Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2026 (44) 2** : 1–14.
ISSN-0037-8712
doi:10.5269/bspm.80528

Prediction of Self-similar Behavior Internet user's Traffic Data through ANN and Multilinear Regression Models

Pravalika Dhadi, Archana Reddy R., Pushpalatha Sarla* and Ramesh Alluguvelli

ABSTRACT: One of the major challenges in network management is effectively managing and scheduling the arrival patterns of internet users across various web centers. In this study, the Hurst method is employed to analyse the self-similarity of online user traffic, helping to identify performance degradation in page loading during periods of high traffic. A linear regression model is utilized to fit the observed data and to formulate a mathematical equation for forecasting user arrival patterns. Furthermore, an Artificial Neural Network (ANN) with a multilayer perceptron architecture is implemented to predict online user behavior more accurately. The study also evaluates and compares the prediction accuracies of both models. Traditional web traffic prediction models, such as Poisson and Markov-based approaches, fail to adequately capture the self-similar and bursty nature of internet user arrivals. Moreover, most existing studies focus primarily on traffic characterization rather than predictive modeling, with limited comparative analysis between statistical and machine learning approaches.

Keywords: Self-similarity, Hurst exponent method, multi linear regression, ANN.

Contents

1. Introduction

Due to its high level of complexity, calculate similarity mathematically is difficult to represent and explain. Hydrology was the field in which the Hurst exponent estimation was first introduced. The Hurst [1] parameter is fascinating since it has applications in many branches of mathematics, such as wavelets, fractals, autocorrelation, etc. The Hurst index gives a general indication of whether a time series has a long-term dependency. This facilitated studies in network traffic analysis and modelling. Although Mandelbrot was not the first to think of the ideal dam, the reference provides a clear numerical formulation. Clearly, this elegant numerical explanation excludes problems like error margins, evaporation losses, etc., but the idea is still valid. In fact, as Hurst [2] noted, "Increased storage losses are ignored because the location is unsuitable for long-term storage, unless they are negligible". For analysing

* Corresponding author.
2020 *Mathematics Subject Classification:* 62M10.
Submitted December 01, 2025. Published March 14, 2026

inconsistencies in a time series, Benoit Mandelbrot's fractal theory can be used [3]. A formation with a advanced FD is additional complex or uneven than one with a lower capacity and occupies spare area [4]. Fractal theory is a concept that explains the dimensions of non-integer numbers, which are called Fractal Dimension (FD), as the measure of the complexity of the shape. The self-similarity is calculated using the Hurst parameter. We have types of methods on the side of calculating The Hurst exponent, and the outcomes from each method are very diverse. Correlogram, R/S, Periodogram, Percentile, and other approaches are a few examples. fitting a Linear regression model and its applications for the self-similar nature of web traffic data [5]. Managing and scheduling internet user arrivals at web centers is a significant challenge due to the self-similar and bursty nature of web traffic. Traditional Poisson and Markov-based models fail to capture these characteristics, leading to inaccurate performance predictions. Previous studies have shown that internet traffic exhibits long-range dependence, which can be analyzed using the Hurst exponent [6,7] While linear regression models have been used to fit self-similar traffic data [8] and neural networks have been applied for traffic classification [9] limited research has compared their predictive capabilities [10]. To address this gap, the present study introduces a hybrid framework that integrates Hurst-based self-similarity analysis with both linear regression and ANN models to enhance the accuracy and adaptability of web traffic prediction.

