International Journal of Reliability, Quality and Safety Engineering
Vol. 21, No. 5 (2014) 1450026 (48 pages)

DOI: 10.1142/S0218539314500260

World Scientific
www.worldscientific.com

TRANSFER FUNCTION MODELING IN WEB SOFTWARE FAULT PREDICTION IMPLEMENTING PRE-WHITENING TECHNIQUE

SUBHASHIS CHATTERJEE* and ARUNAVA ROY†

Department of Applied Mathematics
Indian School of Mines Dhanbad
Jharkhand 826004, India
**chatterjee_subhashis@rediffmail.com*
†*royism.arunava@gmail.com*

Received 15 August 2013
Revised 20 June 2014
Accepted 29 July 2014
Published 17 September 2014

The issue of Web reliability is gaining importance, as different Web-based applications are getting popularity with time. In order to enhance the reliability of a Web system, the Web administrator have to determine if there exists any relationship or correlation among different Web workload characteristics and the errors having an impact on the reliability of the Web system, so that he will be able to predict them accurately. It may not be possible to establish a generalized relationship among different Web workload characteristics. Hence, in this paper, we have performed principal component analysis (PCA) to check whether different Web workload characteristics, for particular Web software are correlated or not. Then, we have proposed a transfer function based model for Web software fault prediction. Also, we have used the pre-whitening technique to eliminate the noise present in the data for developing an efficient transfer function based model to predict the cumulative occurrences of different Web failures having an impact on the reliability of the Web software.

Keywords: Web software reliability; time series; transfer function; pre-whitening technique; HTTP; HTTP logs; Web server.

Nomenclature

ACF : Autocorrelation function.
PACF : Partial autocorrelation function.
AR(p) : Autoregressive process of order p, where, p is a non-negative integer.
MA(q) : Moving average process of order q where, q is a non-negative integer.
ARMA(p, q) : Autoregressive moving average process of order (p, q).

ARIMA(p, d, q) : Autoregressive integrated moving average process with order (p, d, q).
PCA : Principal component analysis.
URI : Uniform resource identifier.
URL : Uniform resource locator.
HTTP : Hypertext transfer protocol.
HTML : Hypertext markup language.

1. Introduction

The issue of reliability is gaining importance with time for Web system due to the popularity of different Web based applications. The need of highly reliable Web systems is increasing as different companies, research organizations, academic and other institutions are switching to online applications for their business and other activities. In order to enhance the reliability of a Web system, some methods are required to measure the current reliability of the system. To estimate the reliability of a system various metrics are available like, MTTF, MTTR, MTBF, failure intensity etc. One such important measure is the number of faults present in the system. There are several models available in the literature to measure the reliability of general software.[1–12] Unfortunately, these models cannot be applied directly to the Web software due to its some specific nature.[13–20] Specific characteristics those make Web workload measurement different from the traditional software systems are given below:

- Massiveness and diversity: Web applications provide cross-platform universal access to Web resources for everyone with an Internet access. Again the Web system can interact with several different external systems consisting of massive user population, diverse hardware/software configurations, and the varied usage patterns need to be reflected in the selected workload measures.
- Document and information focus: Traditional workload concentrates on the computational focus but on the other hand the Web system has a document and information focus. Hence, the traditional workload measurement may not be applicable in case of a Web system.

Web software reliability can be defined as the probability of failure free Web operation completion.[13] To measure the reliability of Web software, Kallepalli and Tian,[13] Tian *et al.*,[14] Ma *et al.*[17] and Popstojanova *et al.*[16] have defined the following four different workload characteristics or measures for Web applications.

- Number of hits: It is the most obvious Web workload measure due to the following reasons:
  (a) Each hit represents a specific activity associated with Web access.
  (b) Each entry in the access log corresponds to a particular hit. Hence, it can be easily extracted.

- The number of transferred bytes: It may be an alternative Web workload measure present in the log file of Web software. Tian *et al.*[14] have proposed that it has finer granularity than hit count measure.
- Number of users: It can be an alternative measure for Web workload, meaningful to the organization that maintains Websites and several services at the user level. For finding it, we treat each unique IP in the access log as a unique user. It is not true in all cases. For example, if a proxy server exists between the user and the server, IP address of the proxy will be there in the access log of the server rather than the address of the originating machine. Hence, this measure is suffering from coarse granularity.
- Number of sessions: Session is defined as a sequence of requests from the same user during a single visit to the Website. A session begins when the user issues a request for a particular page on a Website.[13–19] It may be a better measure of overall workload than the number of users, because big access gaps are typically associated with changes of users or non-Web related activities by the same user. But in many cases, it may provide some confusing results, which have been discussed later.

In the previous studies,[13–20] the authors mainly emphasize on finding the relationship among different Web workload characteristics and found no straightforward relation among them. For example, Popstojanova *et al.*[16] have established that the reliability based on the session work load is lower than the reliability based on the hit-count for the Websites under their studies which may not be a general result. A session begins when the user generates a request for a particular Web page on a Website.[13–20] When the user Web browser receives the response, it simply parses the file and generates response for all the embedded files. Therefore, a session may be present when the client requests for a single Web page, e.g., accessing a home page involves requesting the HTML page and then making further requests for all the embedded images and documents. Hence, corresponding to a particular hit, multiple sessions can be generated which may not be the general situation. Similarly, a large number of hits may guarantee a large number of sessions. For example, if a large number of users having different IP addresses, from different corners of the world are making a large number of requests for a particular Web page of a Website, we may find a huge number of hits and sessions simultaneously. Hence, there is no straightforward relationship between the number of sessions and hit-count. Based upon circumstances, the above statement may be incorrect. We can have a similar conclusion in the case of session-count and user-count as a single user may or may not create several sessions. Therefore, better session reliability may not guarantee better user reliability and vice versa. Similarly, a large number of hits may not imply a huge number of users, as a single user can make a very large number of requests. In case of other Web workload characteristics we can draw a similar conclusion. Hence, in general, we may not be able to establish relationship among different Web workload characteristics. In those cases, there must be a joint

contribution of all the uncorrelated Web workload characteristics on the combined occurrences of all the Web failures having an impact on the reliability of the Web software.

Tian *et al.*[14] have established that among all the measures of the reliability of the Web software, session reliability is the better measurement. The limitation of session-count lays with the use of dynamic IP's or proxy servers at Internet Service Providers (ISPs). For example, if user A visits a site and immediately leaves, but user B comes to the site within the time frame defined, using the same source IP address, both visitors will be counted as one visitor. If, on the other hand, user A visits the site then leaves the system for more than the defined time frame (de-facto standard 30 min), then return to the site and pull up a second page, he would be counted as two users. Hence, in those cases, it may not be appropriate to say that, only the number of session is responsible for the occurrences of different Web errors having an impact on the reliability of the Web software. The shortcomings of bytes transfer workload is that, from access and error logs we cannot get the total size of the file rather the amount of transferred bytes. The user-count workload has some limitations too. Here, we address the inaccuracies introduced by using the IP addresses as surrogate for users. For example, if a proxy server exists between the user and the server, the IP address in the Web access log will be the address of the proxy, rather than the address of the originating machine. Furthermore, even when a unique IP address is assigned to a single machine, it may be a machine available for public access, such as for example machines in the university laboratories. The hit-count is also suffering from some disadvantages as it becomes misleading if the workload represented by individual hit shows high variability.[13–20] From the above study, it is quite clear that, all the uncorrelated Web workload characteristics are responsible for the occurrences of different Web failures.

Hence, to determine that, if any correlation among different Web workload characteristics exists, initially the PCA[21,22] has been carried out here. It is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possible correlated variables into a set of values of linearly uncorrelated variables called principal components (PCs).[22] Its fundamental idea is to reduce the dimensionality of a data set consisting of a large number of interrelated variables. This is achieved by transforming to a new set of variables, the principal components (PCs), which are linearly uncorrelated, and which are ordered such that the first few retain most of the variation present in all the original variables. For this purpose, the PCA of the Web workload characteristics has been carried out to find that if there is any correlation between them or not. Different probabilistic and data driven approaches have been used for software reliability analysis.[1–12] There are various models available in data driven approaches like time series, neural network, etc. The advantage of data driven approach is the models are assumption free and they can be used for modeling of any types of software failure data. Using conventional and fuzzy time series approach, various researchers[10,11,23,24] have developed different types of time series based software reliability models like

AR, MA, ARMA, ARIMA and transfer function based models. Among these AR, MA, ARMA, ARIMA are univariate models, i.e., the output is dependent on single input or independent variable. On the other hand the transfer function models are not only assumption free but also in this approach the output can be modeled as a function of multiple independent input variables. Transfer function model was first proposed by Singpurwalla.[23]

Previously, we have shown that, all the uncorrelated Web workload characteristics have contributions on the occurrences of different Web errors. However, all the existing Web software reliability models[13–20] use only the Nelson's model (a static reliability model) for estimating the reliability with respect to individual Web workload characteristics. It is unrealistic as in many cases almost equal influences of all the Web workloads can be found. Keeping this in mind, in the present paper, an efficient statistical transfer function based Web error prediction model has been developed that can forecast the occurrences of different Web errors having an influences on the reliability, considering the impact of all the uncorrelated Web workload characteristics. But in practice, the output may not always be a deterministic function of the input variables. The output is often disturbed by noise or has its own dynamic structure. Since, the noise component and the input variables might be serially correlated or dependent would in general may not provide consistent estimate of the output variable. Hence, in order to predict the output variable more accurately, we have applied the pre-whitening technique. Moreover, the performance of the proposed model has been compared with a MANOVA based forecasting model as well as some nonlinear regression models. But, in every case, the proposed model shows its better predictive accuracy. In short, the proposed model can be described as follows:

(1) Extract the occurrences of different Web error codes, having an influence on the reliability of the Web software, different Web workload characteristics from the HTTP log files.
(2) Find the uncorrelated Web workload characteristics.
(3) Form the transfer function model, considering the uncorrelated Web workload characteristics as inputs and the occurrences of different Web errors having an influence on the reliability of the Web software as output.

This proposed method can be applicable in forecasting the occurrences of every individual Web errors having an influence on the reliability of the Web software. But, for this purpose, we have to implement the proposed method for all the Web errors having impact on the reliability, which is a lengthy procedure and will increase the length of the paper. Hence, for simplicity, in this paper, the authors have implemented their methods for predicting the cumulative occurrences of all the Web errors having an influence on the reliability of the Web software.

The rest of the paper is organized as follows: Section 2 represents the research methodology which consist of a brief description about different Web testing tools available, log files analysis, recording the Web server's workloads, session tracking

procedure, determination of session threshold, some frequently occurred Web error codes and the proposed method. Section 3 describes the overview of the Websites in this study. Results and discussions are given in Sec. 4. Section 5 represents the performance analysis of the proposed method. Section 6 concludes the work.

2. Research Methodology

In this section, we have highlighted the steps involved in our research work. The highlights of the steps are given as follows:

2.1. *Different Web testing tools*

In this subsection, we have discussed about some Web testing tools, which can be used for functionality testing or analyzing the HTTP log files (access and error logs) of Web software. Some of them are tabulated in Table 1.

HTML valuators such as Weblint and W3C Validator listed in Table 1 can parse HTML files and check their conformance to relevant language specifications and document standards. Link checking tools like Net Mechanic listed in Table 1 can be used to check the entire site for broken links. Tools like Doctor HTML listed in Table 1 can be used to check input types and variable names in various forms. Java applets, which work on the clients' side, or other Java applications, which work on the server side, need to be tested. Tools such as TCAT for Java listed in Table 1 can be used to perform coverage-based testing for Java components. Tools such as SilkTest or Visual Test listed in Table 1 can be used to support multiple types of functionality testing. Analog and FastStats, mentioned in Table 1 can be used for HTTP log file analysis purpose.[13]

2.2. *Analysis of log files (access and error logs)*

We need to analyze the HTTP logs (access and error logs) as proposed by Kallepalli and Tian,[13] Tian *et al.*,[15] Popstojanova *et al.*[16] and extract all the factors, i.e., all the errors having an impact on the reliability of the Web system (minimizing main factors), number of sessions (maximizing main factor), etc.

Table 1. Existing Web testing tools and Web log analyzers.

Tools	Online information
Web Testing Tools	
Weblint	www.weblint.org
W3C Validator	validator.w3.org
Net Mechanic	www.netmechanic.com
Doctor HTML	www2.imaginewarecom/RxHTML
TCAT for JAVA	www.soft.com/Products/Coverage.msw/tcatj.html
Log files Analyzers	
Analog	www.mach5.com/fast
FastStats	www.analog.cx/

2.2.1. *Recording Web servers workloads*

Web servers can be configured to record (in an access log) information about all of the requests and responses processed by the server.[13–20] Each line from the access log contains information on a single request for a document. The log entry for a normal request is of the form:

hostname — [dd/ mmm/ yyyy: hh: mm: ss tz] request status bytes (Ref. 18). From each log entry, it is possible to determine the name of the host machine making the request, the time that the request was made, and the name of the requested document. The entry also provides information about the server's response to this request, such as if the server was able to satisfy the request (if not, a reason why the response was unsuccessful is given) and the number of bytes transmitted by the server, if any. The access log provides most of the data needed for workload characterization studies of Web servers. However, they do not provide all of the information of interest. For example, the log entries tell only the number of bytes transferred for a document, not its actual size[13–20] there is no record of the elapsed time required for a document transfer[13–20]; and there is no information on the complete set of files available on the server, other than those documents that are accessed in the logs.[13–20] Furthermore; there is no record of whether a file access was human-initiated or software-initiated (e.g., by a Web crawler or Web robot), or what caching mechanisms, if any, are in place at the client and/or the server.

One point must be kept in mind that the log files may contain requests from robots and other automated systems that should be removed as they are not actual requests from Web users. Automated systems are classified as systems that repeatedly request a resource from the Website after a predefined period of time. Several techniques to identify them can be used by Web administrators to remove automated requests. Most well-known robots have a signature line that is included with every request as part of the USER AGENT field in the log file, especially HTTP error logs of the corresponding Web server.

2.2.1.1. Session tracking procedure

Nowadays, we are using IPv4, i.e., a 32 bit addressing style, which implies that in a network there can be maximum 2^{32} unique computers. With the current explosion in the number of Internet users, the total amount of IP addresses available is shrinking rapidly. To avoid this situation, IPv6 addressing style has been adopted. It uses a 128-bit address, allowing 2^{128} or approximately 3.4×10^{38} unique addresses, i.e., more than 7.9×10^{28} times as many as IPv4. Thus, many methods now exist that allow one public IP address to be used for a group of machines; some of these methods include proxy servers and personal routers. Since the original study suggests counting one unique IP as a user, there is a strong possibility that this "user" is actually a group of users. As personal routers and proxy servers become more dominant this issue is also becoming more prominent.

Session tracking (for those who have not heard of it) is a concept which allows us to maintain a relation between two successive requests made to a server on the Internet. Whenever a user browses any Website, he uses HTTP (the underlying protocol) for all the data transfers taking place. HTTP is a stateless protocol. When a user requests for a page the server returns that Web page to the user. When the user once again clicks on a new link the server once again sends the new page that was requested. The server (because of the use of HTTP as the underlying protocol) has no idea that these two successive requests have come from the same user. The server is not at all bothered about who is asking for the pages. All it does it return the page that has been requested. This is exactly what stateless means. There is no connection between two successive requests on the Internet.

There are many instances where some sort of connection is required between two requests made by the same user. Since all transfers on the WWW use HTTP at the lowest level this sort of connection cannot be made. For example if you are at a Website buying books online, then you may add books to your Cart and continue searching for more books. Every time you click on a new page your old selected books in the Cart should not disappear. In case you use the default way the WWW works, then since two successive requests (by the same user) have no connection, there would be no books in your Cart every time you click on a new link, which means every click would be considered as a separate request and none of them having relation to previous request. Thus as you browse, all the information that relates to you should be maintained and should be carried on as you browse more and more. Your previous Shopping Cart contents should be present when you want to add a new book to the Cart. This is what session tracking enables us to do. It let us to maintain an active session as long as we are browsing and it gives HTTP a sort of new quality with every successive request having some relation to previous requests within the same session.

Session tracking is so common that we may not even realize that it is present. It is used on almost every possible site you visit on the net. For example, at Hotmail once you enter your username and password and you reach your inbox, had there been no session tracking then every time you click on a particular link in your inbox, you would be asked for your password. This would be the case since there would be no way to understand that the one who had originally entered his username-password is the same person who is currently asking for more pages. Session tracking allows you to store the information that you have successfully logged in and this information would be checked every time you do anything within your inbox. Thus you would not be asked to enter your password with every click.

