). By applying the Dropout technique [35], where randomly selected neurons are ignored during training, it is guaranteed a correct generalization of the system in processing new source codes not previously used. The Loss and the Accuracy trend of the training LSTM phase are illustrated in Figure 7a,b, respectively. By comparing the training and testing plots, it was observed that the training epoch number to achieve the best results was higher if compared with the testing epoch number: this was due to the initial training of the LSTM network and was a function of the complexity of the web page. The epoch number of the testing efficiency decreases, thus confirming the correct structuration of the training dataset and a certain similarity of the testing web pages with the training ones. The designer front-end interface of the developed platform is illustrated in Figure 8, where it is possible to distinguish the following fields:

- Command to select an HTML web page source for the training and for the testing data process;
- 2. Training command related to training process;
- 3. LSTM score visualizing the output of Figure 3;
- 4. User behavior analysis as separate evaluation of the UX (UX dashboard).

Figure 9a illustrates an example of user navigation embedding the GUI for the scoring (user feedback). The same figure also indicates an example of a post- processing reconstruction of a mouse hot map, indicating mouse movement distributions in the analyzed web page. Figure 9b illustrates the web page structuration of Figure 9a, embedding the related tags.

All data of mouse movement, number of clicks  $n_c$ , and navigation times T are stored in to the SQL database, and visualized by the UX dashboard of Figure 10a, also indicating the prediction of the UX trend for each web page as a regression. By this dashboard, it is possible to estimate the outliers. Thus, the platform provides additional information through graphical dashboard results, as the regression analysis between the score associated with mouse movements and the score of the actions performed (movements and clicks). Appendix C describes the code used for the tracking of mouse movement and time spent during the web page navigation, reporting data of an experimental test. This analysis allows understanding furthermore if the movement of the mouse, and the number of actions, are correlated or not. The outliers are also predicted by means of the ANN-MLP network. The forecast algorithm adopted a dataset divided into a training set (80%) and test set (20%). The ANN-MLP trains on the first UX values of a specific web page and performs a prediction on the last values. The prediction results are related to the next 21 days after the last day of the available data. As discussed in [36] about ANN-MLP Rapid Miner data processing, only a few days are more reliable. In the analyzed case, the first predicted days present similar values and consecutively are more reliable. Figure 10b illustrates an example of a prediction of a possible UX outlier in the next four days, where no anomalous behaviors are predicted.



**Figure 7.** LSTM performance of the training phase (the training dataset is the 95% of the whole dataset): (**a**) Loss trend versus epoch number. (**b**) Accuracy trend versus epoch number. LSTM performance of the testing phase: (**c**) Loss trend versus epoch number. (**d**) Accuracy trend versus epoch number.



Figure 8. "ANDUIA" platform GUI of the web designer interface.

# (a)



**Figure 9.** (a) Score user: example of the user interface for the scoring provided during his web page navigation. In the example is also reconstructed a mouse hot map, indicating a lot of mouse movements in the red region (in the color scale red is high, orange is average, and yellow is low). (b) web page layout structured by tags of Figure 9a.

A basic way to identify outliers is to consider the linear regression in the plane number of mouse movements and  $n_c$ , which are UX variables. Figure 10a indicates an experimental case of a web page navigation, finding an outlier, which is characterized to be far from the regression line. The outliers are not considered for the construction of the training dataset thus improving the performance of the LSTM algorithm. Table 4 shows the results concerning the testing of 21 web page templates, indicating for each template the user feedback, the  $n_c$ , the average time spent by the user T, the estimated outliers, the predicted outlier, and the final LSTM score. By reading the results, it was evident that templates 2 and 8 have the maximum score and some outlier to not consider in the training dataset because it could distort the results. Template 19, for its high time (60 s), can be an outlier candidate, but for its number of  $n_c$  it was evaluated as a moderately impact web page. In this case, if the ANN-MLP will always provide outlier conditions, the high-scored web pages will be redesigned, finding possible further improvement to avoid outliers. Furthermore, the outlier prediction also provides information about user reliability.



**Figure 10.** UX and outlier dashboard: (a) mouse movements versus the number of clicks  $n_{c}$ , (b) Example of ANN-MLP outlier prediction: the dates refer to the next four predicted days (Predicted Day 1 is the day after the calculus day).