1.1. Artificial Neural Networks

The Artificial Neural Networks are computer programs that mimic functions of the brain, typically derived from organic sensory systems. They provide a lot of advantages over traditional modeling techniques among all statistical models built on the artificial neural networks. Artificial Neural Networks have established their worth by offering substantial advantages, especially in managing intricate data and resolving issues that pose challenges for conventional techniques [11]. These are autonomous self-learning systems capable of improving their performance and decision-making without the need for explicit programming or manual intervention [12]. ANN is essential for measuring surface roughness. To determine how well ANN models the problem, several previous studies are reviewed [13]. The development of a Back-Propagation Neural Network (BPNN) was designed to address the issue of non-linear properties in network traffic data [14]. The Back Propagation Neural Network is also a feed-forward neural network used in MLP planning with supervised learning. Its construction uses function approximation theory. After Paul Werbos introduced the BPNN method in 1974, David Parker revived it in 1982, and Rumelhart and McCelland popularized it in 1986. BP method works by sending a train pattern input to the hidden layer and then to the outcome layer. The Outcome layer responds with network outcome. When a network's outcome deviates from its intended outcome, the hidden layer's outcome should be called backward and directed towards the input layer's neurons [15]. Artificial neural network (ANN) modeling yields superior analytical outcomes and works well for forecasting [16]. Limitations of ANN: Storing information on the entire network: traditional programming is not stored on the database but it stores in entire network. The network can still operate properly if certain elements in one area disappear. Having fault tolerance: ANN will outcome the data even if one or many cells of it are impaired. This is how the networks are fault tolerant. Machine learning capability: The neural networks are smart enough to learn and think systematically by suggesting related events. Parallel processing ability: ANNs function by using massive parallelism which enables them to execute many tasks at a time. Artificial neural networks are in this class. Choosing a suitable network structure: It is impossible to have an artificial neural network made up of a certain set of rules. One has to go through the stages of trial and error to find a perfect network structure. Difficulty in communicating the issue to the network: Artificial Neural Networks may only process numerical data. Prior to using ANN issues need to turn into mathematical values. The network's lifespan is unknown: When the network is reduced to a specific value of the sample error, training is said to have finished. We don't get the best outcomes with this number. The ANN model was effectively defined through input and outcome data. Among others, the internet traffic forecasting problem's problem (turning data into numbers) was solved. ANN, particularly the Multilayer Perceptron model copes with basic challenges such as the adaptation to complex non-linear data and the problem formulation.

1.2. Multi-Linear Regression

Multiple linear regression is a statistical model type which is used to determine the combined effect of several independent variables on one dependent variable. It represents a generalized regression modeling that covers both linear and logistic variations, consequently making it easier to make predictions created on input data. An integration of traditional statistical regression methods with artificial intelligence (AI) and machine-learning (ML) techniques has become increasingly prominent across diverse application domains. compared MLR and multiple ML methods (neural networks, SVM) to predict soil compaction and shear stress from electrical parameters, confirming that ANN models outperformed MLR by a substantial margin [17]. integrating ANN with appropriate feature-selection methods yielded superior predictive performance compared to penalized linear models, underscoring the advantage of nonlinear AI-based regression frameworks in economic forecasting tasks [18]. The M/G/K queuing model to fuel station congestion and contrasted Runge–Kutta (RK4) solutions with predictions from Artificial Neural Networks (ANN), proving that ANN can accurately simulate conventional queuing dynamics and be a useful tool for service operations optimization [19]. The literature review indicates that Artificial Neural Networks and Multiple Linear Regression are not yet extensively employed in predicting web users, as their applications remain relatively in this field of study. To the authors knowledge, only a limited no. of studies has utilized ANN and MLR to predict internet traffic.

2. Methods and Data Description

This study investigates the nature of web user's data using self-similarity methods and predicts the internet traffic to be able to expect busy traffic, consuming the Artificial neural networks and Multiple linear regression. Table 1 represents the data of four variables, the page loads, the unique visits, first-time visits, and returning visits were collected for the year 2020 from the Kaggle database https://www.kaggle.com/datasets/bobnau/daily-website-visitors..Regression.

Table 1: website visitors data used for prediction

Date	Page loads	First time visits	Unique visits	Returning visits
01-01-2020	1554	1105	870	235
01-02-2020	2820	2083	1754	329
01-03-2020	2970	2180	1859	321
01-04-2020	2111	1526	1300	226
01-05-2020	2393	1788	1514	274
01-06-2020	3704	2770	2341	429
01-07-2020	3760	2915	2492	423
01-08-2020	3698	2844	2402	442
01-09-2020	4243	3374	2910	464
01-10-2020	3177	2415	2043	372
–	–	–	–	–
–	–	–	–	–

2.1. Hurst Exponent Methods

The Hurst coefficient (H) is the standard parameter employed to measure the level of self-similarity. The growth of Hurst was before 1951, H.E. Hurst [1] and his team dedicated numerous years to analyzing optimal dam sizing for storing water and assessing the drought conditions of the Nile River. In the financial market, the Hurst parameter is used to ascertain whether to trade refuges. It is also used in ecology to control population growth and decline. India's CPI headline inflation to identify self-similarity and long-range dependence using various Hurst index estimation methods [20]. The parameter, which measures self-similarity, has a range of $0.5 < H < 1$. Let $X = \{X_t : t = 1, 2, ..\}$ represents the second-order stochastic process [21]. This process is characterised by a constant mean $\mu(\forall\, t)$, constant Variance