The session count also suffers from the similar problem as that of user count because "one session" may actually be several sessions from several different users who are sharing the same public IP. Thus, a methodology needs to be developed to distinguish different users before accurate reliability analysis can be performed.

A unique characteristic of Web workload is the concept of session. A session is defined as a sequence of requests from the same user during a single visit to the

Website.[13–20] Tracking the overall Website sessions is the best and most accurate way to determine the site's performance. There are certain ways to implement the session tracking. Some of the most popular ways are discussed as follows:

(i) Hidden Fields in Forms
(ii) URL Rewriting
(iii) Cookies.

(i) Hidden Fields in Forms

This is the simplest and extremely useful way to implement session tracking. With the help of an online book buying Website example, we have explained this concept.

In case of an online book buying Website, a user can select books and click on an Add to Cart Submit button. A sample code for such a page is shown below. Remember this is just what the code may look like and not the exact page. We should try to understand the logic rather than focus on the syntax. Also remember that these are all dynamic pages being generated using some language such as JSP.

$< b > Search\ results\ for\ books < /b >$
$< form\, method = \text{“}post\text{”}\ action = \text{“}serverprogram.jsp\text{”} >$
$< input\ type = checkbox name = \text{bookID} value = 100 >$
Java Servlet Programming
$< br >$
$< input\, type = \text{checkbox} name = \text{bookID} value = 101 >$
Professional JSP
$< br >$
$< input\, type = \text{submit} name = \text{Submit} value = \text{Add to Cart} >$
$< br >$
$< /form >$

Suppose a page similar to the above one was generated when the user searched for some books. The above page has only two search results ("Java Servlet Programming" and "Professional JSP"). There is a form with two checkboxes, each next to the name of a book and a Submit button to add any selected books to the Cart.

Now suppose the user clicks on the checkbox next to book named "Java Servlet Programming", and then clicks on the Submit button. Note that the value of a checkbox is used in this case to store the bookID. Generally when there are many checkboxes each representing one-of-many kind of entity then the value for that checkbox differentiates between all of them. In our case since all the checkboxes represent books, each value represents a different bookID and thus a different book (one book-of-many books).

Now coming back to the point, in case the user checked the checkbox next to the book named "Java Servlet Programming" and then clicked the Submit button, the contents of the form are all bundled together and sent to the server side program. In our case the program is named serverprogram.jsp. Now suppose at any further

instant when the same user is searching for more books then on a search result he might be presented with page such as the one shown below. Remember that he has already selected a book previously. So that book should be present in his Cart and now he would like to add more books.

$< b > Search\ results\ for\ books < /b >$
$< form\ method = \text{“}post\text{”}\ action = \text{“}serverprogram.jsp\text{”} >$
$< input\ type = \text{“}hidden\text{”}\ name = \text{“bookID”}\ value = \text{“100”} >$
$< input\ type = \text{checkbox} name = \text{bookID} value = 150 >$
Teach yourself WML Programming
$< br >$
$< input\ type = \text{checkbox} name = \text{bookID} value = 160 >\ Teach\ yourself\ C{+}{+}$
$< br >$
$< input\ type = \text{“}submit\text{”}\ name = \text{“}Submit\text{”}\ value = \text{“}Add\ to\ Cart\text{”} >< br >$
$< /form >$

The new search result produced once again two new books. One book named “Teach yourself WML Programming” with a bookID of 150 and another book named “Teach Yourself C++” with a bookID of 160. So a form was generated with the names of these two books and with two checkboxes so that the user may select any of these books and add them to the Cart. But there is one more important thing in the form that was generated. There is a hidden input field named bookID and having a value of 100. We might have noticed that 100 was the bookID of the book named “Java Servlet Programming” which the user had initially selected. This line describing a hidden input does not make any difference on the HTML page displayed in the browser. It would be totally invisible to the user. But within the form it makes a lot of difference. This way when the user keeps adding more and more books, there would be many hidden input fields each with a different value, each representing a previously selected book. When this form is submitted to the server side program, that program would not only fetch the newly selected checkboxes (newly selected books) but also these hidden fields each representing a previously selected book by that user. Note that all the input fields have the same name bookID but their values are different. Within the server side program we would simply expect a parameter called bookID which would be an array with different values. We can extract all the values and then use them as required. It is the job of the server side program to add these lines indicating hidden fields whenever it generates a new page. Once again, the main concept to be understood is that a hidden field displays nothing on the HTML page. So the user who is browsing the page sees nothing unusual, but the value associated with these hidden fields can be used to hold any kind of data that you want.

The disadvantage of session tracking is that in case we do not want the user to know what information is being passed around to maintain a session (in case that information is somewhat vital, maybe a password or something) then this method

is not the best one since the user can simply select to View the Source of the HTML page and get to see all the hidden fields present in the Form.

(ii) URL Rewriting

This is another popular session tracking method used by many. But it has a few bad points associated with it. In spite of that we would like to use this method. It does not require a lot of understanding to get the work done. URL Rewriting basically means that when the user is presented with a link to a particular resource instead of simply presenting the URL as you would normally do, the URL for that resource is modified so that more information is passed when requesting for that resource. We will try to explain URL Rewriting with the same Shopping Cart example used in the hidden field method.

Once again assume that a user has searched for some books and he has been presented with a search result that has two books listed. It is basically a Form with two checkboxes, each for one book and a Submit button to add any of these book to his Cart.

$< b > Search\ results\ for\ books < /b >$
$< form\ method =$ "post" $action =$ "$serverprogram.jsp$"$>$
$< input\ type =$ checkbox$name =$ bookID$value = 100 >$
$Java\ Servlet\ Programming < br >$
$< input\ type =$ checkbox$name =$ bookID$value = 101 >$
$Professional\ JSP < br >$
$< input\ type =$ "$submit$" $name =$ "$Submit$" $value =$ "$Add\ to\ Cart$" $>< br >$
$< /form >$

Now once again suppose the user selects the book named "Java Servlet Programming" and then clicks on the Submit button. This would pass the contents of the form to the server side program called serverprogram.jsp which should read the selected checkboxes and do the necessary (i.e., make some arrangements to keep a track of the selected books, which basically means implement session tracking). Now suppose the user continues browsing and searches for more books and is presented with a new search result just like in the previous example. For better understanding, we shall once again give you the same two results as shown in hidden field method. The two books named "Teach yourself WML Programming" and "Teach yourself C++"

$< b > Search\ results\ for\ books < /b >$
$< form\ method =$ "$post$" $action =$ "$\boldsymbol{serverprogram.jsp?\ BookID} = 100$" $>$
$< input\ type =$ checkbox$name =$ bookID$value = 150 >$
$Teach\ yourself\ WML\ Programming$
$< br >$
$< input\ type =$ checkbox$name =$ bookID$value = 160 >\ Teach\ yourself\ C++$
$< br >$

$< input\ type = \text{submit} name = \text{Submit} value = \text{Add to Cart} >$
$< br >$
$< /form >$

In the above html source, the target for the form has been changed from serverprogram.jsp to serverprogram.jsp? bookID = 100. This is exactly what URL Rewriting means. The original URL which was only serverprogram.jsp has now been rewritten as serverprogram.jsp? bookID = 100. The effect of this is that the any part of the URL after the "?" (Question mark) is treated as extra parameters that are passed to the server side program. They are known as GET parameters. GET method of submitting forms always uses URL Rewriting. Now when the serverprogram.jsp fetches the parameters by the name bookID it would be presented with the one that was present after the "?" in the URL as well as the newly selected checkboxes by the user in that Form.

Consider a general example where a user has selected 2 values, then whenever a program generates a new Form the target for that form should look something like

$$< form\ method = \text{“}post\text{”}\ \ action = \text{“}serversideprogram.jsp?\ \ name1 = value1 + name2 = value2\text{”} > .$$

This sort of URL would keep on increasing as more and more values have to be carried on from one page to another.

The basic concept of URL Rewriting is that the server side program should continuously keep changing all the URLs and keep modifying them and keep increasing their length as more and more data have to be maintained between pages. The user does not see anything on the surface as such but when he clicks on a link he not only asks for that resource but because of the information after the "?" in the URL he is actually sending previous data to the program.

The disadvantage of URL Rewriting (though it is a minor one) is that the displayed URL in the browser is of course the rewritten URL. Thus the clean simple URL that was seen when hidden fields were used, is replaced with a one with a "?" followed by many parameter values. This does not suit those who want the URL to look clean. Another disadvantage is that some browsers specify a limit on the length of a URL. So once the data which is being tracked exceeds beyond a certain limit, you may no longer be able to use URL Rewriting to implement session tracking. But that limit is generally large enough and so do not feel afraid to use this method.

(iii) Cookies

This is one of the most famous methods and the one used by almost all professional sites. This allows us complete flexibility and whatever we want as far as session tracking is concerned. But it is not as easy as the other two methods. Besides some applications may not allow cookies in which case we have to revert back to the other two methods. Websites designed using Wireless Markup Language (WML)

which worked on WAP based cell phones. Unfortunately the cell phones did not have enough memory to support cookies. Hence, we had to use hidden fields to get session tracking working. But cookies would work on almost every computer, except when a user may have blocked all cookies for security reasons in which case we would once again have to use either of the other two methods.

Using cookies is probably the best and the neatest of all the methods to maintain sessions. Cookies are basically small text files that are stored on the user's computers. This has information pertaining to that user. Once the cookie is created on the user's computer then for every further request made by that user in that session, the cookie is sent along with the request. The value of every cookie is unique (for users browsing a particular Website), so the server side program can differentiate between various users.

The method to program cookies is different for different languages. Most of the languages provide some class that covers all the details of cookie creation and maintenance. For example in Java we have "ajavax.servlet.http.Cookie" class that is used to work with cookies.

2.2.1.2. Determination of the session threshold

A unique characteristic of Web workload is the concept of the session. A session is defined as a sequence of requests from the same user during a single visit to the Website.[13–20] Hence, we can define a session as a sequence of requests issued from the same IP address within a certain time period less than some predefined threshold value. For this purpose our first objective should be to identify the user by the IP address. Though we have previously stated the inaccuracies introduced by using the IP addresses as surrogate for users, in spite of that for simplicity we have used the IP address as the reasonable approximation of the number of distinct users.

Our next objective is to find the threshold value that delimits the sessions, i.e., the time limit after which a session will be expired. Web servers close sessions after a predefined period of inactivity to save resources allocated to inactive sessions. If the Website does not enforce a threshold, we need to estimate the threshold from the HTTP access logs of the Web server. Popstoganeva *et al.*[16] have observed that if the threshold value (in minutes) increases, the number of sessions decreases. Once the threshold values larger than 30 min are used there is a little further reduction in the total number of sessions even with the substantial increase in the threshold value. Similarly, we have also conducted the test to find the threshold value. Figure 1 depicts the effects of different threshold values (i.e., $1, 3, 5, 10, 20, 30, 40, 50$ and 60 min) on the total number of sessions for the two data sets described in Table 2. It shows as the threshold value increases, the total number of session's decreases. The arrow in Fig. 1 indicates that there is very little further reduction in the number of sessions if we increase the threshold value beyond 30 min, which supports the *de-facto* standard of taking the session length as 30 min. For example, Google Inc.

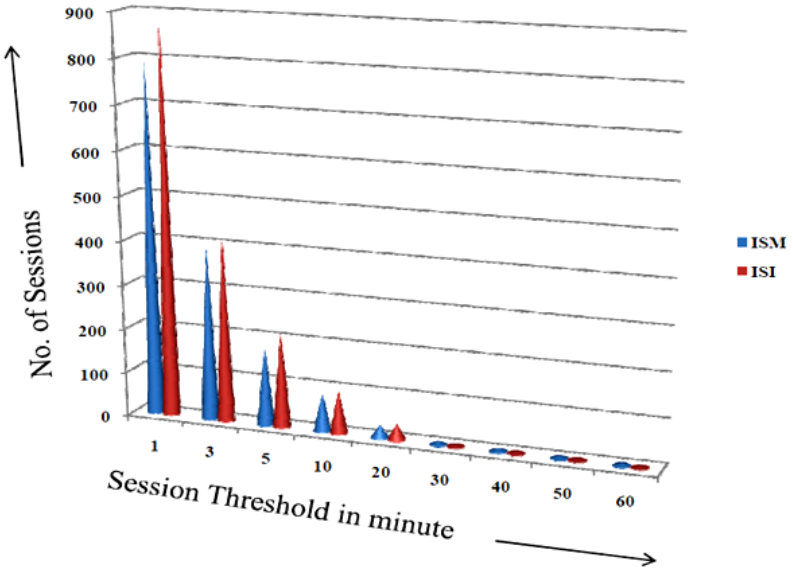


Fig. 1. Effect of session threshold on the number of sessions.

Table 2. Classification of error response code.

Error response codes	Reasons
$1XX$	Informational (100, 101, etc.)
$2XX$	Successfull (200, 201, 203, 204, 205, 206)
$3XX$	Redirectional (300, 301, 302, 303, 304, 305, 306, 307)
$4XX$	Client error (400, 401, 402, 403, 404, 405, etc.)
$5XX$	Server error (500, 501, 502, 503, 504, 505)

uses the 30 min timeout value for their Analytics web application. Hence, in this paper, we fix 30 min time interval as the threshold value for the session.

2.2.2. *Some frequently occurred error codes*

Several types of errors can occur in the system, like system error, HTTP error, etc. These errors are represented by some numbers or digits or some hexadecimal codes called error response codes. In this paper, our area of interest is HTTP error. There is a wide range of HTTP error codes, which are classified in Table 2.

In case of the HTTP error, these error response codes are embedded in the access log and these codes can be mapped to the error entries in the error logs, e.g., in case of HTTP error, a "file not found" error in the log usually corresponds to a 404 error code in the access log. Hence, as described by authors in Refs. 13–20, using access logs is a reasonable method to gather information unless the detailed information about the Web error is required. Among those occurred errors, a few

have a relevant impact on the reliability of the Web software.[19] Following are the descriptions of some frequently occurred error codes for any Website.

200 (Successful): It comes under the class $2XX$. This class of status code indicates that the client's request was successfully received, understood, and accepted. The information returned with the response is dependent on the method used in the request, for example: GET an entity corresponding to the requested resource is sent in the response; HEAD the entity-header fields corresponding to the requested resource are sent in the response without any message-body; POST an entity describing or containing the result of the action; TRACE an entity containing the request message as received by the end server. This will not be considered as the web software failure.

400 (Bad request): It comes under the class $4XX$. The request could not be understood by the server due to its malformed syntax. The client should not repeat the request without modification. This code should not be used in the reliability analysis as in this case, the client is violating the HTTP standard. It is neither a server side error nor the web software error. Since this is a client side issue, it does not make sense to estimate a website's reliability based on this code.

401 (Unauthorized): It comes under the $4XX$ class. The server does not accept the client's authentication credentials, i.e., the request requires user authentication. It occurs when the user requests for a resource, but the user does not have the privilege to access it. If the request already included authorization credentials, then the 401 response indicates that the authorization has been refused for those credentials. This error occurs in the final step above when the client receives a HTTP status code it recognizes as 401. In their paper, Hyunh and Miller[19] have classified it into two categories, viz., source content failures (SCF) and external failures (EFs). Between those two categories, 401 occurred due to the SCF can be considered for the reliability analysis of the Web software.

403 (Forbidden): It comes under $4XX$ class. The server understood the request but refusing to fulfill it. The reason is same as that of the error corresponding to the error response code 401, i.e., authentication failure. If the server does not wish to make this information available to the client then it can issue 404 (not found). Hyunh and Miller[19] have classified this error code into two categories, viz., SCFs and EFs. Between those two categories, 403 occurred due to the SCF can be considered for the reliability analysis of the Web software.

404 (Not found): It comes under $4XX$ class. The server cannot find anything matching the request URI. This error is currently the dominating error code and represented the focus of result given in Refs. 13–20. No indication is given on the condition, i.e., whether it is temporary or permanent. The 410 (Gone) status code should be used if the server knows, through some internally configurable mechanism that, an old resource is permanently unavailable and has no forwarding address. This status code is commonly used when the server does not wish to reveal exactly

why the request has been refused or when no other response is applicable. Hyunh and Miller have classified it into two categories, viz., SCFs and EFs. Between those two categories, 404 occurred due to the SCF can be considered for the reliability analysis of the Web software.