Web Page (Template)	Score (User Feedback)	Clicks for Page [n <sub>c</sub> ]	Time T [s]	Estimated Outlier * by UX (Mouse Movement, Clicks and Time)	Predicted Outlier (ANN Confirming the Estimation)	LSTM Score (Based on UX Adjustment)
1	1	3	5	0	0	1
2	7	6	20	1	1	10
3	7	2	10	0	0	2
4	7	4	10	0	0	2
5	9	3	15	0	0	4
6	6	5	20	1	1	7
7	6	1	5	0	0	1
8	8	5	20	1	1	10
9	6	4	15	0	0	4
10	7	3	10	0	0	2
11	6	2	10	0	0	2
12	6	3	10	0	0	2
13	9	1	15	0	0	2
14	7	2	15	0	0	3
15	7	1	10	0	0	1
16	7	2	10	0	0	2
17	6	1	5	0	0	1
18	7	1	15	0	0	1
19	6	15	60	0	0	5
20	6	2	5	0	0	1
21	6	4	10	0	0	1

Table 4. Data of tests performed about UX.

\* 0: No Outlier, 1: Outlie.

Five users are adopted for the experimentation test. Each user checked four web pages. Only a user checked three web pages. The users have been assigned within the research team developing the project. The data have been collected during the testing phase of the project "*ANDUIA*". The network was characterized by two LSTM type layers and three DENSE type layers. All the layers consist of 256 neurons. In particular, in order to avoid overfitting the network during training, after the last LSTM layer and after the first DENSE type layer, DROPOUT (0.35) type layers have been added. The used loss function was the categorical cross-entropy providing the probability that results belong to each of the known classes. The sum of the outputs assigned to each class was 1. The activation function that best matched the categorical cross-entropy function was the softmax function. In fact, all its outputs were limited between 0 and 1, which made it excellent for modeling the probability distribution in a multi-class problem. The network has been trained over 100 epochs using a batch size of 95. The batch size was a hyperparameter that defined the number of samples to process before updating the internal model parameters.

#### 4. Discussion

# 4.1. Attribute Distribution Function

The applied methodologies define a new approach based on UX to find the best web page tags (attributes) structuration. The tags structuration is the web page tags structuration defining the web page layout. The LSTM algorithm is suitable to be applied for multiattribute data processing. In order to simply control the performance of the LSTM algorithm has been adopted the Keras Python libraries. The processed attributes are extracted automatically by the platform, which behaves as a data entry for the LSTM network. The scoring process is automated and takes into account the distribution functions g(x) of a specific attribute. By considering, for example, the UX scoring formulation of Figure 3, the implemented algorithms provide the distribution of the attributes insides or outside the Gaussian function g(x), thus identifying and estimating the scoring automatically as illustrated in Figure 11. The function g(x) will change with the training of the network and assumes other characteristics for the other attributes (by considering both the attributes *T* and  $n_{cr}$  two distribution functions are considered, such as g(x), and f(x)).



**Figure 11.** UX scoring and outliers condition (L: Low; A: Average; H: High):  $\mu$  indicates the main value, and  $\sigma$  the standard deviation. g(x) is the Gaussian distribution function related to a specific UX attribute.