$\sigma^2(\forall\ t)$ and ACF $\gamma\left(s\right)$ with lag 's' i.e.

$$\gamma\left(s\right) = \frac{Cov(X_t, X_{t+s})}{Var(X_t)},\ s \geq 0 \tag{2.1}$$

Then the aggregating process, $X_t^{(p)}$ is calculated by the initial process X_t as

$$X_t^{(p)} = \frac{1}{p}\sum_{i=1}^{p} X_{(t-1)p+i}\ ,\ t = 1, 2, \ldots \tag{2.2}$$

In this p is an integer (≥ 1) that represents the size of the slice for the averaging. The ACF of $X_t^{(p)}$ be specified as $\gamma^{(p)}\left(s\right)$since it is likewise a second-order stationary process. The stochastic procedure 'X' is characterized as a second-order self-similar with variance $\sigma^2$and H is the Hurst index if

$$\gamma\left(s\right) = \frac{\sigma^2}{2}\left[\left(s+1\right)^{2H} - 2s^H + \left(s+1\right)^{2H}\right],\ \forall s \geq 1 \tag{2.3}$$

In the stochastic development 'X 'is characterized as asymptotically second-order self-similar with variance σ^2 and the Hurst index H if

$$\sum_{m\to\infty} \gamma^{(p)}(s) = \frac{\sigma^2}{2}\left[\left(s+1\right)^{2H} - 2s^H + \left(s+1\right)^{2H}\right],\ \forall s \geq 1 \tag{2.4}$$

In variance-time series analysis, 'X' is characterized as precisely second-order self-similar with the variance σ^2 and the Hurst index $H = 1 - \frac{\beta}{2}$ if

$$Var\left(X^{(p)}\right) = \sigma^2 p^{-\beta}, \forall\ p \geq 1 \tag{2.5}$$

From Eq. 2.5, it follows that for $H \neq 0.5$,

$$\gamma\left(s\right) = H\left(2H-1\right)p^{2H-2} as\ s \to \infty \tag{2.6}$$

And thus,

$$\sum_p \gamma(p)\ c \sum_p\ s^{-\beta}, c = H\left(2H-1\right). \tag{2.7}$$

The series $c\sum_p\ s^{-\beta}$is divergent if$0.5 < H < 1$ or $0 < \beta < 1$ then they are convergent, comprising a series of positive terms. Therefore, for $0.5 < H < 1$, the autocorrelation function (ACF) exhibits hyperbolic decay, categorizing the random process X as Long-Range Dependent. And for $0 < H < 0.5$, $\sum_p \gamma(p)$ is finite, indicating that the random process X is categorized as Short-Range Dependent.

2.1.1. R/S Method. R/S (Rescaled range) Analysis is a commonly used method in studying the fractal properties of time series. Fractal analysis utilizes this non-parametric statistical principle to calculate the Hurst exponent, which refers to a measure of the variable nature of the statistical population. The Hurst exponent is determined by the ratio of the range of values in a subseries and the standard deviation of the same subseries. Clearly, a series length N is partitioned into d time series of length n, so that $d * n = N$. For each period of sub-length n, a rescaled range is determined. This technique aids in understanding the persistence and memory of the time series. The time series $X = (x_1, x_2,\ x_3 \ldots\ldots x_n)$ is thought about as the set of observations Calculation of the mean value

$$m = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{2.8}$$

1. Calculation of the adjusted series Y

$$Y_t = X_t - m, t = 1, 2 \ldots n \tag{2.9}$$

2. Calculation of the cumulative deviation

$$Z_t = \sum_{i=1}^{t} Y_i, t = 1, 2...n \tag{2.10}$$

3. Design of the range series -R

$$R(n) = max\,(Z_1, Z_2,\ Z_3 \ldots\ldots Z_n) - \min\,(Z_1, Z_2,\ Z_3 \ldots\ldots Z_n) \tag{2.11}$$

4. Calculation of the standard deviation series -S

$$s\,(n) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - m)2} \tag{2.12}$$

where m denotes the mean value computed at point 1.