405 (Method not allowed): It comes under class $4XX$. The server is unable to give a response to the method used by the client. For example, suppose the client is using the POST request but the server is unable to give a response to that request. One of the reasons behind it is that, the server is not configured for executing the particular request. Since, this error is due to the configuration, we eliminate this error from reliability analysis of the web software. In this case, the response must include a header containing a list of valid methods for the requested resource. Since this code occurs only due to configuration issue, it can be discarded it for reliability purpose.

406 (Not acceptable): It comes under $4XX$ class. It occurs when the client is incapable to accept the response made by the server. This error can also be neglected as the client is unable to get the data sent by the server. HTTP/1.1 servers are allowed to return responses, which are not acceptable according to the accept headers sent in the request. In some cases, this may even be preferable to sending a 406 response. User agents are encouraged to inspect the headers of an incoming response to determine if it is acceptable. This code can be discarded from the reliability point of view as the server's content does not support the client used to access it.

407 (Proxy authentication required): It comes under the class$4XX$. If the client does not authenticate it to the proxy, this type of error will occur. The nature of this error is similar to 401 (unauthorized). This code should be discarded from the reliability point of view as the client does not authenticate with the server before attempting to access restricted content.

408 (Request timeout): It comes under the class $4XX$. The client is unable to send a request within the time that the server is prepared to wait. We can say that it is a network failure. Hence, this will not be considered for reliability analysis.

409 (Conflict): It comes under class $4XX$. The Web server (running the Website) is thought that, the request submitted by the client (e.g., your Web browser or our CheckUpDown robot) cannot be completed because it conflicts with some rule hitherto established. For example, we may get a 409 error if we try to upload a file to the web server which is older than the one already there, resulting in a version control conflict. This error is also discarded from the reliability point of view as it happens due to the version of the browser.

410 (Gone): It comes under $4XX$ class. This will occur when the resource requested by the client is removed from the server. For example, suppose the client is requesting for a particular file which has been removed; the result is a 410 error.

The nature of this error is same as 404 (not found). So in case of reliability analysis we will do the same as that of response code 404.

411 (Length required): It comes under $4XX$ class. The server is refusing the request sent by the client as the size of the data is not specified. This error occurs due to the size of the data is not specified. Since, it is a browser error it will not be considered for the web software reliability analysis.

412 (Precondition failed): It comes under $4XX$ class. The requested resource failed to satisfy the defined precondition, the result is a 412 error. This response code allows the client to place preconditions on the current resource meta-information (header field data) and thus, prevent the requested method from being applied to a resource other than the one intended. It is considered in analyzing the reliability of the web software.

413 (Request entity too large): It comes under $4XX$ class. The server is refusing to process a request because; the request entity is larger than the server is willing or able to process. Hence, this error occurs because of the configuration of the server and will not be taken into consideration for reliability analysis of the Web software.

414 (The request-URI too long): It comes under $4XX$ class. The server is refusing to service as the requested URI is too long to interpret. Typically, Web servers set fairly generous limits on length for genuine URLs, e.g., up to 2048 or 4096 characters. If the URL is particularly long, we can usually try shorter variations to see roughly where the limit is. The general cause of this phenomenon is that, the client is trying to incorporate some vulnerability in the web server. It contains a large amount of malicious code, which will overflow the buffer. If the long URL is indeed valid, then the Web server may need to be reconfigured to allow your URLs through. If we find this code very frequently in the access logs, we have to understand that the server is under attack. In reliability analysis we will not consider it, as the client is trying to manipulate the server. Understand that Web servers have to set some reasonable limit here, because they have to deal with badly programmed clients trying to give them huge garbage URLs. Since this error is also due to the configuration of the server, it will not be taken into consideration for reliability analysis of the Web software.

415 (Unsupported media type): It comes under $4XX$ class. The server does not agree with the media type specified in the request. For example, the browser or the client request a text file but the server is unable to respond as the server recognizes the file as a binary one. It is also possible that, the actual media type is incompatible with the HTTP method specified in the request. Since, it is a browser failure; we will not consider it for reliability analysis.

416 (Requested range not satisfiable): It comes under $4XX$ class. Here the client is requesting for a file whose size is not valid. We will omit this error code for reliability analysis.

500 (Internal error): It comes under $5XX$ class. The server encountered an unexpected condition which prevented it from fulfilling the request. Therefore, it must be considered for reliability analysis.

501 (Not implemented): It comes under $5XX$ class. In this case the server does not understand the request of the client. It must be included in the reliability analysis of the Web software.

502 (Bad gateway): It comes under $5XX$ class. The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request. This error should be included in reliability analysis.

503 (Service unavailable): It comes under $5XX$ class. The server is overloaded and unable to process the further request. If a very large number of requests come to the server, it is unable to give a response or the response time becomes too long. For example, at the time of publication of the GATE result the number of requests to the respective Indian Institute of Technology (IIT) — Website is so large that we have to wait for a very long time to see the result. The existence of the 503 status code does not imply that a server must use it when becoming overloaded. Some servers may wish to simply refuse the connection. This failure is due to the server. Therefore, in the reliability analysis of the web software we will consider this error code.

504 (Gateway timeout): It comes under $5XX$ class. In this case the server is acting as a gateway or a proxy server. This problem is entirely due to slow IP communication between back-end computers, possibly including the Web server. Hence, this error will not be considered for reliability analysis of the web software.

505 (HTTP version not supported): It comes under $5XX$ class. The server does not support the HTTP version used by the client. Suppose the client is using the HTTP1.1, but the server is running some lower version of HTTP. Therefore, the server is unable to give the response. This error will also be rejected from Web software reliability analysis.

2.3. *Proposed method*

In this subsection, we have developed a transfer function based model to forecast the combined occurrences of all the errors having an impact on the reliability of the Web software. Detail study about transfer function modeling is available in Refs. 2, 8–10, 21–26. Transfer function models represent a dynamic relationship between a continuous input and a continuous output. The relationship between the continuous input X_t and the continuous output Y_t, i.e., the transfer between X_t and Y_t is represented by a linear differential equation. In transfer function model building, observations must be considered in pairs (X_t, Y_t), each measured at equispaced times. In the discrete transfer function model X_t, Y_t both are discrete and the

transfer between them is represented parsimoniously[22–26] by the linear difference equation

$$(1 + \xi_1\nabla + \cdots + \xi_r\nabla^r)Y_t = (\eta_0 + \eta_1\nabla + \cdots + \eta_s\nabla^s)X_{t-b}. \quad (1)$$

In Eq. (1), the backward difference operator ∇ is used in place of the differential operator $D = \frac{d}{dt}$ since X_t and Y_t are discrete. Here, $\xi(\nabla) = 1 + \xi_1\nabla + \cdots + \xi_r\nabla^r$, $\eta(\nabla) = \eta_0 + \eta_1\nabla + \cdots + \eta_s\nabla^s$ are different operators, $\xi_1, \xi_2, \ldots, \xi_r$ and $\eta_0, \eta_1, \ldots, \eta_s$ are unknown parameters, which in practice, have to be estimated from the data. Constant b the delay parameter, associated with the leading indicator series X_t indicates which of the previous values X_t affect the present Y_t. The parameter b is called the time delay (or dead time) of the system.[2,8–10,21–26] For example, if $b = 1$, then $v_0 = 0$ and X_t has no impact on Y_t, but X_t will affect Y_{t+1}. In other words, the impact of X_t on the output series $\{Y_t\}$ is delayed for one time period. In general if b unit delay is assumed, then the index t is replaced by $(t-b)$. Equation (1) may be written equivalently in terms of past values of the input and output by substituting $B = 1 - \nabla$, where B is the backward shift operator defined as $BX_t = X_{t-1}$ and $B^bX_t = X_{t-b}$. Therefore, Eq. (1) becomes:

$$\begin{aligned}(1 - \delta_1 B - \cdots - \delta_r B^r)Y_t &= (1 - \omega_1 B - \cdots \omega_s B^s)X_{t-b} \\ &= (\omega_0 B^b - \omega_1 B^{b+1} - \cdots \omega_s B^{b+s})X_t\end{aligned}$$

or,

$$\delta(B)Y_t = \omega(B)B^bX_t = \Omega(B)X_t,$$

where, $\delta(B) = 1 - \delta_1 B - \cdots - \delta_r B^r, \omega(B) = 1 - \omega_1 B - \cdots \omega_s B^s$ and $\Omega(B) = \omega(B)B^b$ are different operators used in time series analysis.[22–26] Alternatively, the pair of observations (X_t, Y_t) is represented by a linear filter:

$$Y_t = v_0X_t + v_1X_{t-1} + v_2X_{t-2} + v_3X_{t-3} + \cdots = v(B)X_t, \quad (2)$$

where $v_0, v_{1\ldots}$ are constant denoting the impact of X_{t-j} on Y_t and $v(B) = v_0 + v_1B + v_2B^2 + \cdots$. The coefficients v_0, $v_{1\ldots}$ are referred to as the impulse response function[2,8–10,21–26] of the system. For the model in Eq. (1) to be meaningful, the response must satisfy certain conditions. A simple condition is that $\sum_{j=0}^{\infty} |v_j| < \infty$, i.e., the impulse responses are absolutely summable.[22–26] In this case, the system is said to be stable. The function $v(B)$ determines the impact of input X_t to output Y_t and it is said to be the transfer function. For the linear filter model in Eq. (2), the transfer function can be expressed as:

$$Y_t = \frac{\omega(B)B^b}{\delta(B)}X_t, \quad \text{i.e., } v(B) = \frac{\omega(B)}{\delta(B)}. \quad (3)$$

It is the final transfer function model of order (r, s) in case of single input and single output. When there are multiple input variables, the transfer function model

becomes

$$Y_t = \left(\frac{\omega_1(B)B^{b_1}}{\delta_1(B)} X_{1t} + \frac{\omega_2(B)B^{b_2}}{\delta_2(B)} X_{2t} + \cdots \right) = \sum_i \frac{\omega_i(B)B^{b_i}}{\delta_i(B)} X_{it}, \tag{4}$$

where, $\omega_i(B)(i = 1, 2, 3, 4)$ and $\delta_i(B)(i = 1, 2, 3, 4)$ are similarly defined in Eq. (3). The parameters $b_i(i = 1, 2, 3, 4)$ are called the time delay (or dead time)[2,8–10,21–26] of the input series $X_{it}(i = 1, 2, 3, 4)$. This delay is due to the heavy traffic, congestion in the network and destruction of the packets. Equation (4) is the final transfer function model for multiple inputs and a single output.

In practice, the output Y_t is not a deterministic function of X_t (in case of single input and single output). It is often disturbed by some noise or has its own dynamic structure. We denote the noise component as N_t. The noise may be serially correlated, and we assume that N_t follows an ARMA(p, q) model, i.e.,

$$\phi(B)N_t = \theta(B)a_t, \tag{5}$$

where, $\theta(B) = 1 - \theta_1 B - \cdots - \theta_q B^q$ and $\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p$ are polynomials in B of degree q and p, respectively, and $\{a_t\}$ is a sequence of independent and identically distributed random variables with mean zero and variance σ_a^2. Often we also assume that a_t is Gaussian. The non-negative integer q, i.e., the order of the MA process indicates the number of lags beyond which the theoretical autocorrelation function (ACF) is effectively 0. The non-negative integer p, i.e., the order of the AR process indicates the number of lags beyond which the theoretical partial ACF is zero.[2,8–10,21–26] For the ARMA(p, q) model given in Eq. (5), $E(N_t) = 0$ and the usual conditions of stationarity and invertibility are also applied. Putting together, we obtain a simple transfer function model as (for single input and single output):

$$Y_t = c + v(B)X_t + N_t = \frac{\omega(B)B^b}{\delta(B)} X_t + \frac{\theta(B)}{\phi(B)} a_t, \tag{6}$$

where, c is a constant, $\theta(B)$, $\phi(B)$, $\omega(B)$ and $\delta(B)$ are defined as before with degree q, p, s and r, respectively, and $\{a_t\}$ are Gaussian white noise series. The noise component N_t should be independent of X_t; otherwise, the model is not identifiable. When there are multiple input variables $X_{it}(i = 1, 2, 3, 4)$ and single output variable Y_t the transfer function model becomes:

$$\begin{aligned} Y_t &= c + \left(\frac{\omega_1(B)B^{b_1}}{\delta_1(B)} X_{1t} + \frac{\omega_2(B)B^{b_2}}{\delta_2(B)} X_{2t} + \cdots \right) + \frac{\theta(B)}{\phi(B)} a_t \\ &= c + \sum_i \frac{\omega_i(B)B^{b_i}}{\delta_i(B)} X_{it} + \frac{\theta(B)}{\phi(B)} a_t, \end{aligned} \tag{7}$$

where, $\omega_i(B)$, $\delta_i(B)$ and b_i are similarly defined as in Eq. (4).

2.3.1. *Pre-whitening technique*

Consider the transfer function model in Eq. (6). Since, X_t and N_t might be serially correlated, the regression

$$Y_t = c + v_0 X_t + v_0 X_{t-1} + \cdots + v_0 X_{t-h} + e_t,$$

where, h is a large positive integer, would, in general, not provide consistent estimates of the $v_i's$.[22–26] The pre-whitening technique has been proposed as a tool to obtain consistent estimates of $v_i's$ whose central idea is to remove the serial dependence in X_t. Suppose that X_t follows the univariate ARMA model given as $\phi(B)X_t = \theta(B)\eta_t$, where, $\{\eta_t\}$ is the sequence of white noises (i.e., iid random variables). Applying the operator $\frac{\phi(B)}{\theta(B)}$ in Eq. (6), we get:

$$\frac{\phi(B)}{\theta(B)}Y_t = c^* + v(B)\frac{\phi(B)}{\theta(B)}X_t + \frac{\phi(B)}{\theta(B)}N_t = c^* + v(B)\eta_t + \frac{\phi(B)}{\theta(B)}N_t, \tag{8}$$

where, c^* is a constant given by $c^* = \frac{\phi(1)}{\theta(1)}c$. Define, $y_t = \frac{\phi(B)}{\theta(B)}Y_t$ and $n_t = \frac{\phi(B)}{\theta(B)}N_t$. Equation (8) reduces to:

$$y_t = c^* + v(B)\eta_t + n_t. \tag{9}$$

Notice that $\{n_t\}$ is independent of $\{\eta_t\}$ and η_t is a series of white noise.[22–26] Multiplying Eq. (9) by $\eta_{t-j}, j \geq 0$, we have

$$y_t\eta_{t-j} = c^*\eta_{t-j} + [v(B)\eta_t]\eta_{t-j} + n_t\eta_{t-j}. \tag{10}$$

Here, $\{n_t\}$ is independent of $\{\eta_t\}$ and $\{\eta_t\}$ is a white noise series. Taking expectation of Eq. (10), we obtain $\mathrm{Cov}(y_t, \eta_{t-j}) = v_j\mathrm{Var}(\eta_{t-j})$. Consequently, we have $v_j = \frac{\mathrm{Cov}(y_t,\eta_{t-j})}{\mathrm{Var}(\eta_{t-j})}$. In practice, the model for X_t can be specified via the univariate time series analysis.[22–26] One can then apply the above model to obtain y_t. This process is called pre-whitening or filtering.