## 4.2. Novelty Elements of the Proposed Approach and Possible Applications

The adopted approach, if compared with the metrics found in state of the art, provides the best web page layout defined by tags estimated by an LSTM algorithm "driven" by UX and user score. In Table 4 are summarized some common metrics found in state of the art and in the proposed research. It is observed that the used system is potentially able to extract all the information to estimate the metric reported in the Table 4, and, furthermore, introduces a new metric approach. This approach is also innovative because it automatizes the construction of the training dataset, by avoiding the scores indicated by outlier conditions: only the web pages labeled with correct scores will be considered for the model training. The proposed user interface of Figure 8 is particularly useful for E-commerce web page design, where it is typically required an accurate web analytics service. The specific analysis of the mouse movements in specific areas of a web page allows understanding if there are objects of impact to be replicated in the structuration of other new web pages. In particular, the extraction of other characteristics, such as the features of a photo or of an image, could further facilitate the design process. An interesting research topic is the implicit feedback of the user, such as the sequences of viewed items (item-sequences) [37]. In this direction, a further improvement of the proposed scoring is the evaluation of information about correlations between more tags of more pages viewed in sequence and image features characterizing the visual impact (colors, shapes, etc.). The LSTM network can also be adopted to predict the user attitudes [38], anticipating the design criteria of the web page layout. Other important aspects are the clustering of web users, in order to define the structural patterns of their usage behavior [39] and the prediction of the degree of page adjustment [40]. Important analyses can also be performed by matching the user clusters with the classified web page contents in order to address the design with a narrow precision. It is also important to distinguish the web pages of interesting for a broad user from the web pages suitable for a particular user cluster (a web page for chemical synthesis products cannot be of interest for a user with a low level of training), thus addressing the design on a specific user cluster. A possible improvement of the applied research for future works can be the pre-classification and the clustering of the users to find a possible correlation with the web page tags. The results of Table 4 represent an example of a website interaction metric adopting the approach described by the flowchart diagram of Figure 3. These results support to formulate for each web page template a design criterium, which is based on the following steps:

- 1. It is analyzed the final LSTM score based on UX adjustment and on the elimination of outlier conditions;
- 2. If the score is too low are analyzed the tags of the same web page to not consider for the realization of a new web page;
- 3. If the final score is high, are analyzed, and extracted the related tags concerning a major mouse interaction (movements and number of clicks); the "high impact" selected tags will also be adopted for the design of similar web page topics;
- 4. If the final score is average are analyzed only the best tags, by substituting the other ones with the best tags of similar web pages characterized by high scores.

An alternative approach to model memory in the information system is the Recursive Neural Networks (RNNs), suitable for approximating the probability distribution of the outputs [41]. Moreover, there are other techniques based on fractional modeling, useful to impress the user response correlated with visual inputs [42], and based on image vision approach [43], thus suggesting to analyze the web page visual impact deeply together with the user behavior. Another possible solution is to adopt ANN [44] for both outlier predictions and final scoring.

#### 4.3. Observations about the Training Dataset

It is observed that the metric of Table 1 summarizes a preliminary web page score for a web page classification used to construct the initial training dataset. A selection of web pages having different preliminary scores is considered as a training dataset. Besides, the tags listed in Table 2 defines the web page layout to be scored by the UX/LSTM proposed approach. Concerning LSTM training model, we observe that by varying the dimension of the training dataset, it is possible to furthermore improve the algorithm performance: By comparing Figure 12a,b with Figure 7a,b, it is possible to observe, for the case training dataset of the 80% of the whole dataset, a faster increase of the Accuracy and a faster decrease of the Loss versus epoch number. By doubling the training dataset from 100 to 200 web pages, are obtained a more stable convergence of the Loss (Figure 13a) and Accuracy (Figure 13b) for an epoch number major then 70. This confirms that by increasing the training dataset, the LSTM model is improved.



**Figure 12.** LSTM performance of the training phase (the training dataset is the 80% of the whole dataset): (**a**) Loss trend versus epoch number. (**b**) Accuracy trend versus epoch number.



**Figure 13.** LSTM performance of the training phase for a dataset of 200 web pages: (**a**) Loss trend versus epoch number. (**b**) Accuracy trend versus epoch number.

## 5. Conclusions

The paper discusses the results of a prototypal platform developed within the framework of a research project addressed on the ideation of innovative web page design tools. The platform prototype is suitable for web page designers and is based on the implementation of different functions. Specifically, the functions implemented in the platform are the UX evaluation about the web page navigation, the direct user feedback as user score of the specific navigated web page, the evaluation and the prediction of anomalous navigation conditions as outliers, and the final score obtained combining web page tags and UX. The LSTM network is adopted to process all the information, thus providing the final web page score for each analyzed page. The platform provides the scoring by means of an automated process and a GUIs, exploitable for the optimization of the training dataset. Moreover, the dashboards of UX, of outliers, and of LSTM score offer complete information about the performed scoring approach by distinguishing each parameter used for the calculus and useful for the design of web pages similar to those with the highest score. The Loss and the Accuracy demonstrate that the experimentation about web page scoring has been performed with good convergence. The innovation proposed in this paper is the methodology to evaluate web pages by analyzing: the HTML code used to create the page, the score assigned to each page by users, and the information collected about the user experience of the users during the evaluation process through the use of an LSTM neural network. The innovative methodology, summarized in Figure 3, consists of classifying the impact that the aesthetics of web pages has on users by analyzing the pages' source code through an LSTM network. Furthermore, the measurement of the user degree attention during the evaluation process as outlier conditions provides a good approach to build the training dataset correctly, and to control the user reliability. The used approach can be a good alternative to the image vision techniques and to the traditional metrics. For the specific tested web pages, the experimentation provided good performances for a low number of documents. Future work will be developed about the augmentation techniques applied to the proposed model by enhancing theoretical approaches about LSTM performance stability. The increase of the training dataset in AD is important for all cases where only a few data of the training dataset are available, as for more practical cases.