5. Computation of the Rescaled range series $\frac{R}{S}$

$$(R/S)_n = \frac{R(n)}{S(n)} \tag{2.13}$$

Finally, the Hurst Index is obtained by the formula

$$H = \frac{log\left(\frac{R}{S}\right)}{\log\,(N)} \tag{2.14}$$

The graph of log (N) against log (R/S) provides an estimate of the Hurst exponent, represented as the slope of the resulting curve.

2.1.2. Correlogram Method. The Correlogram approach, where the predicted relation can be expressed in words of auto-covariance function, is used to approximate the Hurst parameter H in time series analysis. Hurst parameter H, which is the factor of the approximation of the log autocorrelation function vs the log of frequencies in this analysis, is connected to the autocorrelation function by the slope. The autocorrelation function of the series should be determined until the autocorrelation function (ACF) is negative, all positive values of the data series should be used, and a degeneration line should be performed concerning the log of the ACF values vs the general log of the lags of the ACF values. For calculating the Hurst exponent H, by

$$H = 1 + \propto /2 \tag{2.15}$$

Where $\propto$ is the regression's slope. Mandelbrot [3] has some lesser-known sample correlation issues. Again, possessing a substantial quantity of data points is essential. The range of the data need not be a power of 2, unlike the rescaled range technique. Compared to the rescaled range approach, the ACF method is simpler to calculate. Constant variance and stationarity are presumptions made for the time series in the correlation method. Additionally, when examining long-range memory models, where the long-time scales are what we want to explore, the inaccuracy is equally substantial for big-time lags. Additionally, these errors may result in some negative covariance that cannot be studied in a logarithmic plot.

2.2. Back Propagation Neural Network

The principal actions of a feedforward neural network mostly occur in three steps namely the input, the propagation and the outcome. The input layer receives an information in the form of data given as input. Each of the input-nodes is tied to a unique aspect of the data. Every link in the output and the hidden layer figures the weighted sum and bias. Each neuron adds some weights to the product of its weights and inputs, and then add a bias value. Based on the BPNN model, creating a prediction system requires four processes which included the data gathering process, sample data procurement and training, data standardization, training optimization and testing ANNs and the effective use of data for training and testing is critical for increasing model performance. Backpropagation is a common approach in feedforward neural networks that uses the gradient descent algorithm to reduce errors. The procedure consists of two primary phases: forward propagation and backward propagation, which are performed iteratively over a set number of epochs until the model converges.

Terminology of the process:
x: the input vector has length n.
y: the outcome vector has length m.
σ: simulation function.
X_{ji}: i^{th} input to unit j.

Finding the ideal weights and biases are used to diminish the loss function is necessary in order to train the FFNN. The variance between the intended goals and the anticipated outcomes is measured by this loss function. We can symmetrically build, train, and optimize this model to attain high classification task accuracy by utilizing the FFNN formulas. The problem type determines the output layer activation function. Let $x_1, x_2, \ldots, x_n$ denotes the n input values, each with weights $w_1, w_2, \ldots, w_n$ After being multiplied by their corresponding weights, the input values are added up.

$$\theta = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n = \sum_{i=1}^{n} w_i x_i \tag{2.16}$$

The weighted sum produces a function $y = f(\theta)$. The output neuron y depends on the weighted sum $y = f(\theta)$. This function is known as stimulating function. In case of linear function, apply the bias as in Eq. 2.16,

$$f(\theta) = a + \theta = a + \sum_{i=1}^{n} w_i x_i \tag{2.17}$$

A neuron transforms into a linear model with the bias 'a' existing the point of contact with the weights, $w_1, w_2, \ldots, w_n$, existence the slopes.

$$\sigma(\theta) = \frac{1}{1 + e^{-(w_1 x_1 + w_2 x_2 + \ldots + w_n x_n)}} \tag{2.18}$$

This is essential instruction for determining the input and outcome of each link. The stimulating function:

$$\sigma(\theta) = \frac{1}{1 + e^{-\theta}} \tag{2.19}$$

$$\sigma(\theta) = \frac{1}{1 + e^{-(w_1 x_1 + w_2 x_2 + \ldots + w_n x_n)}} \tag{2.20}$$

For errors: L=The Mean square error

$$L = \frac{1}{2}\left[\sum_{i=1}^{n} \left(y_{actual\ values} - y_{target\ values}\right)^2\right] \tag{2.21}$$

With a backpropagation neural network, the weighted parameters are updated whenever an error is detected in the network's output. The stochastic gradient descent (SGD) algorithm is employed to formulate the weight update rule of the backpropagation process. This backward propagation of error information enables the network to iteratively refine its connection weights through the gradient descent mechanism, thereby minimizing the loss function and progressively improving the model's predictive accuracy throughout the training phase.