In case of multiple inputs $\{X_{it}\}(i = 1, 2, 3, 4)$ and single output $\{Y_t\}$, the pre-whitening can be described as follows:

Let, $\{X_{it}\}$ $(i = 1, 2, 3, 4)$ follows the model given as $\phi_i(B)X_{it} = \theta_i(B)\eta_t, (i = 1, 2, 3, 4)$ where, $\{\eta_t\}$ are the sequences of white noises. Then applying the operator $\frac{\phi_i(B)}{\theta_i(B)}$ (i is fixed) in Eq. (7), we get:

$$\begin{aligned}\frac{\phi_i(B)}{\theta_i(B)}Y_t &= c^* + \frac{\phi_i(B)}{\theta_i(B)}\sum_j v_j(B)X_{jt} + \frac{\phi_i(B)}{\theta_i(B)}N_t \\ &= c^* + \frac{\phi_i(B)}{\theta_i(B)}\sum_j v_j(B)X_{jt} + n_t,\end{aligned} \tag{11}$$

where, c^* is a constant given by $c^* = \frac{\phi_i(B)}{\theta_i(B)}c$. Define, $y_t = \frac{\phi_i(B)}{\theta_i(B)}Y_t$ and $n_t = \frac{\phi_i(B)}{\theta_i(B)}N_t$. The function $v_j(B)$ determines the impact of input $X_{jt}(j = 1, 2, 3, 4)$ to output Y_t and it is said to be the transfer function. Here, $\{n_t\}$ is independent of the white noise series $\{\eta_t\}$. For example, there are two independent input $\{X_{it}\}(i = 1, 2)$ and

single output $\{Y_t\}$ variables. Then, $\phi_1(B)X_{1t} = \theta_1(B)\eta_t$ and $\phi_2(B)X_{2t} = \theta_2(B)\eta_t$. Equating them, we get $\frac{\phi_1(B)}{\theta_1(B)}X_{1t} = \frac{\phi_2(B)}{\theta_2(B)}X_{2t}$. Hence, $X_{2t} = \frac{\phi_1(B)}{\theta_1(B)}\frac{\theta_2(B)}{\phi_2(B)}X_{1t}$. Operating, Eq. (11) by $\frac{\phi_1(B)}{\theta_1(B)}$, we get:

$$\begin{aligned}\frac{\phi_1(B)}{\theta_1(B)}Y_t &= c^* + \frac{\phi_1(B)}{\theta_1(B)}\sum_j v_j(B)X_{jt} + \frac{\phi_1(B)}{\theta_1(B)}N_t \\ &= c^* + \frac{\phi_1(B)}{\theta_1(B)}v_1(B)X_{1t} + \frac{\phi_1(B)}{\theta_1(B)}v_2(B)X_{2t} + \frac{\phi_1(B)}{\theta_1(B)}\frac{\phi_1(B)}{\theta_1(B)}N_t \\ &= c^* + v_1(B)\eta_t + v_2(B)\frac{\phi_1(B)}{\theta_1(B)}\frac{\phi_1(B)}{\theta_1(B)}\frac{\theta_2(B)}{\phi_2(B)}X_{1t} + \frac{\phi_1(B)}{\theta_1(B)}n_t \\ &= c^* + v_1(B)\eta_t + v_2(B)\frac{\phi_1(B)}{\theta_1(B)}\frac{\theta_2(B)}{\phi_2(B)}\eta_t + \frac{\phi_1(B)}{\theta_1(B)}n_t\end{aligned}$$

or,

$$y_t = c^* + v_1(B)\eta_t + v_2(B)\frac{\phi_1(B)}{\theta_1(B)}\frac{\theta_2(B)}{\phi_2(B)}\eta_t + \frac{\phi_1(B)}{\theta_1(B)}n_t.$$

Multiplying Eq. (11) by $\eta_{t-k}k \geq 0$, we have

$$y_t\eta_{t-k} = c^*\eta_{t-k} + \frac{\phi_i(B)}{\theta_i(B)}\sum_j v_j(B)X_{jt}\eta_{t-k} + n_t\eta_{t-k}. \tag{12}$$

Taking expectation of Eq. (12) and using the techniques described in Refs. 22–26 for obtaining the consistent estimates of the lth element of the expansion of $v_j(B)$, i.e., $v_{lj}(j = 1, 2, 3, 4)$, we have derived the parameter values. The computation has been done by using the SPSS 20 and R software. One can then apply the model to obtain y_t. This process is called pre-whitening or filtering.

2.3.2. *Stepwise procedure*

In this subsection, we have described the step wise procedure for developing a transfer function model with pre-whitening technique to predict of the occurrences of Web errors having an impact on the reliability of the Web software.

Step I.

(1) Extract all the Web workload characteristics ($X_{it}(i = 1, 2, 3, 4)$), i.e., number of hits, number of bytes transferred, number of users and the number of sessions created from the HTTP log files (access and error logs) of the corresponding Web software, given in Sec. 2. In order to determine the existing correlation among different Web workload characteristics PCA has been performed. Assume that, there is $m(\leq 4)$ PCs, i.e., uncorrelated Web workload characteristics in the Web system.

(2) Extract the occurrences of all the error codes (Y_t), having an impact on the reliability of the Website from the HTTP logs (access and error logs) of the corresponding Web software, based on the analysis given in Sec. 2.2. Add their each day's occurrences to get the combined occurrences.

Step II.

(1) Prior to build the model, at first careful screening of data is needed. This is done by normalizing the data with suitable transformation like log transfer to remove the non-stationarity in the data.
(2) After normalizing the data, find noise component (N_t) present in the data using the method given in Sec. 2.3.1.1. Develop a transfer function model for single (multiple) input (s) and single output based on the number of PCs(≤ 4). This can be accomplished by an examination of the partial autocorrelation, autocorrelation and cross-correlation.[2,8–10,21–27] X_t and N_t might be serially correlated (in case of univariate time series). Hence, apply pre-whitening technique to get the consistent estimate of v_j. In case of multiple input series (X_{it}) and single output (Y_t), if $X_{it}(i = 1, 2, 3, 4)$ and N_t are serially dependent, apply pre-whitening technique to get the consistent estimate of v_{ij}.
(3) Use the maximum likelihood estimation technique to estimate the model parameters by minimizing the conditional sum of square function as given in Refs. 22–27.
(4) Equations (3), (4), (6) and (7) are then used to predict the remaining faults present in the Web software.

3. Overview of the Websites Used in this Study

We have validated our proposed method using real failure data obtained from two different Websites. One is www.ismdhanbad.ac.in, the official Website of Indian School of Mines Dhanbad, India and the other is www.isical.ac.in, the official Website of Indian Statistical Institute (ISI), Kolkata, India. In this section, brief descriptions about these two Websites have been given. Also, we have analyzed different frequently occurred error codes having an impact on the reliability of the two above mentioned Websites. www.ismdhanbad.ac.in a non-commercial, dynamic Website, utilizes the PHP (http://www.php.net) scripting language, MySql (http://www.mysql.com) for the backend database and is hosted on an Apache HTTP Daemon. To investigate the stability and reliability of the data, the log files (HTTP access and the error logs) were chosen to cover 25 consecutive days starting from 30th September 2010 to 24th October 2010. During this period, the Website has received approximately 636,793 hits, 18,839 unique visitors, 2,533 unique user agents (Mozilla/5.0+(compatible;+Googlebot/2.1;++http://www.google.com/bot.html)) and transferred a total amount of 8,764,646 KB data. www.isical.ac.in, is also a non-commercial, dynamic Website utilizes the PHP scripting language and an Apache HTTP daemon. To analyze the stability and reliability of this Website,

Table 3. Brief overview and comparative studies of ISM and ISI Websites.

Date set	Log duration (days)	Starting date	Sessions	Avg. sessions/day	Data transferred
ISM	25	30/10/10	3,137	125.48	8,764,646 KB
ISI	34	16/09/12	5,304	156	10.1 GB

Table 4. Total occurrences of different error codes present in ISI and ISM Web servers' log files.

Data	404	406	401	200	304	206	207	301	302	403	412	416	500	501
ISI	333,009	22	1	1698,441	160,391	160,981	0	28,371	40,414	5,252	13	25	589	4
ISM	75,717	1,712	41	490,713	39,251	26,697	2,382	49	5	149	4	51	19	3

34 consecutive day's (starting from 16th September to 19th October 2012) log files have been collected. During this period of time, this Website has transferred a total amount of 10.1 GB of data, received approximately 841,791 hits. Tables 3 and 4 describe a brief overview about the number of sessions created, total amount of data transferred, different frequently occurred errors of www.ismdhanbad.ac.in and www.isical.ac.in.

In both the cases, we found that the error code 404 numerically dominates the others as noted by Tian *et al.*[15] According to the survey results from 1994 to 1998 by the Graphics, Visualization, and Usability Center of Georgia Institute of Technology (http://www.gvu.gatech.edu/user_surveys/), 404 errors are the most common errors that users encounter while browsing the Web. Ma and Tian[17] found that a majority of these 404 errors is caused by internal bad links while only a small percentage are caused by external factors such as the user mistyping the URL, robots from search engines, external links (links from other Websites), old bookmarks, etc. No analysis exists as to the "value" (of the information) encoded within the various error types for Website administrators. Therefore, we will examine all of the error codes encountered to determine which errors are truly SCFs (have value) and which are attributed to other uncontrollable factors (no value). No analysis exists as to the "value" (of the information) encoded within the various error types for Website administrators. Therefore, we will examine all of the error codes encountered to determine which errors are truly SCFs (have value) and which are attributed to other uncontrollable factors (no value). The errors of type SCF can be considered in the reliability analysis as they can be handled or rectified by the site administrator. For example, we found that the error codes (in case of www.ismdhanbad.ac.in and www.isical.ac.in) 401, 403, 404, 500 and 501 that have either SCF or host failure or EF as a potential failure source; hence, these error codes will be examined in detail in order to determine their exact failure sources. Further, the 401, 403 and 404 error codes have both SCF and EFs as failure modes or sources. We will show that a little amount (0.0004%) of 404 response errors have value for www.ismdhanbad.ac.in, as they are generated due to SCF and the site administrator is expected to respond and correct the 404 errors immediately due to the potential loss in sales and customers that this error code can cause, whereas the 401, 403 error response code have no

value for www.ismdhanbad.ac.in because all of the 404 recorded errors are caused by factors beyond the reach of the site administrator. All the 500 and 501 (server related errors) error response codes are generated due to the SCF in the case of the above mentioned Website. Again in case of www.isical.ac.in, we can find a very less amounts of the 404 (0.014%) and 403 (0.05%) error response codes are of SCF type. All the 500 and 501 error response codes are generated due to the SCF in the case of www.isical.ac.in, which is very much within the reach of the site administrator. Hence in case of www.ismdhanbad.ac.in, we consider 404 (due to SCF), 500 and 501 are the errors having an impact on its reliability. Again in case of www.isical.ac.in, 404, 403 (type SCF), 500 and 501 are the error codes having an impact on its reliability. Table 5 demonstrates the above statistics.

One common argument is that if information is available, EFs can also be resolved. This logic is not valid for several reasons. A site administrator can only be reactive to EFs rather than being proactive, i.e., until an EF occurs, a site administrator will not have enough information to resolve that failure. Furthermore, depending on circumstances, the failure may not always be resolved. For example, an external Website has a link to a Web page on the Web system under

Table 5. Possible error codes that have an impact on the reliability of www.ismdhanbad.ac.in and www.isical.ac.in.

Error codes	Probable reasons	% of occurrence		SCF(%)		EF(%)	
		ISI	ISM	Impact on reliability = YES		Impact on reliability = NO	
				ISI	ISM	ISI	ISM
401	Source Content Failure External Failure	6,274 (1%)	6,354 (4%)	3,513 (67%)	4,956 (78%)	2,760 (33%)	1,398 (22%)
403	Source Content Failure External Failure	28,822 (3%)	25,419 (2%)	16,140 (56%)	14,997 (58%)	12,681 (44%)	10,422 (42%)
404	Source Content Failure External Failure	796,046 (95%)	635,491 (96%)	51,742 (65%)	470,263 (74%)	278,616 (35%)	165,227 (26%)
500	Source Content Failure External Failure	627 (0.01%)	127 (0.02%)	545 (87%)	108 (85%)	81 (13%)	19 (15%)
501	Source Content Failure External Failure	1,882 (0.03%)	127 (0.02%)	1,355 (72%)	96 (76%)	527 (28%)	30 (14%)
502	Source Content Failure External Failure	56 (0.009%)	57 (0.009%)	44 (79%)	45 (79%)	12 (21%)	12 (21%)
503	Source Content Failure External Failure	4,391 (0.07%)	381 (0.06%)	3,908 (89%)	339 (89%)	483 (11%)	24 (11%)

examination. However, due to recent changes, that Web page is no longer valid. The site administrator will not be aware of this issue until a user follows the link from the external Website. Once the failure occurs, the site administrator can attempt to resolve it by attempting to contact the external Website's Webmaster to get the link updated. However, this process requires cooperation from the external Website's Webmaster. Furthermore, the process becomes tedious when there are thousands of Websites linking to this invalid Web page. The site administrator can also attempt to redirect the user to the correct page. However, this requires the site administrator to have a complete mapping of all invalid pages to valid pages which is clearly infeasible. Because of these potential issues, the site administrator cannot resolve EFs adequately.

Previously it was noted that the error response codes can be associated to one or more failure sources, which are classified as SCF and the EFs. Hence, in the present case study a survey was performed and found that a large portion of the error codes $401, 403, 404, 500, 501, 502, 503$ occurred due to the SCF and as a consequence, only the aforementioned error codes have the influences on the reliability of both the Websites (www.ismdhanbad.ac.in, www.isical.ac.in). From Table 6, it is found that error 404 is the most dominant error in case of both the above mentioned Websites. It is also found that, 26% of the total occurrences of 404 in case of www.ismdhanbad.ac.in and 35% of the total occurrences of 404 in case of www.isical.ac.in are occurred due to the EF, which are beyond the scope of the respective Web administrator and have no influence on the reliability of the respective Websites. The next most dominant Web errors in case of www.isical.ac.in is 401 and that in the case of www.ismdhanbad.ac.in is 403.

After rigorous analysis of the HTTP log files (access and the error logs) of both the Websites under the present study, it is discovered that, for these Websites, the SCF can further be classified into following two types:

SCF_ADMIN: These are errors on the Website that should be recognized and rectified by the Web administrators or content providers. These Web errors can be identified by careful inspection of the "referer field" of the HTTP access logs of the respective Websites (mentioned in Sec. 4) as follows:

If the "referer field" of an error entry in the HTTP access log contains the Website's URL, then the corresponding error can be classified as the SCF_ADMIN category. A procedure to extract the data present in the "referer fields" of the HTTP access log. These errors have an impact on the reliability of the Website.

SCF_OLDER: These are usually links of external Websites pointing to an older version of the Websites under study. This old version still exists on the HTTP Daemon for archival purposes and has no connections to the current Websites. Hence, it is not maintained and can mainly contain several bad, broken, and disabled hyperlinks. For better and clearer understanding, the example of "Wikipedia" can be cited, which consists of the hyperlinks of a number of different external Websites.

When a client visits the old version hyperlink present in the external Website — through search engines (Google, Yahoo, etc.), old bookmarks, old emails, etc. — and clicks on one of these bad or disabled hyperlinks, the log data will record that the error is caused by an internal source. Notwithstanding this, these errors are very much under the direct control of system administrators and as a consequence it can be considered as the SCF_OLDER type. It can be identified using the following method:

If the referer URL corresponding to the HTTP access logs of the Website leads to an old version of the Website, then the error can be classified to SCF_OLDER type. These errors have an impact on the reliability of the Website too. Table 6 shows the number of SCF_ADMIN and SCF_OLDER in case of both the Websites. The main motive behind this distinction is to prevent people from being misguided by the errors of SCF_OLDER category as, it is quite similar to the EF though, it is very much within the scope of the respective Web administrator. As a consequence, the errors belonging to SCF_OLDER category are having influences on the reliability of the Website.

Likewise, the EF can also be distinguished into two categories, viz., EF_OLD and EF_EXTERNAL which are defined as follows:

EF_OLD: It generally signifies the old bookmarks, bad hyperlinks of the other Websites. People often find it very much similar to SCF_OLDER. A closer inspection of the HTTP access logs unveils the errors of this category. This type of error has no influence on the reliability of the Website.

Table 6. Number of SCF_ADMIN and SCF_OLDER in case of www.ismdhanbad.ac.in and www.isical.ac.in.

Error codes	SCF_ADMIN		SCF_OLDER	
	Impact on reliability = YES		Impact on reliability = YES	
	ISI	ISM	ISI	ISM
401	2,353 (67%)	4,212 (85%)	1,159 (33%)	743 (15%)
403	12,750 (79%)	11,697 (78%)	3,389 (21%)	3,299 (22%)
404	393,246 (76%)	395,020 (84%)	124,183 (24%)	122,268 (26%)
500	430 (79%)	97 (90%)	114 (21%)	11 (10%)
501	1,246 (91%)	90 (94%)	121 (09%)	4 (6%)
502	42 (97%)	31 (70%)	2 (03%)	14 (30%)
503	3,321 (85%)	294 (87%)	586 (15%)	44 (13%)

Table 7. Number of EF_OLD and EF_EXTERNAL in case of www.ismdhanbad.ac.in and www.isical.ac.in.

Error codes	EF_OLD		EF_EXTERNAL	
	Impact on reliability = NO		Impact on reliability = NO	
	ISI	ISM	ISI	ISM
401	1,352 (49%)	1,385 (99%)	1,407 (51%)	13 (1%)
403	12,046 (95%)	10,317 (99%)	634 (5%)	104 (1%)
404	108,660 (39%)	270,257 (97%)	169,955 (61%)	8,358 (3%)
500	77 (95%)	72 (90%)	6 (5%)	9 (10%)
501	516 (98%)	511 (97%)	9 (02%)	16 (3%)
502	44 (99%)	12 (100%)	1 (01%)	0 (0%)
503	335 (99%)	483 (100%)	4 (01%)	0 (0%)

EF_EXTERNAL: It is due to the scanners of the external attackers which are mainly out of reach of the respective Web administrator. If the requested resources belong to Web applications not installed for the Website, then the errors can be classified as EF_EXTERNAL. Table 7 shows the number of EF_OLD and EF_EXTERNAL in case of both the Websites. The errors belong to this category have no influence on the reliability of the Website.