**Author Contributions:** Conceptualization, A.M., V.B., and D.G.; methodology, A.M.; software, D.G. and A.M.; validation, A.M. and A.M.G.; formal analysis, A.M.; investigation, A.M.; resources, A.M.G.; data curation, D.G. and A.M.; writing—original draft preparation, A.M.; writing—review and editing, A.M, A.M.G. and D.G.; visualization, A.M.; supervision, A.M.; project administration V.B., and A.M.G.; funding acquisition, A.M.G. and V.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not Applicable, the study does not report any data.

Acknowledgments: The proposed work has been developed within the framework of the industry project titled: '*Algoritmo di rilevamento di progettazione della User Interface: "ANDUIA"*'. Acknowledgments: A. Panella, C. Migliavacca, M. Bocedi, G. Bolzonella, A. Pauciullo, and E. Costa.

Conflicts of Interest: There are no conflicts of Interest.

#### Appendix A

This LSTM algorithm [45] was inserted in a prototype platform created with the Django framework accessible via browser.

```
import re
import glob
import numpy as np
import matplotlib.pyplot as plt
from keras.layers import LSTM, Activation, Dense, Dropout, Input, Embedding
from keras.optimizers import RMSprop, Adam
from keras.preprocessing.text import Tokenizer
from keras.preprocessing import sequence
from keras.models import Sequential
html_files = glob.glob("data/*.html")
x = []
y = []
for html_file in html_files:
     try:
         f = open(html_file, "r", encoding = "utf8")
         lines = f.readlines()
         f.close()
     except:
         f = open(html_file, "r")
         lines = f.readlines()
```

```
f.close()
          source = "
         for line in lines:
               source + = line
          tags = re.findall(r'<[^]+>', source)
          voto = html_file.split('.')[0].split('_')[1]
          text = "
         for t in tags:
              text + = t
          x.append(text)
         label = [0] * 10
         label[int(voto)] = 1
         y.append(label)
     y = np.asarray(y)
    print(len(html_files), 'files processed')
     max words = 10,000
     max \ len = 5000
     tok = Tokenizer(num_words = max_words)
     tok.fit_on_texts(x)
     sequences = tok.texts_to_sequences(x)
     sequences_matrix = sequence.pad_sequences(sequences, maxlen = max_len)
     model = Sequential()
     model.add(Embedding(max words, 64, input length = max len))
     model.add(LSTM(units = 256, return_sequences = True))
     model.add(LSTM(units = 256))
     model.add(Dropout(0.35))
     model.add(Dense(256, activation = 'relu'))
     model.add(Dropout(0.35))
     model.add(Dense(256, activation = 'relu'))
     model.add(Dense(units = 10, activation = 'softmax'))
     model.compile(optimizer = Adam(), loss = 'categorical_crossentropy', metrics = ['accu-
racy'])model.summary()
     sequences_matrix_train = sequences_matrix[:95]
     sequences_matrix_test = sequences_matrix[95:]
    y_train = y[:95]
     y_{test} = y[95:]
    history = model.fit(sequences_matrix_train, y_train, epochs = 100, batch_size = 95)
    plt.plot(history.history['loss'])
    plt.show()
    plt.plot(history.history['acc'])
    plt.show()
     preds = model.predict(sequences_matrix_test)
    print(y_test)
```

# Appendix B

This ANN MLP network has been implemented by the Rapid Miner workflow of Figure A1. The workflow is able to predict outliers by processing as input the estimated outliers, the  $n_c$ , the user score, and the *time* attributes.