2.3. Multi-Linear Regression Model

Among the groups under supervised learning, MLR, may produce a model for establishing the relationship between two or more interpretative factors and response variables in the practical data by using the necessary linear equations.

$$Y_i = b_0 + b_1 X_{i1} + b_2 X_{i2} + \ldots + b_k X_{ik} + \varepsilon_i \tag{2.22}$$

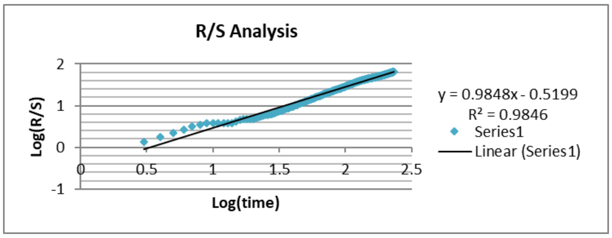

Figure 1: Rescaled range analysis of web users' data

Where, Y_i represents the dependent variable; b_0 denotes a constant; X_{ik} signifies an independent variable; b_k constitutes the regression coefficient vector(slope); and ε_i indicates random errors.

3. Numerical Results and Discussion

3.1. Self-similarity (Hurst-index) by R/S Method

figure 1 shows a graphic representation of the rescaled range analysis for the data of the internet user. In this graph, time is represented on the X-axis in minutes and the Y-axis is the web user's data the trend line $Y = 0.9848X - 0.5199$ with the slope $\propto$ is 0.9848 shows the Hurst index value $H = 0.9848(0.5 < H < 1)$, reveal the Extended Dependency or Self-similar characteristics; R square value represents the high degree of similarity. Table 2 depicts the determined R/S values for each web user as well as the logarithmic values of R/S values, these were calculated using the SPSS statistical tool.

Table 2: R/S values of Internet user's data

Web users	Log (Time)	Log(R/S)
1554	0.47712125	0.12703092
2820	0.60205999	0.23916442
2970	0.69897	0.33837034
2111	0.77815125	0.42443293
2393	0.84509804	0.492491
3704	0.90308999	0.54608918
3760	0.95424251	0.57334145
3698	1	0.58542416
4243	1.04139269	0.58213577
3177	1.07918125	0.57409409
——-	——-	——-

3.2. Hurst Index Using Correlogram Method

The Correlogram (i.e ACFv/sLag) describing the ACF represented the data is significant and thus figure 2 and 3 is the plot of the autocorrelation function of Web user's data, in this graph each bar denotes the direction and magnitude of the correlation, bars extending beyond the line indicate statistical significance, meaning that there should be a significant correlation between the data.

The Lag 1-lag16 all are significant and reported in table 3. the fundamental process, is predicated on the asymptotic chi-square approximation, is considered to be independent (white noise). The statistical significance level was considered at 0.05. By the ACF method which was estimated from Eq. 2.5, the Hurst exponent (H) value was found to be 0.99. The Ljung-Box test is employed to assess serial correlation in time series data. In table 3 the "Autocorrelation" column shows the dataset's correlation with itself over lags, showing how closely data at one period is related to data at another. The autocorrelation standard error is in the Std. Error column. The Box-Ljung Statistic column lists the test statistic values used to test the null hypothesis that data are independently distributed. Table 4 represents Time steps