4. Results and Discussions

Extract all the Web workload characteristics (number of hits, bytes transferred, number of users and number of sessions created) and the errors having an impact on the reliability of the Websites under study (www.ismdhanbad.ac.in and www.isical.ac.in) from the HTTP logs collected from the respective Web servers as described in Sec. 2. Previously it has been mentioned that, we have 25 consecutive days log files of www.ismdhanbad.ac.in and 34 consecutive day log files of www.isical.ac.in which are not sufficient for time series model fitting. To overcome this problem, the daily data was analyzed in two parts, one from 12:00 am to 11:59 am and other from 12:00 pm to 11:59 pm. Thus the 25 days continuous data of www.ismdhanbad.ac.in were converted into 50 numbers of observations with an interval of 12 h and 34 consecutive day occurrences of www.isical.ac.in were converted to 68 numbers of observations of 12 h interval. From simple plots of the Web workload measures and errors over 12 h interval, we can find a high variability in the usage pattern of both the Websites. Hence, to normalize the data, log transformation has been applied. Different Web workload characteristics of www.ismdhanbad.ac.in have been given

in Table 7. In case of www.ismdhanbad.ac.in, first 34 observations of Y_t and X_t with an interval of 12 h have been taken for the model fitting purpose and the remaining 16 observations of Y_t with an interval of 12 h for prediction purpose. Different Web workload characteristics of www.isical.ac.in have been tabulated in Table 10.

Similarly in this case, first 44 observations of $X_{1t} = \log(\text{Hits})$, $X_{2t} = \log(\text{BytesTransferred})$, $X_{3t} = \log(\text{Users})$ and Y_t with an interval of 12 h have been taken for the model fitting purpose and the remaining 24 observations of Y_t with an interval of 12 h for the prediction purpose. The input $\{X_{it}\}(i = 1, 2, 3, 4)$ selection procedure has been described in Sec. 2.3.2.

In order to determine if there exists any correlation among the Web workload characteristics of www.ismdhanbad.ac.in or not, PCA proposed in Step II of Sec. 2.3.2, has been performed with the data given in the first four columns of Table 8. Table 9 shows the correlation matrix of www.ismdhanbad.ac.in and Fig. 2(a) shows the corresponding Scree plot. The x-axis represents the Principal Components (PCs) sorted by decreasing fraction of total variance explained. (The numerical labels assigned to each PC are according to this ordering, and persist whether or not the Scree Plot is actually displayed.) The y-axis contains the fraction of total variance explained. The plot shows that only one component has the proportion of variance as 1 (the red line shows the fluctuation of variance) and all other components after PC1 appear to level off, which suggests that only one component is of importance. However, Website administrators should select the Web workload characteristic most suitable for their requirements. In this case, we have taken the log(12 h Bytes transferred-count) (i.e., X_t) as the input and predict log(cumulative occurrences of errors) (i.e., Y_t) having an impact on the reliability of www.ismdhanbad.ac.in. From Eq. (3), we have developed a single input and single output transfer function model between the output (Y_t) and the input variable (X_t) as $\phi(B)Y_t = \theta(B)X_t$. After analyzing the partial ACF and ACF, we have decided

$$\begin{aligned}\phi(B) = 1 &- 8.882 \times 10^{-16}B - 6.661 \times 10^{-16}B^2 + 2.47 \\ &\times 10^{-15}B^3 - 3.775 \times 10^{-15}B^4 + 5.551 \times 10^{-16}B^5\end{aligned}$$

and

$$\begin{aligned}\theta(B) = 1 &+ 6.661 \times 10^{-16}B + 6.384 \times 10^{-16}B^2 \\ &- 2.442 \times 10^{-15}B^3 + 4.163 \times 10^{-16}B^4 - 4.441 \times 10^{-16}B^5.\end{aligned}$$

The transfer function model between the output (Y_t) and the input (X_t) is given in Eq. (13)

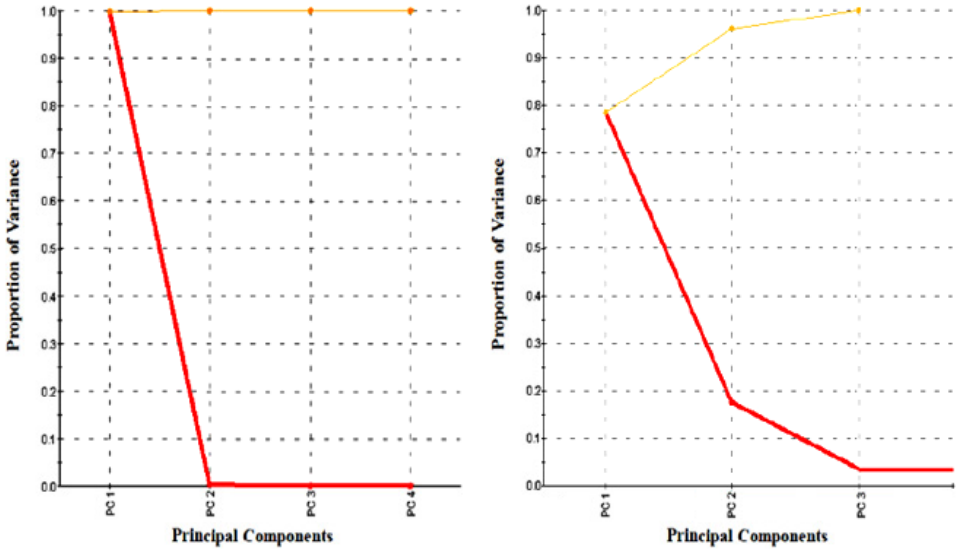
$$Y_t = \frac{\theta(B)}{\phi(B)}X_t = \frac{\begin{array}{c}1 + 6.661 \times 10^{-16}B + 6.384 \times 10^{-16}B^2 - 2.442 \times 10^{-15}B^3 \\ + 4.163 \times 10^{-16}B^4 - 4.441 \times 10^{-16}B^5\end{array}}{\begin{array}{c}1 - 8.882 \times 10^{-16}B - 6.661 \times 10^{-16}B^2 + 2.47 \times 10^{-15}B^3 \\ - 3.775 \times 10^{-15}B^4 + 5.551 \times 10^{-16}B^5\end{array}}X_t. \tag{13}$$

Table 8. Different Web workload characteristics and their log transformations along with the occurrences of different errors having an impact on the reliability of www.ismdhanbad.ac.in.

Group	log(Hits)	X_t	log(Sessions)	log(Users)	$Y_t = \log(\text{errors})$	$\hat{Y}_t$
A_1	4.274134748	8.756348968	0.832508913	2.53403	0.698970004	8.835231618
A_2	4.049760552	8.461269125	0.77815125	2.5302	0.77815125	8.816059857
A_3	4.272491401	8.757324228	0.806179974	2.53275	0.77815125	8.816059857
A_4	4.080410007	8.519577848	0.740362689	2.47857	0.602059991	8.859675815
A_5	4.092720645	8.534136513	0.812913357	2.53403	0.77815125	8.816059857
A_6	4.070887144	8.422097758	0.748188027	2.4609	0.698970004	8.835231618
A_7	4.006294858	8.365783605	0.698970004	2.43297	0.602059991	8.859675815
A_8	3.979275148	8.346017895	0.698970004	2.43616	0.84509804	8.80039534
B_1	4.295413146	8.717581276	0.892094603	2.60314	0.77815125	8.816059857
B_2	4.200604292	8.662325835	0.812913357	2.53148	0.602059991	8.859675815
B_3	4.240599164	8.751858023	0.86332286	2.59218	0.698970004	8.835231618
B_4	4.063333359	8.493365208	0.73239376	2.46389	0.602059991	8.859675815
B_5	4.305222352	8.732705216	0.908485019	2.63548	0.698970004	8.835231618
B_6	4.085861174	8.639720656	0.792391689	2.47857	0.77815125	8.816059857
B_7	4.276139985	8.759314859	0.897627091	2.62428	0.698970004	8.835231618
B_8	4.041511113	8.592867935	0.716003344	2.4624	0.602059991	8.859675815
B_9	4.129818744	8.637703046	0.799340549	2.51055	0.698970004	8.835231618
B_{10}	4.03702788	8.510355993	0.707570176	2.4609	0.698970004	8.835231618
B_{11}	4.029261996	8.447051793	0.69019608	2.44404	0.698970004	8.835231618
B_{12}	4.035869814	8.511780853	0.72427587	2.45484	0.77815125	8.816059857
B_{13}	4.266466895	8.751655647	0.886490725	2.60314	0.602059991	8.859675815
B_{14}	4.144075806	8.643302801	0.799340549	2.48996	0.698970004	8.835231618
B_{15}	4.305415864	8.764782707	0.913813852	2.65031	0.602059991	8.859675815
B_{16}	4.114310677	8.647530193	0.806179974	2.52244	0.77815125	8.816059857
A_9	4.292743271	8.780751439	0.897627091	2.617	0.698970004	8.835231618
A_{10}	4.095866453	8.630975554	0.812913357	2.52504	0.77815125	8.816059857
C_1	4.261524556	8.730982679	0.929418926	2.65992	0.698970004	8.835231618
C_2	3.806451323	8.313155225	0.568201724	2.35025	0.698970004	8.835231618
C_3	4.162833144	8.614682797	0.84509804	2.48572	0.602059991	8.859675815
C_4	4.135546068	8.59011298	0.832508913	2.47857	0.698970004	8.835231618
C_5	4.02514195	8.526629471	0.672097858	2.42651	0.698970004	8.835231618
C_6	3.94512377	8.398251897	0.591064607	2.40824	0.698970004	8.835231618
C_7	4.386570302	8.774092943	0.968482949	2.69285	0.84509804	8.80039534
C_8	4.093456707	8.564873619	0.785329835	2.49693	0.77815125	8.816059857
...	...	...	...	...	...	...

Table 9. The correlation matrix for www.ismdhanbad.ac.in.

	Hits count	Bytes count	Sessions count	User count
Hits count	1	0.93	0.93	0.92
Bytes count	0.93	1	0.89	0.87
Sessions count	0.93	0.89	1	0.95
User count	0.92	0.87	0.95	1



(a) (b)

Fig. 2. (a) Scree plot for the workload characteristics of www.ismdhanbad.ac.in. (b) Scree plot for the Web workload characteristics for www.isical.ac.in.

Here, in this case, X_t is not a deterministic function of Y_t and also disturbed by some noise. In order to eliminate the noise components from the inputs, i.e., the log(12 h Bytes transferred_count) (i.e., X_t), a hypothesis $\hat{Y}_t = 7 \times 10^{-8}X^6 - 8 \times 10^{-6}X^5 - 0.007X^3 + 0.072X^2 - 0.337X + 9.038$ has been designed and the calculated noise component (i.e., $N_t = \hat{Y}_t - Y_t$) corresponding to each occurrence has been tabulated in Table 8. Here, $\hat{Y}_t$ is the estimated log(cumulative occurrences of errors) having an impact on the reliability of www.ismdhanbad.ac.in. The corresponding independently and identically distributed (iid) noise series ($a_t \sim (0, 0.019)$) of www.ismdhanbad.ac.in has been tabulated in Table 8. For simplicity, we have assumed a_t is same as η_t. Then, the transfer function model between the noise component (N_t) and a_t (as given in Eq. (5)) can be expressed as given in Eq. (14).

$$N_t = \frac{\theta(B)}{\phi(B)}a_t = \frac{\begin{array}{c}1 + 8.327 \times 10^{-17}B + 1.388 \times 10^{-16}B^2 \\ + 7.355 \times 10^{-16}B^3 - 4.163 \times 10^{-16}B^4\end{array}}{\begin{array}{c}1 + 8.327 \times 10^{-17}B - 4.996 \times 10^{-16}B^2 - 4.302 \times 10^{-16}B^3 \\ + 2.776 \times 10^{-16}B^4 + 4.163 \times 10^{-16}B^5\end{array}}a_t. \tag{14}$$

The order of the above ARMA model has been decided by analyzing the partial ACF and the ACF. Hence, from Eq. (6), the final transfer function model between the output (Y_t) and the input (X_t), the noise (a_t) after analyzing the partial ACF, ACF and the cross-correlation function (given in Eq. (6)), can be expressed as given

in Eq. (15).

$$Y_t = \frac{\begin{array}{c}1 + 6.661 \times 10^{-16}B + 6.384 \times 10^{-16}B^2 - 2.442 \times 10^{-15}B^3 \\ + 4.163 \times 10^{-16}B^4 - 4.441 \times 10^{-16}B^5\end{array}}{\begin{array}{c}1 - 8.882 \times 10^{-16}B - 6.661 \times 10^{-16}B^2 + 2.47 \times 10^{-15}B^3 \\ - 3.775 \times 10^{-15}B^4 + 5.551 \times 10^{-16}B^5\end{array}} X_t$$

$$+ \frac{\begin{array}{c}1 + 8.327 \times 10^{-17}B + 1.388 \times 10^{-16}B^2 \\ + 7.355 \times 10^{-16}B^3 - 4.163 \times 10^{-16}B^4\end{array}}{\begin{array}{c}1 + 8.327 \times 10^{-17}B - 4.996 \times 10^{-16}B^2 - 4.302 \times 10^{-16}B^3 \\ + 2.776 \times 10^{-16}B^4 + 4.163 \times 10^{-16}B^5\end{array}} a_t. \tag{15}$$

From Table 8, we can find that X_t and N_t are serially correlated. Hence, in order to get the consistent estimates of $v_j s$, the pre-whitening technique, proposed in Sec. 2.3.1, has been applied. Hence, we have to develop the univariate ARMA model between the input variable (X_t) and the white noise (η_t) as $\phi(B)X_t = \theta(B)\eta_t$. For simplicity, we have assumed η_t and a_t are same. Hence, the above equation will be changed to $\phi(B)X_t = \theta(B)a_t$. After analyzing the partial ACF, ACF and cross-correlation we have decided it as a ARMA(2,4) process, where, $\phi(B) = 1 + 2.644 \times 10^{-16}B + 4.624 \times 10^{-17}B^2$ and $\theta(B) = 1 - 2.367 \times 10^{-16}B + 2.868 \times 10^{-16}B^2 + 4.4.5 \times 10^{-16}B^3$. Hence, as described in Sec. 2.3.1, a_t can be expressed as

$$a_t = \frac{\phi(B)}{\theta(B)} X_t = \frac{1 + 2.644 \times 10^{-16}B + 4.624 \times 10^{-17}B^2}{1 - 2.367 \times 10^{-16}B + 2.868 \times 10^{-16}B^2 + 4.4.5 \times 10^{-16}B^3} X_t. \tag{16}$$

Similarly, y_t and n_t can be expressed as follows (given in Sec. 2.3.1):

$$y_t = \frac{\phi(B)}{\theta(B)} Y_t = \frac{1 + 2.644 \times 10^{-16}B + 4.624 \times 10^{-17}B^2}{1 - 2.367 \times 10^{-16}B + 2.868 \times 10^{-16}B^2 + 4.4.5 \times 10^{-16}B^3} Y_t, \tag{17}$$

$$n_t = \frac{\phi(B)}{\theta(B)} N_t = \frac{1 + 2.644 \times 10^{-16}B + 4.624 \times 10^{-17}B^2}{1 - 2.367 \times 10^{-16}B + 2.868 \times 10^{-16}B^2 + 4.4.5 \times 10^{-16}B^3}$$

$$\times \frac{\begin{array}{c}1 + 8.327 \times 10^{-17}B + 1.388 \times 10^{-16}B^2 \\ + 7.355 \times 10^{-16}B^3 - 4.163 \times 10^{-16}B^4\end{array}}{\begin{array}{c}1 + 8.327 \times 10^{-17}B - 4.996 \times 10^{-16}B^2 - 4.302 \times 10^{-16}B^3 \\ + 2.776 \times 10^{-16}B^4 + 4.163 \times 10^{-16}B^5\end{array}} a_t. \tag{18}$$

Hence the final transfer function model, after analyzing the partial ACF, ACF and the cross correlation function (as given in Eq. (9)) can be expressed as given in Eq. (19).

$$y_t = \frac{1 + 2.644 \times 10^{-16}B + 4.624 \times 10^{-17}B^2}{1 - 2.367 \times 10^{-16}B + 2.868 \times 10^{-16}B^2 + 4.4.5 \times 10^{-16}B^3}$$

$$\times \frac{\begin{array}{c}1 + 6.661 \times 10^{-16}B + 6.384 \times 10^{-16}B^2 - 2.442 \times 10^{-15}B^3 \\ + 4.163 \times 10^{-16}B^4 - 4.441 \times 10^{-16}B^5\end{array}}{\begin{array}{c}1 - 8.882 \times 10^{-16}B - 6.661 \times 10^{-16}B^2 + 2.47 \times 10^{-15}B^3 \\ - 3.775 \times 10^{-15}B^4 + 5.551 \times 10^{-16}B^5\end{array}} X_t$$

$$+ \frac{1 + 2.644 \times 10^{-16}B + 4.624 \times 10^{-17}B^2}{1 - 2.367 \times 10^{-16}B + 2.868 \times 10^{-16}B^2 + 4.4.5 \times 10^{-16}B^3}$$

$$\times \frac{\begin{array}{c}1 + 8.327 \times 10^{-17}B + 1.388 \times 10^{-16}B^2 \\ + 7.355 \times 10^{-16}B^3 - 4.163 \times 10^{-16}B^4\end{array}}{\begin{array}{c}1 + 8.327 \times 10^{-17}B - 4.996 \times 10^{-16}B^2 - 4.302 \times 10^{-16}B^3 \\ + 2.776 \times 10^{-16}B^4 + 4.163 \times 10^{-16}B^5\end{array}} a_t. \tag{19}$$

The above equation can be written as: $y_t = c^* + v(B)a_t + n_t$ which is similar to Eq. (9). Find the consistent estimates of $v_j's$ as given in Sec. 2.3.1 and predict the remaining instances of the occurrences of the Web errors having an impact on the reliability of the Web software. The predicted remaining log(cumulative occurrences of the Web errors) having an impact on the reliability of www.ismdhanbad.ac.in are tabulated in Table 10. Figure 3 represents the original versus predicted log(cumulative occurrences of the Web errors) having an impact on the reliability of www.ismdhanbad.ac.in.