In Figure A2 is shown the graph model of the adopted ANN MLP network. The model is constructed by the block "Apply Model" of Figure A1. The network is characterized by five input nodes, indicating the five attributes considered for the estimation of the predicted outlier (date, estimated outlier, score,  $n_c$ , and time), and by one hidden layer constituted by four hidden nodes and by one threshold node. The labeled input is the estimated outlier. Therefore, the prediction is performed about this parameter. The input



dataset (clicks, navigation time, and mouse movements) is partitioned as 80% of training dataset and 20% of the testing dataset.

Figure A1. ANN MLP Rapid Miner workflow predicting outliers.

# Input Hidden 1 Output



Figure A2. ANN MLP graph model.

# Appendix C

The user behavior library is included in the HTML page to be monitored via a js code: <*script>* 

```
}
                  });
                }
              });
         </script>
, where is set to record the movements of the mouse and the time spent on the page. As
output, the script provides a data file in json format of the type:
     #### Sample Response Object
    {
            "userInfo": {
                "appCodeName": "Mozilla",
                "appName": "Netscape",
                "vendor": "Google Inc.",
                "platform": "MacIntel",
                "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWe-
bKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.134 Safari/537.36"
            },
            "time": {
                   "totalTime": 27,
                   "timeOnPage": 19
            },
            "clicks": {
                   "clickCount": 4,
                   "clickDetails": [
                       {
                          "timestamp": 1440001422756,
                          "node": "<input type = \"text \" name = \"email \" placeholder =
\"E-mail address \">",
                          "x": 849,
                          "u": 206
                       },
                          "timestamp": 1440001438535,
                          "node": "<div class = \"ui fluid large teal submit button\" id =
\mbox{"myActionItem}\">Login</div>",
                          "x": 893,
                          "y": 312
                       },
                       {
                          "timestamp": 1440001454767,
                          "node": "<input type = \"text \" name = \"email \" placeholder =
\"E-mail address \">",
                          "x": 853,
                          "y": 198
                       },
                       {
                          "timestamp": 1440001457409,
                          "node": "<div class = \"ui fluid large teal submit button\" id =
\"myActionItem \">Login </div>",
                          "x": 880,
                          "y": 321
                       }
                  ]
              },
```

All the coordinates can be groped to generate a mouse movement "hot" map of the web page layout.