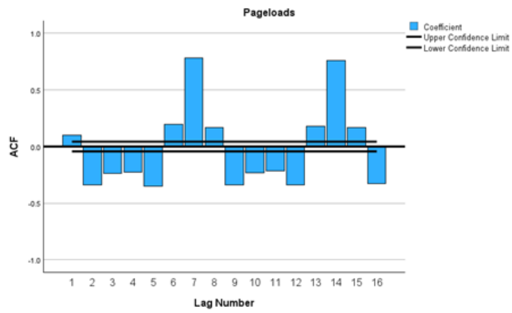

Figure 2: Correlogram graphs for the web user's data

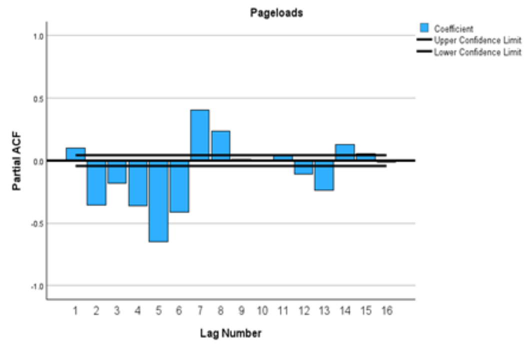

Figure 3: Lag number Vs. PACF

between data points are shown in the Lag column. The partial autocorrelation column controls for the time series values at all smaller lags to illustrate its correlation through its own lagged values. That values are -0.649 to 0.127. The Std. Error column shows that all standard errors are 0.021, indicating partial autocorrelation estimate variability. Table 4 shows how past time series values connect to current values when other past values are taken into account. It is essential for time-series analysis pattern identification and predictive modeling.

Table 3: Statistics of autocorrelations of Internet users

Lag	Autocorrelation	Std Error[a]	Statistical value	df	Std Error[b]
1	0.1	0.021	21.637	1	<.001
2	-0.341	0.021	274.382	2	<.001
3	-0.24	0.021	399.195	3	<.001
4	-0.225	0.021	509.453	4	<.001
5	-0.348	0.021	772.106	5	<.001
6	0.196	0.021	856.025	6	<.001
7	0.786	0.021	2189.469	7	<.001
8	0.166	0.021	2249.454	8	<.001
9	-0.341	0.021	2502.664	9	<.001
10	-0.231	0.021	2618.666	10	<.001
11	-0.213	0.021	2717.452	11	<.001
12	-0.338	0.021	2966.969	12	<.001
13	0.178	0.021	3035.75	13	<.001
14	0.757	0.021	4286.739	14	<.001
15	0.167	0.021	4347.969	15	<.001
16	-0.33	0.021	4585.588	16	<.001

[a] The original process presumed to be independent (white noise)
[b] Based on the asymptotic chi-square approximation

According to figure 4 The log autocorrelation plot shows the line with a slope of -0.0028, which helps estimate the Hurst parameter. The lag values are represented on the x-axis, while the autocorrelation function is designed on the y-axis. log autocorrelation plot of figure 4 Depicts the value of Hurst Index calculated from $H = 1+\alpha/2$ where α is the slope of log autocorrelation plot, here $H = 1+(-0.0028/2) = 0.9986$.

3.3. Prediction of Web Users Using Artificial Neural Network

Table 5 represents total of 233 samples. Training of 164 samples, representing 70.7% of the total. Testing of 43 samples, representing 18.5% of the total. Holding out 25 samples, representing 10.8% of the total.232 samples are valid representing 100.0% of the total (this seems to be a sum of the training,

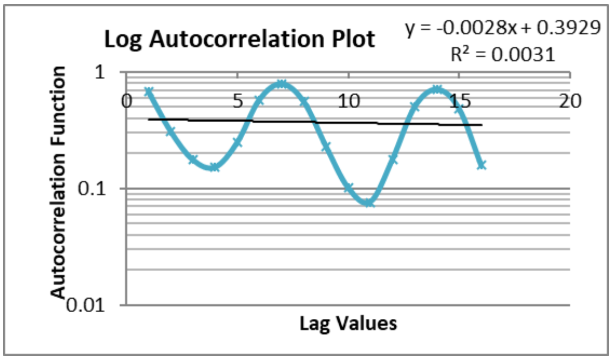

Figure 4: Autocorrelation plot

Table 4: PACF Statistics of Internet users

Lag	Partial Auto-correlation	Std. Error
1	0.1	0.021
2	-0.355	0.021
3	-0.183	0.021
4	-0.362	0.021
5	-0.649	0.021
6	-0.412	0.021
7	0.405	0.021
8	0.235	0.021
9	0.008	0.021
10	-0.001	0.021
11	-0.044	0.021
12	-0.106	0.021
13	-0.236	0.021
14	0.127	0.021
15	0.057	0.021
16	-0.14	0.021

Table 5: Summary of the Case Processing

		N	Percent
Sample	Training	164	70.7%
	Testing	43	18.5%
	Holdout	25	10.8%
Valid		232	100.0%
Excluded		1	
Total		233	

testing, and holdout sets). Excluded only one sample.