Table 11 shows consecutive 12 h occurrences of different Web workload characteristics of www.isical.ac.in extracted from the HTTP logs (access and error logs) of the corresponding Web server.

Here, $X_{1t} = \log(\text{Hits})$, $X_{2t} = \log(\text{BytesTransferred})$, $X_{3t} = \log(\text{Users})$, $X_{4t} = \log(\text{Sessions})$ and Y_t stands for the log of cumulative occurrences of errors having an

Table 10. Original and predicted occurrences of errors of remaining 16 instances of www.ismdhanbad.ac.in.

Original log (errors)	Predicted log (errors)	Predicted log (errors) by VAR(1)
0.77815125	0.785329835	0.5835
0.778319738	0.790419738	0.5308
0.76366604	0.77576604	0.5604
0.71413938	0.72623938	0.5738
0.713488986	0.725588986	0.5486
0.764014096	0.776114096	0.5096
0.769613955	0.780713955	0.5955
0.731287816	0.728387816	0.5816
0.728034937	0.740134937	0.5937
0.700854588	0.687854588	0.5588
0.713746042	0.720746042	0.5042
0.710708582	0.737708582	0.5582
0.690941496	0.681941496	0.4496
0.692844081	0.704944081	0.4481
0.68601564	0.69811564	0.4564
0.659255484	0.671355484	0.4484
0.672562929	0.681662929	0.4929

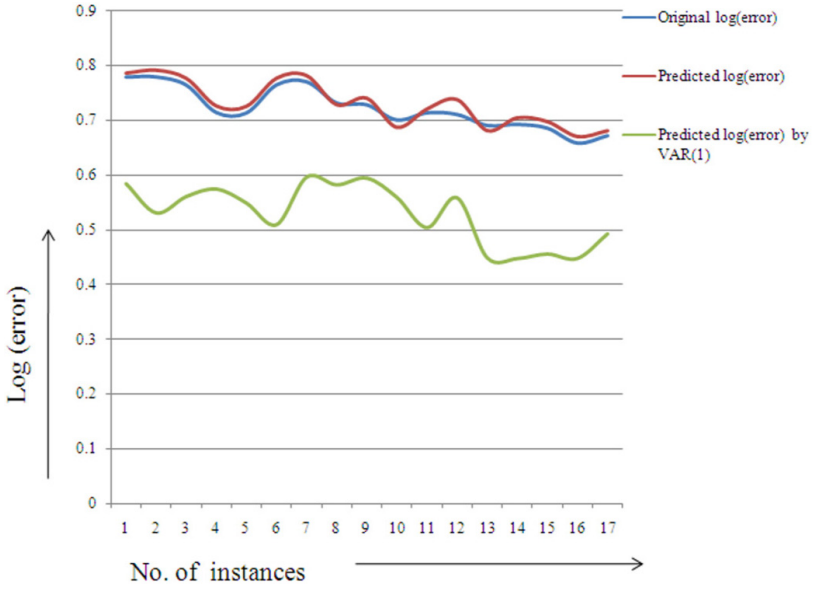


Fig. 3. Original versus predicted errors (by the proposed method and the VAR(1) model) having an impact on the reliability of www.ismdhanbad.ac.in.

impact on the reliability of www.isical.ac.in. In order to determine if any correlation among the different Web workload characteristics exists or not, PCA has been performed with the data given in first four columns of Table 11. Figure 2(b) shows the corresponding Scree plot. The "elbow" in the red line at PC3 in Fig. 2(b) shows that the PC1, PC2 and PC3 are the most important. However, Website administrators should select those workload characteristics most suitable for their requirements. In this case, we have taken the $X_{1t} = \log(\text{Hits})$, $X_{2t} = \log(\text{Bytes Transferred})$ and $X_{3t} = \log(\text{Users})$ as the inputs and predict the log(cumulative occurrences of all the Web errors) having an impact on the reliability of www.isical.ac.in. Here, first 44 observations of $X_{1t} = \log(\text{Hits})$, $X_{2t} = \log(\text{Bytes Transferred})$, $X_{3t} = \log(\text{Users})$ and Y_t with an interval of 12 h have been taken for the model fitting purpose and the remaining 24 observations of Y_t with an interval of 12 h for the prediction. In this case, we apply Eq. (7) for the transfer function modeling. Analyzing the ACF and the partial ACF, the transfer function model between the output (Y_t) and $X_{1t} = \log(\text{Hits})$, $X_{2t} = \log(\text{Bytes Transferred})$ and $X_{3t} = \log(\text{Users})$ is given as follows:

$$Y_t = \frac{\begin{array}{c}1 - 3.015 \times 10^{-16}B - 5.702 \times 10^{-17}B^2 + 1.664 \times 10^{-16}B^3 \\ - 1.589 \times 10^{-15}B^4 + 1.088 \times 10^{-16}B^5\end{array}}{\begin{array}{c}1 - 5.831 \times 10^{-16}B - 1.65 \times 10^{-16}B^2 \\ - 1.942 \times 10^{-16}B^3 - 1.887 \times 10^{-15}B^4\end{array}} X_{1t}$$

Table 11. Different Web work load characteristics and their log transformations along with the occurrences of different errors having an impact on the reliability of www.isical.ac.in.

Groups	X_{1t}	X_{2t}	X_{3t}	X_{4t}	Y_t	$\hat{Y}_t$
A_1	4.411602873	8.886841741	2.800029359	2.021189299	1.230448921	1.329517053
A_2	4.494349426	8.897213036	2.797959644	1.986771734	1.361727836	1.342735746
A_3	4.458305392	8.887564102	2.799340549	2.004321374	1.361727836	1.336949544
A_4	4.462906872	8.919501404	2.770852012	1.963787827	1.322219295	1.337925928
A_5	4.510276844	8.870454797	2.800029359	2.008600172	1.361727836	1.34507976
A_6	4.502058223	8.822364489	2.761927838	1.968482949	1.342422681	1.343709208
A_7	4.479229695	8.920205779	2.748188027	1.939519253	1.322219295	1.340580557
A_8	4.470322271	8.914781347	2.749736316	1.939519253	1.380211242	1.339151486
B_1	4.473384771	8.858472689	2.838849091	2.06069784	1.361727836	1.339046591
B_2	4.554743484	8.819242892	2.798650645	2.008600172	1.322219295	1.351900642
B_3	4.437782262	8.883520803	2.832508913	2.041392685	1.342422681	1.333579503
B_4	4.499274582	8.909252888	2.763427994	1.959041392	1.322219295	1.343645271
B_5	4.450156695	8.86945902	2.857935265	2.071882007	1.342422681	1.335386948
B_6	4.507667007	8.867016548	2.770852012	1.995635195	1.361727836	1.344692548
B_7	4.460687407	8.889039838	2.851258349	2.064457989	1.342422681	1.337151396
B_8	4.49140372	8.898518278	2.762678564	1.949390007	1.322219295	1.342387691
B_9	4.537617636	8.865822202	2.787460475	2	1.342422681	1.349406875
B_{10}	4.489817908	8.859653989	2.761927838	1.944482672	1.342422681	1.341996397
B_{11}	4.487095934	8.892056469	2.753583059	1.934498451	1.342422681	1.341725888
B_{12}	4.475104348	8.916413569	2.758911892	1.954242509	1.361727836	1.339869414
B_{13}	4.585122186	8.936759579	2.838849091	2.056904851	1.322219295	1.357025629
B_{14}	4.530635056	8.903007562	2.776701184	2	1.342422681	1.348452369
B_{15}	4.465427468	8.893102357	2.866877814	2.075546961	1.322219295	1.337883308
B_{16}	4.518658681	8.894399445	2.793790385	2.004321374	1.361727836	1.346512705
A_9	4.471614378	8.894096128	2.846955325	2.064457989	1.342422681	1.338898082
A_{10}	4.511482289	8.901751831	2.79518459	2.008600172	1.361727836	1.345395408
C_1	4.451187525	8.896663051	2.872738827	2.086359831	1.342422681	1.335615202
C_2	4.421669724	8.900728919	2.710117365	1.86923172	1.342422681	1.331619037
C_3	4.538435481	8.909447248	2.774516966	2.029383778	1.322219295	1.349622444
C_4	4.527152817	8.913361509	2.770852012	2.021189299	1.342422681	1.347880023
C_5	4.499632105	8.922322934	2.745074792	1.924279286	1.342422681	1.343858326
C_6	4.529083705	8.929510681	2.736396502	1.880813592	1.342422681	1.348670827
C_7	4.535977311	8.900049817	2.893206753	2.113943352	1.380211242	1.348942784
C_8	4.510558414	8.938106967	2.780317312	1.991226076	1.361727836	1.345446979
	...	...	...	...	...	...

$$+\frac{1-1.305\times10^{-15}B+2.22\times10^{-16}B^2-1.943\times10^{-16}B^3}{1-1.332\times10^{-15}B-3.053\times10^{-16}B^2 \\ -2.776\times10^{-17}B^3+1.735\times10^{-17}B^4}X_{2t}$$

$$+\frac{1-1.082\times10^{-15}B+1.388\times10^{-16}B^2 \\ +8.327\times10^{-17}B^3-1.232\times10^{-16}B^4}{1-1.554\times10^{-15}B-5.551\times10^{-16}B^2 \\ +2.776\times10^{-16}B^3+1.232\times10^{-16}B^4}X_{3t}. \tag{20}$$

After analyzing the partial ACF, ACF and cross-correlation function we have decided the above transfer function model. In order to eliminate the noise

components from the input series, a multiple regression model $\hat{y} = 0.603 + 0.158X_{1t} + 0.004X_{2t} - 0.003X_{3t}$ has been fitted and the corresponding noise components (i.e., $N_t = \hat{Y}_t - Y_t$) having standard deviation 0.1442 have been tabulated in Table 10. Here, X_{1t}, X_{2t}, X_{3t} are the input series for www.isical.ac.in and $\hat{Y}_t$ is the estimated cumulative occurrence of different errors having an impact on its reliability. The corresponding independently and identically distributed (iid) noise series ($a_t \sim (0, 0.019)$) of www.isical.ac.in has been tabulated in Table 11. The ARMA model between the noise component and the iid noise is given as follows:

$$N_t = \frac{\theta(B)}{\phi(B)}a_t = \frac{1 - 1.804 \times 10^{-16}B + 8.327 \times 10^{-17}B^2}{1 + 2.22 \times 10^{-16}B + 1.943 \times 10^{-16}B^2 \atop - 1.11 \times 10^{-16}B^3 - 1.11 \times 10^{-16}B^4}a_t. \tag{21}$$

Using Eq. (7), the transfer function model for www.isical.ac.in is given in Eq. (22):

$$\begin{aligned} Y_t = {} & \frac{1 - 3.015 \times 10^{-16}B - 5.702 \times 10^{-17}B^2 + 1.664 \times 10^{-16}B^3 \atop - 1.589 \times 10^{-15}B^4 + 1.088 \times 10^{-16}B^5}{1 - 5.831 \times 10^{-16}B - 1.65 \times 10^{-16}B^2 - 1.942 \times 10^{-16}B^3 \atop - 1.887 \times 10^{-15}B^4}X_{1t} \\ & + \frac{1 - 1.305\times10^{-15}B + 2.22\times10^{-16}B^2 - 1.943 \times 10^{-16}B^3}{1 - 1.332\times10^{-15}B - 3.053 \times 10^{-16}B^2 \atop - 2.776 \times 10^{-17}B^3 + 1.735 \times 10^{-17}B^4}X_{2t} \\ & + \frac{1 - 1.082\times10^{-15}B + 1.388\times10^{-16}B^2 \atop + 8.327 \times 10^{-17}B^3 - 1.232 \times 10^{-16}B^4}{1 - 1.554\times10^{-15}B - 5.551 \times 10^{-16}B^2 \atop + 2.776 \times 10^{-16}B^3 + 1.232 \times 10^{-16}B^4}X_{3t} \\ & + \frac{1 - 1.804 \times 10^{-16}B + 8.327 \times 10^{-17}B^2}{1 + 2.22 \times 10^{-16}B + 1.943 \times 10^{-16}B^2 - 1.11 \times 10^{-16}B^3 \atop - 1.11 \times 10^{-16}B^4}a_t. \end{aligned} \tag{22}$$

From Table 11, we can find that $X_{it}(i = 1, 2, 3)$ and N_t are serially correlated. Hence, in order to get the consistent estimates of $v_{lj}s$, the pre-whitening technique for multiple inputs and single output, proposed in Sec. 2.3.1, has been applied. Hence, we have to develop the univariate ARMA model between the input variable (X_{it}) and the white noise (η_t) as $\phi_i(B)X_{it} = \theta_i(B)\eta_t$. For simplicity, we have assumed η_t and a_t are same. Hence, the above equation will be changed to $\phi_i(B)X_{it} = \theta_i(B)a_t$, i.e., $\phi_1(B)X_{1t} = \theta_1(B)a_t, \phi_2(B)X_{2t} = \theta_2(B)a_t, \phi_3(B)X_{3t} = \theta_3(B)a_t$.