# References

- 1. Gonzalez-Rodriguez, M.; Diez-Diaz, F.; Vidau, A. Remote navigability testing using data gathering agents. In Proceedings of the 4th International Conference on Universal Access in Human Computer Interaction (UAHCI), Beijing, China, 22–27 July 2007.
- 2. Obendorf, H.; Weinreich, H.; Hass, T. Automatic support for web user studies with SCONE and TEA. In Proceedings of the Conference on Human Factors in Computing Systems (CHI), Vienna, Austria, 24–29 April 2004.
- 3. Ivory, M.; Hearst, M. Statistical profiles of highly-rated web sites. In Proceedings of the Conference on Human Factors in Computing Systems (CHI), Minneapolis, MN, USA, 20–25 April 2002.
- 4. Katsanos, C.; Tselios, N.; Avouris, N. InfoScent evaluator: A semi-automated tool to evaluate semantic appropriateness of hyperlinks in a web site. In Proceedings of the OZCHI 2006, Sydney, Australia, 20–24 November 2006; pp. 373–376.
- 5. Chiew, T.K.; Salim, S.S. Webuse: Website usability evaluation tool. *Malays. J. Comp. Sci.* 2003, 16, 47–57.
- Cybulski, P.; Horbiński, T. User Experience in Using Graphical User Interfaces of Web Maps. ISPRS Int. J. Geo-Inf. 2020, 9, 412. [CrossRef]
- 7. Massaro, A.; Vitti, V.; Mustich, A.; Galiano, A. Intelligent Real-time 3D Configuration Platform for Customizing E-commerce Products. *Int. J. Comp. Grsph. Animat. (IJCGA)* **2019**, *9*, 13–28. [CrossRef]
- 8. Atterer, R.; Wnuk, M.; Schmidt, A. Knowing the user's every move—User activity tracking for website usability evaluation and implicit interaction. In Proceedings of the International World Wide Web Conference Committee (IW3C2), Edinburgh, Scotland, 22 May 2006.
- 9. Liu, L.; Chen, J.; Song, H. The Research of web mining. In Proceedings of the 4th World Congress on Intelligent Control and Automation, Shanghai, China, 10–14 June 2002.
- 10. Pal, S.; Talwar, V.; Mitra, P. Web Mining in soft computing framework: Relevance, state of the art and future directions. *IEEE Trans. Neural Netw.* **2002**, *13*, 1163–1177. [CrossRef]
- 11. Filippi, S. PERSEL, a Ready-to-Use PERsonality-Based User SELection Tool to Maximize User Experience Redesign Effectiveness. *Multimodal Technol. Interact.* 2020, *4*, 13. [CrossRef]
- 12. Alonso-Virgós, L.; Rodríguez Baena, L.; Pascual Espada, J.; González Crespo, R. Web Page Design Recommendations for People with Down Syndrome Based on Users' Experiences. *Sensors* **2018**, *18*, 4047. [CrossRef]
- 13. Oyibo, K.; Vassileva, J. The Effect of Layout and Colour Temperature on the Perception of Tourism Websites for Mobile Devices. *Multimodal Technol. Interact.* **2020**, *4*, 8. [CrossRef]
- 14. Buber, E.; Diri, B. Web Page Classification Using RNN. Proc. Comp. Sci. 2019, 154, 62–72. [CrossRef]
- 15. Chen, R.C.; Hsieh, C.H. Web page Classification based on a Support Vector Machine using a Weighted Vote Schema. *Exp. Syst. Appl.* **2006**, *31*, 427–435. [CrossRef]
- 16. Wang, C.; Zhao, S.; Kalra, A.; Borcea, C.; Chen, Y. Webpage Depth Viewability Prediction Using Deep Sequential Neural Networks. *IEEE Trans. Know. Data Eng.* **2019**, *31*, 601–614. [CrossRef]
- 17. Spiliotopoulos, K.; Rigou, M.; Sirmakessis, S. A Comparative Study of Skeuomorphic and Flat Design from a UX Perspective. *Multimodal Technol. Interact.* **2018**, *2*, 31. [CrossRef]
- 18. Khrais, L.T. Role of Artificial Intelligence in Shaping Consumer Demand in E-Commerce. Future Internet 2020, 12, 226. [CrossRef]
- 19. Li, R.; Sun, T. Assessing Factors for Designing a Successful B2C E-Commerce Website Using Fuzzy AHP and TOPSIS-Grey Methodology. *Symmetry* 2020, 12, 363. [CrossRef]
- Chaudhary, N.; Roy Chowdhury, D. Data Preprocessing for Evaluation of Recommendation Models in E-Commerce. *Data* 2019, 4, 23. [CrossRef]
- 21. Saura, J.R.; Palos-Sánchez, P.; Cerdá Suárez, L.M. Understanding the Digital Marketing Environment with KPIs and Web Analytics. *Future Internet* 2017, 9, 76. [CrossRef]
- 22. Kaushik, A. Web Analytics 2.0: The Art of Online Accountability and Science of Customer Centricity; John Wiley & Sons: Hoboken, NJ, USA, 2009.