Then, $a_t = \frac{\phi_1(B)}{\theta_1(B)}X_{1t}, a_t = \frac{\phi_2(B)}{\theta_2(B)}X_{2t}$ and $a_t = \frac{\phi_3(B)}{\theta_3(B)}X_{3t}$. Equating them, we get:

$$X_{2t} = \frac{\phi_1(B)}{\theta_1(B)}\frac{\theta_2(B)}{\phi_2(B)}X_{1t} \quad \text{and} \quad X_{3t} = \frac{\phi_1(B)}{\theta_1(B)}\frac{\theta_3(B)}{\phi_3(B)}X_{1t}.$$

The transfer function model between X_{1t} and a_t is given as follows (after analyzing the partial ACF, ACF and cross-correlation function):

$$X_{1t} = \frac{\theta_1(B)}{\phi_1(B)} a_t$$

$$= \frac{1 - 5.551 \times 10^{-17} B + 5.551 \times 10^{-17} B^2}{\begin{array}{c} 1 - 8.327 \times 10^{-16} B - 4.163 \times 10^{-17} B^2 - 1.11 \times 10^{-16} B^3 \\ + 4.158 \times 10^{-32} B^4 - 1.11 \times 10^{-16} B^5 - 5.551 \times 10^{-16} B^6 \end{array}} a_t.$$

The transfer function model between X_{2t} and a_t is given as follows (after analyzing the partial ACF, ACF and cross-correlation function):

$$X_{2t} = \frac{\theta_2(B)}{\phi_2(B)} a_t = \frac{1 - 9.437 \times 10^{-17} B + 2.914 \times 10^{-16} B^2}{1 + 5.551 \times 10^{-17} B - 2.776 \times 10^{-16} B^2} a_t.$$

The transfer function model between X_{3t} and a_t is given as follows (after analyzing the partial ACF, ACF and cross-correlation function):

$$X_{3t} = \frac{\theta_3(B)}{\phi_3(B)} a_t = \frac{1 - 1.929 \times 10^{-15} B + 1.055 \times 10^{-15} B^2}{\begin{array}{c} 1 + 4.829 \times 10^{-15} B - 8.327 \times 10^{-16} B^2 - 2.304 \times 10^{-15} B^3 \\ - 4.441 \times 10^{-16} B^4 - 4.33 \times 10^{-15} B^5 \end{array}} a_t.$$

Now, applying

$$\frac{\begin{array}{c} 1 - 8.327 \times 10^{-16} B - 4.163 \times 10^{-17} B^2 - 1.11 \times 10^{-16} B^3 \\ + 4.158 \times 10^{-32} B^4 - 1.11 \times 10^{-16} B^5 - 5.551 \times 10^{-16} B^6 \end{array}}{1 - 5.551 \times 10^{-17} B + 5.551 \times 10^{-17} B^2}$$

in Eq. (22) we get the final transfer function model which is given in Eq. (23) as

$$y_t = \frac{\begin{array}{c} 1 - 8.327 \times 10^{-16} B - 4.163 \times 10^{-17} B^2 - 1.11 \times 10^{-16} B^3 \\ + 4.158 \times 10^{-32} B^4 - 1.11 \times 10^{-16} B^5 - 5.551 \times 10^{-16} B^6 \end{array}}{1 - 5.551 \times 10^{-17} B + 5.551 \times 10^{-17} B^2}$$

$$\times \frac{\begin{array}{c} 1 - 3.015 \times 10^{-16} B - 5.702 \times 10^{-17} B^2 + 1.664 \times 10^{-16} B^3 \\ - 1.589 \times 10^{-15} B^4 + 1.088 \times 10^{-16} B^5 \end{array}}{\begin{array}{c} 1 - 5.831 \times 10^{-16} B - 1.65 \times 10^{-16} B^2 \\ - 1.942 \times 10^{-16} B^3 - 1.887 \times 10^{-15} B^4 \end{array}} X_{1t}$$

$$+ \frac{\begin{array}{c} 1 - 8.327 \times 10^{-16} B - 4.163 \times 10^{-17} B^2 - 1.11 \times 10^{-16} B^3 \\ + 4.158 \times 10^{-32} B^4 - 1.11 \times 10^{-16} B^5 - 5.551 \times 10^{-16} B^6 \end{array}}{1 - 5.551 \times 10^{-17} B + 5.551 \times 10^{-17} B^2}$$

$$\times \frac{1 - 1.305 \times 10^{-15} B + 2.22 \times 10^{-16} B^2 - 1.943 \times 10^{-16} B^3}{\begin{array}{c} 1 - 1.332 \times 10^{-15} B - 3.053 \times 10^{-16} B^2 \\ - 2.776 \times 10^{-17} B^3 + 1.735 \times 10^{-17} B^4 \end{array}} X_{2t}$$

$$
\begin{aligned}
&+\frac{\begin{array}{l}1-8.327\times10^{-16}B-4.163\times10^{-17}B^2-1.11\times10^{-16}B^3\\ \quad+4.158\times10^{-32}B^4-1.11\times10^{-16}B^5-5.551\times10^{-16}B^6\end{array}}{1-5.551\times10^{-17}B+5.551\times10^{-17}B^2}\\
&\times\frac{\begin{array}{l}1-1.082\times10^{-15}B+1.388\times10^{-16}B^2\\ \quad+8.327\times10^{-17}B^3-1.232\times10^{-16}B^4\end{array}}{\begin{array}{l}1-1.554\times10^{-15}B-5.551\times10^{-16}B^2\\ \quad+2.776\times10^{-16}B^3+1.232\times10^{-16}B^4\end{array}}X_{3t}\\
&+\frac{\begin{array}{l}1-8.327\times10^{-16}B-4.163\times10^{-17}B^2-1.11\times10^{-16}B^3\\ \quad+4.158\times10^{-32}B^4-1.11\times10^{-16}B^5-5.551\times10^{-16}B^6\end{array}}{1-5.551\times10^{-17}B+5.551\times10^{-17}B^2}\\
&\times\frac{1-1.804\times10^{-16}B+8.327\times10^{-17}B^2}{\begin{array}{l}1+2.22\times10^{-16}B+1.943\times10^{-16}B^2\\ \quad-1.11\times10^{-16}B^3-1.11\times10^{-16}B^4\end{array}}a_t.
\end{aligned}
\tag{23}
$$

Then finding the consistent estimates of $v'_{ij}s(j=1,2,3)$ of the above equation as given in Sec. 2.3.1 predict the remaining instances log(original occurrences of the Web errors) having an impact on the reliability of the Web software. The predicted occurrences of the errors having an impact on the reliability of www.isical.ac.in are tabulated in Table 12. Figure 4 represents the graph of log(original occurrences of the Web errors) and log(predicted occurrences of the Web errors) having an impact on the reliability of www.isical.ac.in.

4.1. *Data analysis using MANOVA*

The dependence of different independent variable on each other as also on the dependent variable can also be determined using the MANOVA test on the data sets of www.ismdhanbad.ac.in and www.isical.ac.in, which are discussed in the following subsections.

4.1.1. *ISM Dhanbad dataset*

This subsection demonstrates the MANOVA test on the data sets of different Web workload characteristics as also the occurrences of different Web errors having an influence on the reliability of the www.ismdhanbad.ac.in, shown in Table 8.

In case of www.ismdhanbad.ac.in, the 50 days' data set (subjects of the study) can be divided into three groups, viz., $\{A_t\}_{t=1}^{15}$, $\{B_t\}_{t=1}^{15}$ and $\{C_t\}_{t=1}^{20}$, containing 15, 15 and 20 elements respectively, after analyzing the nature of different days' occurrences of several Web errors having an influences on the reliability of the aforementioned Website. Here, $\{A_t\}$ = "occurrences of only SCF_Admin", $\{B_t\}$ = "occurrences of SCF_Older" and $\{C_t\}$ = "occurrences of both SCF_Admin and SCF_Older". Moreover, there are four independent variables, e.g., the number of hits, amount of bytes transferred, number of hits and the number of generated

Table 12. Original and predicted occurrences of errors of remaining 24 instances of www.isical.ac.in.

Original log (errors)	Predicted log (errors)	Predicted log (errors) by TF	Nonlinear regression model
1.234448921	1.224517053	2.253	2.573
1.361727836	1.362735746	2.346	2.476
1.361727836	1.362735746	2.346	2.456
1.332219295	1.337925928	2.388	2.388
1.361727836	1.365079766	2.366	2.686
1.342422681	1.343709208	2.088	2.088
1.332219295	1.340580557	2.357	2.597
1.340211242	1.339151486	2.386	2.896
1.341727836	1.339046591	2.391	2.971
1.352219295	1.351900642	2.342	2.482
1.342422681	1.333579503	2.303	2.083
1.342219295	1.343645271	2.371	2.781
1.342422681	1.335386948	2.348	2.448
1.341727836	1.344692548	2.348	2.678
1.342422681	1.337151396	2.966	2.966
1.329919295	1.342387691	2.391	2.951
1.342422681	1.349406875	2.375	2.745
1.342422681	1.341996397	2.397	2.497
1.342422681	1.341725888	2.388	2.868
1.351727836	1.339869414	2.314	2.184
1.362219295	1.357025629	2.329	2.929
1.342422681	1.348452369	2.369	2.069
1.322219295	1.337883308	2.308	2.308
1.351727836	1.346512705	2.305	2.705

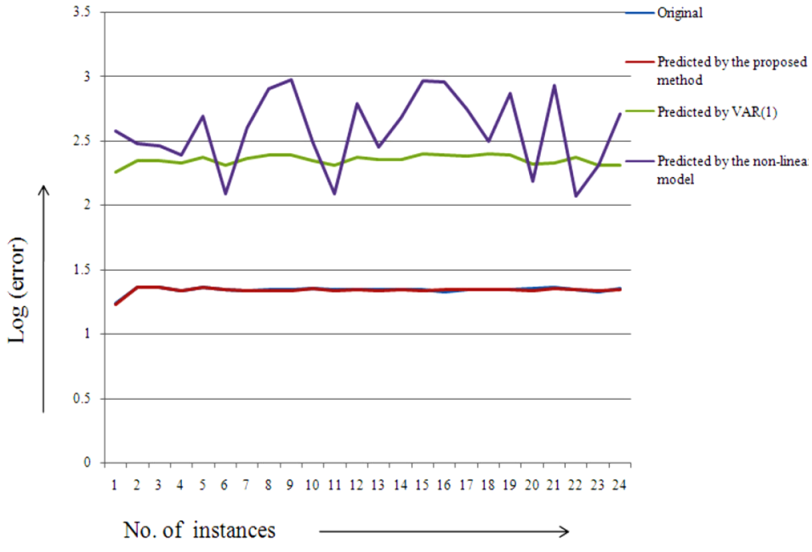


Fig. 4. Original versus predicted errors having an impact on the reliability of www.isical.ac.in.

sessions. Here, the object of the study is to discover whether these variables are significantly different for the aforementioned three groups and, as a consequence, the MANOVA test has been applied. For this purpose, in the present study, the authors have used the SPSS 20 and three ANOVAs could be carried out, i.e., one ANOVA with each of $\{A_t\}, \{B_t\}$ and $\{C_t\}$. However, it may be possible that all the three variables are highly correlated. Thus, the three analyses are not independent, whilst, the independent ANOVA ignores the interrelation between variables. Consequently, substantial information may be lost, and the resultant p values for tests of hypothesis for the three independent ANOVAs are incorrect.

All four aforementioned tests explore whether the means for each of the groups are the same. The first line contains the values of the parameters (A_t, B_t, C_t) used to discover significant levels in tables of the exact distributions of the statistics. For the first three tests, the value of the test statistic is given, followed by its transformation to a statistic that has approximately a F distribution. The next two columns contain the numerator (hypothesis) and denominator (Error) degrees of freedom for the F statistic. The next column gives us the observed significance levels which are translated as the probability of observing a difference at least as large as the one found in the sample when there is no difference in the populations. In our case, due to the significance values of 0.000 we can conclude that the null hypothesis — that there is no difference is rejected. Therefore, we know that there are significant differences between the three groups on the means of the four variables, i.e., all or some subsets of the independent variables may have influence on the different groups of the dependent variables.

Again, we can also perform some tests of homogeneity in order to know whether the variances for each of the groups (dependent variables) are the same, as follows:

Univariate Homogeneity of Variance Tests[25–28]
Independent Variable 1: Number of Users
Cochrans $C = 0.561, p = 0.071$ (approx.)

Table 13. The result of the multivariate test.

Effect	Value	F	Hypothesis d.o.f	Error d.o.f	Sig.
Intercept					
Pillai's Trace	0.956	249.681	3.000	0.000	0.0000
Wilk's X	0.054	249.681	3.000	0.000	0.0000
Hotelling's Trace	17.721	249.681	3.000	0.000	0.0000
Roy's Largest Root	17.721	249.681	3.000	0.000	0.0000
Group					
Pillai's Trace	0.464	2.683	9.000	0.000	0.0000
Wilk's X	0.592	2.737	9.000	0.000	0.0000
Hotelling's Trace	0.597	2.698	9.000	0.000	0.0000
Roy's Largest Root	0.326	4.777	3.000	0.000	0.0000

Bartlett-Box F (Refs. 25–28) $= 1.780, p = 0.16$
Independent Variable 2: Number of Bytes Transferred
Cochrans $C = 0.61, p = 0.37$ (approx.)
Bartlett-Box $F = 0.732, p = 0.391$
Independent Variable 3: Number of Hits
Cochrans $C = 0.688, p = 0.071$ (approx.)
Bartlett-Box $F = 1.890, p = 0.205$
Independent Variable 4: Number of Sessions generated
Cochrans $C = 0.65, p = 0.069$ (approx.)
Bartlett-Box $F = 1.187, p = 0.102$.

As can be seen through the above given Cochran's C (Refs. 25–28) and the Bartlett Box F tests, the significance levels indicate that there is no reason to reject the hypotheses that the variances in the three groups are equal (all values are greater than 0.05). The aforementioned tests are univariate and are a convenient starting point for examining homogeneity (covariance); however, we also need to simultaneously consider both the variances and the covariances. Consequently, "Box's M" test can be applicable in this case. Box's M is based on the determinants of the variance–covariance matrices in each cell, as well as the pooled variance–covariance matrix. Thus, Box's M[25–28] provides us with a multivariate test for homogeneity. The results of the Box's test of homogeneity are given as follows in Table 14:

$p = 0.488 > 0.5$ (approx.), i.e., we have no reason to suspect that homogeneity has been violated (values greater than 0.05). Hence, it can safely be concluded that the dependent variables are correlated and, as a consequence, multivariate analysis of variance is a procedure used when the dependent variables are correlated. The following table demonstrates the tests between the dependent and the independent variables, i.e., tests between the subjects and effects.

From Table 15, it can be found that no independent variables have significant influences on any of the groups of the dependent variable; however, the significant influence of one independent variable has been detected by PCA.[25–28] Hence, for forecasting purpose, using MANOVA,[25–28] a VAR(1) model has been fitted instead of a transfer function model and the construction of the model is given as follows:

The basic requirement of the VAR(1) model is that the series must be stationary. For stationarity checking purpose, Dickey–Fuller test has been carried out on the three dependent variables, i.e., $\{A_t\}_{t=1}^{15}$, $\{B_t\}_{t=1}^{15}$ and $\{C_t\}_{t=1}^{20}$, and found that all of

Table 14. Box's M test result.

Box's M	14.539
F	0.939
dof of A	14
dof of B	14
dof of C	19

Table 15. Tests between subject (independent variables) and effects (dependent variables).

Source			Type III sum of squares	Mean squares	F	Sig.
Corrected model	Dependent variable	Hits	20.576	10.288	1.132	0.0122
		Bytes	1.121	0.561	0.111	0.0231
		Users	108.121	54.061	5.317	0.0000
		Sessions	85.485	2.742	7.455	0.0078
Intercept	Dependent variable	Hits	22.970	22.970	95.858	0.000
		Bytes	20.742	20.742	341.830	0.000
		Users	304.379	4.379	23.382	0.000
		Sessions	33.333	3.333	11.654	0.000
Groups	Dependent variable	Hits	20.576	10.288	1.132	0.000
		Bytes	1.121	0.561	0.111	0.000
		Users	108.121	54.061	5.317	0.000
		Sessions	85.485	2.742	7.455	0.000
Error	Dependent variable	Hits	72.455	9.087		
		Bytes	17.136	5.034		
		Users	40.500	10.167		
		Sessions	61.182	5.733		
Total	Dependent variable	Hits	16.000			
		Bytes	9.000			
		Users	3.000			
		Sessions	80.000			
Correlated total	Dependent variable	Hits	3.030			
		Bytes	8.258			
		Users	8.621			
		Sessions	6.667			

the series are stationary. Now, the corresponding VAR(1) model is given as follows:

$$\begin{pmatrix} A_t \\ B_t \\ C_t \end{pmatrix} = \begin{pmatrix} 2.349657 \times 10^4 \\ -6.698394 \times 10^5 \\ 6.149738 \end{pmatrix} + \begin{pmatrix} 0.5678847 & 0.02474939 & 63.72188 & -1.979139 \\ -5.134792. & 0.7243941 & 4.358335 \times 10^4 & -1.175460 \\ 2.947201 \times 10^{-6} & -5.157261 \times 10^{-7} & 0.6389664 & 2.135620 \times 10^{-4} \end{pmatrix} * \begin{pmatrix} A_{t-1} \\ B_{t-1} \\ C_{t-1} \end{pmatrix} + \begin{pmatrix} -2.540463 \times 10^2 \\ 5.566206 \times 10^3 \\ -1.377957 \times 10^{-2} \end{pmatrix}.$$

The predicted values are given in Table 11. Figure 3 shows the original and the predicted occurrences of different Web errors having an impact on the reliability of www.ismdhanbad.ac.in by the proposed method and the VAR(1) model and it can safely be concluded that the performance of the proposed method is better than its competitor. Apart from this, the proposed model has been compared to

the forecasted outcome of another nonlinear multiple regression model, $\hat{y} = 0.902 + 0.198X_{1t}^2$, (according to the PCA) and found that the accuracy of the proposed model is better, which is shown in Table 11.