- 23. Veglis, A.; Giomelakis, D. Search Engine Optimization. Future Internet 2020, 12, 6. [CrossRef]
- 24. López García, J.J.; Lizcano, D.; Ramos, C.M.; Matos, N. Digital Marketing Actions That Achieve a Better Attraction and Loyalty of Users: An Analytical Study. *Future Internet* 2019, *11*, 130. [CrossRef]
- Huang, G.; Chen, Q.; Deng, C. A New Click-Through Rates Prediction Model Based on Deep&Cross Network. *Algorithms* 2020, 13, 342. [CrossRef]
- 26. Seggie, S.H.; Cavusgil, E.; Phelan, S.E. Measurement of return on marketing investment: A conceptual framework and the future of marketing metrics. *Ind. Mark. Manag.* 2017, *36*, 834–841. [CrossRef]
- 27. Fagan, J.C. The suitability of web analytics key performance indicators in the academic library environment. *J. Acad. Librariansh.* **2014**, *40*, 25–34. [CrossRef]
- 28. Wilson, R.D. Using web traffic analysis for customer acquisition and retention programs in marketing. *Serv. Mark. Q.* **2004**, *26*, 1–22. [CrossRef]
- 29. Booth, D.; Jansen, B.J. A Review of Methodologies for Analyzing Websites. In *Web Technologies: Concepts, Methodologies, Tools, and Applications;* IGI Global: Hershey, PA, USA, 2010; pp. 145–166. [CrossRef]
- Toleu, A.; Makazhanov, A.; Tolegen, G. Character-based Deep Learning Models for Token and Sentence Segmentation. In Proceedings of the 5th International Conference on Turkic Languages Processing (TurkLang), 18–21 October 2017.
- 31. Keras. Available online: https://keras.io/ (accessed on 5 February 2021).
- 32. User Behavior Library. Available online: https://github.com/shnere/user-behavior (accessed on 4 February 2021).
- Massaro, A.; Maritati, V.; Galiano, A. Data Mining Model Performance of Sales Predictive Algorithms Based on RapiMiner Workflow. Int. J. Comp. Sci. Inf. Technol. (IJCSIT) 2018, 10, 39–56. [CrossRef]
- 34. Yi, D.; Ahn, J.; Ji, S. An Effective Optimization Method for Machine Learning Based on ADAM. Appl. Sci. 2020, 10, 1073. [CrossRef]
- 35. Lee, S.; Chung, J.Y. The Machine Learning-Based Dropout Early Warning System for Improving the Performance of Dropout Prediction. *Appl. Sci.* **2019**, *9*, 3093. [CrossRef]
- Massaro, A.; Vitti, V.; Galiano, A. Model of Multiple Artificial Neural Networks oriented on Sales Prediction and Product Shelf Design. Int. J. Soft Comput. Artif. Intell. Appl. (IJSCAI) 2018, 7, 1–19. [CrossRef]
- 37. Wang, B.; Ye, F.; Xu, J. A Personalized Recommendation Algorithm Based on the User's Implicit Feedback in E-Commerce. *Future Internet* **2018**, *10*, 117. [CrossRef]
- Alsulami, M.M.; Al-Aama, A.Y. Employing Behavioral Analysis to Predict User Attitude towards Unwanted Content in Online Social Network Services: The Case of Makkah Region in Saudi Arabia. *Computers* 2020, 9, 34. [CrossRef]
- 39. Yu, X.; Li, M.; Kim, K.A.; Chung, J.; Ryu, K.H. Emerging Pattern-Based Clustering of Web Users Utilizing a Simple Page-Linked Graph. *Sustainability* **2016**, *8*, 239. [CrossRef]
- 40. Matošević, G.; Dobša, J.; Mladenić, D. Using Machine Learning for Web Page Classification in Search Engine Optimization. *Future Internet* **2021**, *13*, 9. [CrossRef]
- Martínez-García, M.; Zhang, Y.; Suzuki, K.; Zhang, Y.-D. Deep Recurrent Entropy Adaptive Model for System Reliability Monitoring. *IEEE Trans. Ind. Inform.* 2021, 17, 839–848. [CrossRef]
- 42. Martínez-García, M.; Zhang, Y.; Gordon, T. Memory Pattern Identification for Feedback Tracking Control in Human–Machine Systems. *Hum. Factors* 2021, 63, 210–226. [CrossRef] [PubMed]
- Reinecke, K.; Yeh, T.; Miratrix, L.; Mardiko, R.; Zhao, Y.; Liu, J.; Gajos, K.Z. Predicting Users' First Impressions of Website Aesthetics with a Quantification of Perceived Visual Complexity and Colorfulness. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'13), Paris, France, 27 April–2 May 2013; pp. 2049–2058. [CrossRef]
- 44. Cascardi, A.; Micelli, F.; Aiello, M.A. Analytical Model based on Artificial Neural Network for Masonry Shear Walls Strengthned with FRM Systems. *Compos. Part B Eng.* **2016**, *95*, 252–263. [CrossRef]
- 45. Massaro, A.; Maritati, V.; Giannone, D.; Convertini, D.; Galiano, A. LSTM DSS Automatism and Dataset Optimization for Diabetes Prediction. *Appl. Sci.* **2019**, *9*, 3532. [CrossRef]