4.1.2. *ISI Kolkata data set*

This subsection demonstrates the MANOVA test on the data sets of different Web workload characteristics as also the occurrences of different Web errors having an influence on the reliability of the www.isical.ac.in, shown in Table 11.

Similarly, in case of www.isical.ac.in, the 75 days' data set (subjects of the study) can be divided into three groups, viz., $\{A_t\}_{t=1}^{15}$, $\{B_t\}_{t=1}^{25}$, $\{C_t\}_{t=1}^{35}$, containing 15, 25 and 35 elements respectively, after analyzing the nature of different days' occurrences of several Web errors having an influences on the reliability of the aforementioned Website. In a very similar manner, $\{A_t\}$ = "occurrences of only SCF_Admin", $\{B_t\}$ = "occurrences of SCF_Older" and $\{C_t\}$ = "occurrences of both SCF_Admin and SCF_Older". Moreover, there are four independent variables, e.g., the number of hits, amount of bytes transferred, number of hits and the number of generated sessions. Here, the object of the study is to discover whether these variables are significantly different for the aforementioned three groups and, as a consequence, the MANOVA test has been applied. Apart from this, different univariate (Cochrans C and Bartlett-Box F) and multivariate (Box's M test) tests of homogeneity have been performed to test the correlations among different groups of the dependent variables and confirm the existence of correlations among different groups of the dependent variable. Consequently, the MANOVA can be performed. The following table demonstrates the tests between the dependent and the independent variables, i.e., tests between the subjects and effects.

From Table 16, it has been found that all the independent variables have influence on the groups of the dependent variables, however, only three components can be found by using PCA. Hence, using the MANOVA, we have fitted a transfer function model having four inputs (four independent variables) and single output (the dependent variable) which is given as follows:

$$
\begin{aligned}
y_t = {} & \frac{1 - 7.338 \times 10^{-15}B - 1.13 \times 10^{-13}B^2 - 2.11 \times 10^{-13}B^3 + 2.12 \times 10^{-22}B^4 - 2.12 \times 10^{-12}B^5 - 2.521 \times 10^{-12}B^6}{1 - 2.521 \times 10^{-12}B + 2.251 \times 10^{-12}B^2} \\
& \times \frac{1 - 3.25 \times 10^{-12}B - 2.72 \times 10^{-12}B^2 + 1.24 \times 10^{-12}B^3 - 1.52 \times 10^{-12}B^4 + 1.02 \times 10^{-12}B^5}{1 - 2.82 \times 10^{-12}B - 1.25 \times 10^{-12}B^2 - 1.9 \times 10^{-12}B^3 - 1.8 \times 10^{-12}B^4} X_{1t} \\
& + \frac{1 - 2.7 \times 10^{-12}B - 2.63 \times 10^{-12}B^2 - 2.12 \times 10^{-12}B^3 + 2.58 \times 10^{-22}B^4 - 2.15 \times 10^{-12}B^5 - 2.52 \times 10^{-16}B^6}{1 - 2.5 \times 10^{-17}B + 2.5 \times 10^{-17}B^2}
\end{aligned}
$$

Table 16. Test between subject (independent variables, i.e., different Web workload characteristics) and effect (dependent variables, i.e., groups).

Source			Type III sum of squares	Mean squares	F	Sig.
Corrected model	Dependent variable	Hits	20.513	11.288	1.532	0.0122
		Bytes	1.21	0.361	0.581	0.0231
		Users	108.11	54.061	6.657	0.0000
		Sessions	55.47	6.742	8.655	0.0078
Intercept	Dependent variable	Hits	22.970	22.970	94.858	0.000
		Bytes	20.742	20.742	341.830	0.000
		Users	304.379	4.379	23.382	0.000
		Sessions	33.333	3.333	11.654	0.000
Groups	Dependent variable	Hits	20.513	11.288	1.532	0.071
		Bytes	1.21	0.361	0.581	0.056
		Users	108.11	54.061	6.657	0.059
		Sessions	55.47	6.742	8.655	0.054
Error	Dependent variable	Hits	75.55	10.087		
		Bytes	37.16	12.034		
		Users	46.500	14.167		
		Sessions	71.182	3.733		
Total	Dependent variable	Hits	15.020			
		Bytes	12.092			
		Users	5.047			
		Sessions	98.000			
Correlated total	Dependent variable	Hits	4.030			
		Bytes	3.28			
		Users	6.64			
		Sessions	5.67			

$$\times \frac{1 - 1.25 \times 10^{-15}B + 1.22 \times 10^{-16}B^2 - 2.943 \times 10^{-16}B^3}{1 - 1.32 \times 10^{-15}B - 3.2 \times 10^{-16}B^2 - 7.776 \times 10^{-17}B^3 + 0.735 \times 10^{-17}B^4} X_{2t}$$

$$+ \frac{1 - 2.37 \times 10^{-16}B - 2.13 \times 10^{-17}B^2 - 1.11 \times 10^{-16}B^3 + 4.58 \times 10^{-32}B^4 - 1.21 \times 10^{-16}B^5 - 5.21 \times 10^{-16}B^6}{1 - 2.21 \times 10^{-17}B + 2.21 \times 10^{-17}B^2}$$

$$\times \frac{1 - 2.082 \times 10^{-15}B + 2.38 \times 10^{-16}B^2 + 2.327 \times 10^{-17}B^3 - 2.232 \times 10^{-16}B^4}{1 - 1.554 \times 10^{-15}B - 5.551 \times 10^{-16}B^2 + 2.776 \times 10^{-16}B^3 + 1.232 \times 10^{-16}B^4} X_{3t}$$

$$+ \frac{1 - 2.7 \times 10^{-16}B - 2.1 \times 10^{-17}B^2 - 1.2 \times 10^{-16}B^3 + 4.28 \times 10^{-32}B^4 - 1.21 \times 10^{-16}B^5 - 5.21 \times 10^{-16}B^6}{1 - 2.5 \times 10^{-17}B + 2.51 \times 10^{-17}B^2}$$

$$\times \frac{1 - 1.2 \times 10^{-15}B + 1.3 \times 10^{-16}B^2 + 8.3 \times 10^{-17}B^3 - 1.2 \times 10^{-16}B^4}{1 - 2.5 \times 10^{-15}B - 2.5 \times 10^{-16}B^2 + 2.26 \times 10^{-16}B^3 + 1.22 \times 10^{-16}B^4} X_{4t}$$

$$+ \frac{\begin{array}{c} 1 - 2.3 \times 10^{-16}B - 4.1 \times 10^{-17}B^2 - 1.2 \times 10^{-16}B^3 \\ + 2.158 \times 10^{-32}B^4 - 1.11 \times 10^{-16}B^5 - 5.2 \times 10^{-16}B^6 \end{array}}{1 - 5.551 \times 10^{-17}B + 5.551 \times 10^{-17}B^2}$$

$$\times \frac{1 - 1.2 \times 10^{-16}B + 8.3 \times 10^{-17}B^2}{1 + 0.22 \times 10^{-16}B + 1.9 \times 10^{-16}B^2 - 2.11 \times 10^{-16}B^3 - 2.11 \times 10^{-16}B^4} a_t.$$

The predicted values are shown in Table 13. Apart from this, a nonlinear regression model using all the independent variables has also been established as follows:

$$\hat{y} = 0.813 + 0.98X_{1t} + 0.8X_{2t}^2 + 0.18X_{3t}^2 + 0.9X_{4t}^2.$$

The forecasted outputs of the aforementioned nonlinear regression model are given in Table 13, which shows the superiority of the proposed method. Another advantage of the proposed method is that, the PCA reduces the dimension of the input data, which decreases the difficulty as well as increases the forecasting accuracy. Graphical representations of different forested values are given in Fig. 4.

5. Performance Analysis of the Proposed Method

In this section, we have done the performance analysis of our proposed model by evaluating some performance measures like, sum square error (SSE), root mean square error (RMSE).

The performance measure SSE given in Refs. 10, 11, 16, 19 and 28 is defined as follows:

$$\text{SSE} = \sum_{i=1}^{n} (x_i - \text{predicted}(x_i))^2,$$

where, x_i is the number original faults and predicted (x_i) is the predicted fault.

The RMSE[10,11,16,19,28] is frequently used measure of differences between values predicted and the original values defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \text{predicted}(x_i))^2},$$

where, x_i is the number original Web errors and predicted(x_i) is the predicted Web errors.

Also, we have performed the χ^2-goodness of fit test to validate the proposed model as follows:

$$\chi^2_{\text{computed}} = \sum_{i=1}^{n} \frac{(y_i - \text{predicted}(y_i))^2}{\text{predicted}(y_i)},$$

where n is the size of the data set.

Table 17. Different performance analysis criteria for the proposed method.

Data set	SSE	RMSE	χ^2_{computed}	AIC
ISM	0.0057715	0.0189925	0.4252	21.205
ISI	0.0017219	0.0327443	0.00080662	23.5

The computed values of SSE, RMSE and χ^2_{computed} for proposed transfer function model are given in Table 17. Again, the AIC[28] values of the transfer function models for both the Websites are calculated for finding the best transfer function models and found that the two proposed models are best.

$\chi^2_{\text{computed}} = 0.4252\,(< 42.980)$ at 24 degrees of freedom and 1% level of significance for the data set corresponding to www.ismdhanbad.ac.in. In case of the dataset corresponding to www.isical.ac.in, $\chi^2_{\text{computed}} = 0.00080662\,(< 41.638)$ at 23 degrees of freedom and 1% level of significance. This establishes the validation of the proposed transfer function model.

6. Conclusion

The proposed transfer function model very well demonstrates the use of time series analysis for studying the interrelationship between single or multiple Web workload characteristics and the number of Web errors having an impact on the reliability of particular Web software. The results obtained in the previous sections establish the fact that, transfer function modeling is a better time series tool for the prediction of remaining Web faults. It is better time series tool because transfer function models are useful for prediction of faults present in the software as well as for prediction of different Web workload characteristics. Also, the proposed model is assumption free. The proposed time series model will help the decision maker to assess the correlation of faults. Results obtained showed that the method can be used for any Web software failure data.

Acknowledgments

The authors are thankful to Mr. Rajesh Mishra, System Manager, Indian School of Mines (ISM) Dhanbad, for giving the log files of the ISM Web server. The authors are grateful to Prof. Amitava Dutta (HOD, Computer and Statistical Service Centre), Mr. Subhashis Kumar Pal (Associate Scientist), Mr. Ashish Biswas (Scientific Assistant) for providing the log files of ISI Kolkata Web server. In addition, the authors are thankful to Indian School of Mines, Dhanbad, India, for providing necessary facility. The authors are thankful to the reviewer for his valuable suggestions to improve the quality of the paper.

References

1. F. D. Maio and E. Zio, Failure prognostics by a data-driven similarity-based approach, *Int. J. Reliab. Qual. Saf. Eng.* **20** (2013), doi:10.1142/S0218539313500010.

2. H. Pham, *System Software Reliability* (Springer-Verlag, London, 2006).
3. J. D. Musa, A. Iannino and K. Okumoto, *Software Reliability Measurement, Prediction, Application* (Int. Ed. McGraw-Hill, 1987).
4. J. Tian, Better reliability assessment and prediction through data clustering, *IEEE Trans. Softw. Eng.* **28** (2002) 997–1007.
5. P. C. Sam and S. Chakraborty, Possibilistic safety assessment of hybrid uncertain systems, *Int. J. Reliab. Qual. Saf. Eng.* **20** (2013) 135002.
6. S. Chatterjee and A. Roy, Web software fault prediction under fuzzy environment using MODULO-M multivariate overlapping fuzzy clustering algorithm and newly proposed revised prediction algorithm, *Appl. Soft Comput.* **22** (2014) 372–396.
7. L. A. Walls and A. Bendell, Time series methods in reliability, *Reliab. Eng. Syst. Saf.* **18** (1987) 239–265.
8. M. R. Lyu, *Handbook of Software Reliability Engineering* (IEEE Computer Society Press, McGraw Hill, New York, 1996).
9. M. Xie, *Software Reliability Modeling* (World Scientific Press, London, 1991).
10. S. Chatterjee, S. Nigan, J. B. Sing and L. N. Upadhayaya, Transfer function modelling in software reliability, *Computing* **92** (2011) 33–48.
11. S. Chatterjee, R. B. Misra and S. S. Alam, Prediction of software reliability using an auto regressive process, *Int. J. Syst. Sci.* **28** (1997) 391–396.
12. S. Chatterjee, J. B. Singh and A. Roy, A structure-based software reliability allocation using fuzzy analytic hierarchy process, *Int. J. Syst. Sci.* doi:10.1080/00207721.2013.791001.
13. C. Kallepalli and J. Tian, Measuring and modeling usage and reliability for statistical Web testing, *IEEE Trans. Soft. Eng.* **27** (2001) 1023–1036.
14. J. Offutt, Quality attributes of web software applications, *IEEE Softw.* **19** (2002) 25–32.
15. J. Tian, S. Rudraraju and Z. Li, Evaluating Web software reliability based on workload and failure data extracted from server logs, *IEEE Trans. Softw. Eng.* **30** (2004) 754–769.
16. K. S. Popstojanova, A. D. Singh, S. Mazimdar and F. Li, Empirical characterization of session-based workload and reliability for web servers, *Empire Soft. Eng.* **11** (2006) 71–117.
17. L. Ma and J. Tian, Web testing for reliability improvement **80** (1996) 795–804.
18. M. F. Arlitt and C. L. Williamson, Web server workload characterization: The search for invariants, *IEEE/ACM Trans. Netw.* **5** (1997) 631–645.
19. T. Huynh and J. Miller, An empirical investigation into open source web applications' implementation vulnerabilities, *Empirical Softw. Eng.* **15** (2009) 556–576.
20. L. D. Catledge and J. E. Pitkow, Characterizing browsing strategies in the World-Wide Web. Computer Networks and ISDN Systems, **27**(6) (1995) 1065–1073.
21. H. R. Shumway and S. D. Stoffer, *Time Series Analysis and Its Applications* (Springer, 2008).
22. I. T. Jolliffee, *Principal Component Analysis* (Springer, New York, 1986).
23. N. D. Singpurwalla and R. Soyer, Assessing (software) reliability growth using a random coefficient autoregressive process and its ramifications, *IEEE Trans. Softw. Eng.* **11** (1985) 1456–1464.
24. S. Chatterjee and A. Roy, Novel algorithms for web software fault prediction, *Qual. Reliab. Eng. Int.*, doi:10.1002/qre.1687.
25. G. P. E. Box and G. M. Jenkins, *Time Series Analysis, Forecasting, and Control* (Holden-Day, San Francisco, 1976).

26. P. W. Lai, *Transfer Function Modeling Relationship Between Time Series Variables* (Mid Anglia Litho, 1979).
27. V. Anselmo and L. Ubertini, Transfer function noise model applied to flow forecasting, *Hydrol. Sci.* **24** (1979) 353–359.
28. H. Akaike, A new look at the statistical model identification, *IEEE Trans. Autom. Control.* **19** (1974) 716–723.

About the Authors

Subhashis Chatterjee obtained his B.Sc. (Mathematics) from TDB College, Raniganj, The University of Burdwan, India. He obtained M.Sc. (Mathematics) and Ph.D. from IIT Kharagpur, India. His area of research is software reliability modeling. Presently Dr. Chatterjee is working as Associate Professor, in the Dept. of Applied Mathematics, Indian School of Mines (ISM), Dhanbad, India. He has served GIET, Orissa and SMIT, Sikkim, India, as a faculty. He is a member of IEEE Reliability Society. He has quite a good number of international and national publications. He is reviewer of various international journals. His areas of interest are software reliability, web software reliability, or, stochastic process and fuzzy set.

Arunava Roy has submitted his Ph.D. thesis from the Department of Applied Mathematics, Indian School of Mines, Dhanbad. Presently he is pursuing Post Doctoral Work in University of Memphis, USA. He did his M.Sc. in Mathematics and Computing from Indian School of Mines Dhanbad in 2010. He was a Gold medalist in M.Sc. He got inspire fellowship provided by the Govt. of India. His areas of interest are web software reliability, cyber security, algorithm design and analysis, data structure and programming. He has published a number of papers in different international journals of reputed publication houses, like: Wiley, Elsevier, Springer and Taylor & Francis. He is the reviewer of a number of international journals.