figure 5 illustrates an artificial neural network configuration with an input layer, two hidden layers, and an outcome layer. The input layer comprises three nodes: Unique visits, First-time Visits, and Returning visits. During training, the network learns synaptic weights to connect these nodes to two hidden nodes, H (1:1) and H (1:2). Each hidden node receives bias node input to alter activation function outcome. The hidden layers use the hyperbolic tangent (tanh) activation function to make the model non-linear and capture complicated data patterns. The outcome layer, with one node named "Page loads," outcomes the value without alteration using the identity activation function. For website traffic analysis and performance improvement, this architecture predicts page loads based on website visit kinds.

From table 6 the dependent variable is "Page Loads." These metrics provide insights into the model's performance during different phases of its evaluation. The sum of squares of the error and the relative error are common metrics used to evaluate accuracy and performance of regression models. The stopping rule indicates that the training stopped when there was no further decrease in error. Table 7 is represents the weights that are crucial for the neural network's predictions. During the forward pass, inputs are multiplied by these weights, and the resulting values are passed through activation functions to produce the final predictions. Table 8 represents the unique variable with utmost significance, exhibiting a normalized value of 100.0%. It seems to be the most influential factor in predicting the dependent

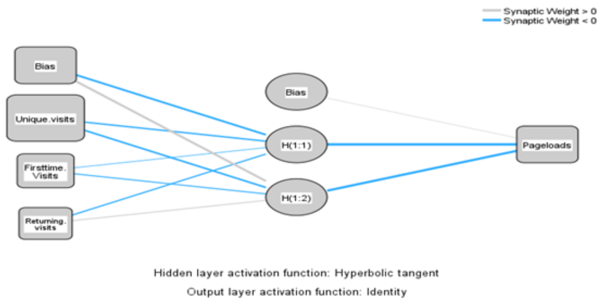

Figure 5: Activation function graph

Table 6: Model Summary

Training	SSE	.715
	Relative Error	.009
	Stopping Rule Used	1-consecutive step(s) with no decrease in error
	Training Time	0:00:00.01
Testing	SSE	.214
	Relative Error	.014
Holdout	Relative Error	.009
The Dependent Variable: Page Loads		
a. Error calculations are based on the testing sample.		

Table 7: Parameter Estimates

Predictor		Predicted		
		Hidden Layer 1		Outcome Layer
		H(1:1)	H(1:2)	Page loads
The Input Layer	(Bias)	-.418	.456	
	Unique visits	-.219	-.312	
	First-time Visits	-.027	-.161	
	Returning visits	-.171	.045	
Hidden Layer 1	(Bias)			.011
	H(1:1)			-1.576
	H(1:2)			-1.484

variable. First-time visits still contribute significantly, with a normalized importance of 36.0%. returning visits have the lowest importance among the three but still contribute to the model with a normalized importance of 24.7%.

Table 8: Normalized importance of Internet visits

	Importance	**Normalized Importance**
Unique Visits	.622	100.0%
First-time Visits	.224	36.0%
Returning Visits	.153	24.7%

figure 6 shows the graph where page loads and predicted values are plotted. Thus, it can be inferred that the page loads and proposed value are getting more congruent. As a result, this data can greatly be improved by the website visitor behavior such as engagement and conversion rates. The higher number of page loads means that the page reliability factor is moving in a positive direction.

figure 7 depicts the differences in our data between observed values and estimated values. The dependent variable is page loads. The vertical axis shows a range of residuals from -400 to 200, while the horizontal axis shows expected values from 0 to 7000. The data points appear to be distributed around a horizontal line at zero, which implies that the observed and expected values are the same. This graph

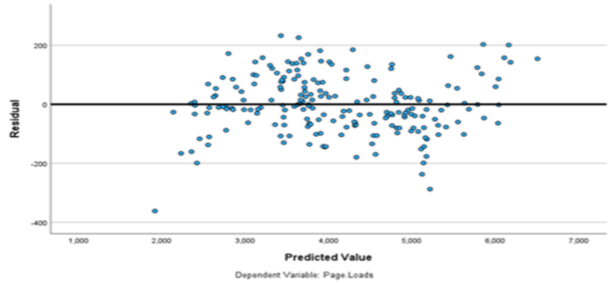

Figure 6: Website visitor's prediction graph

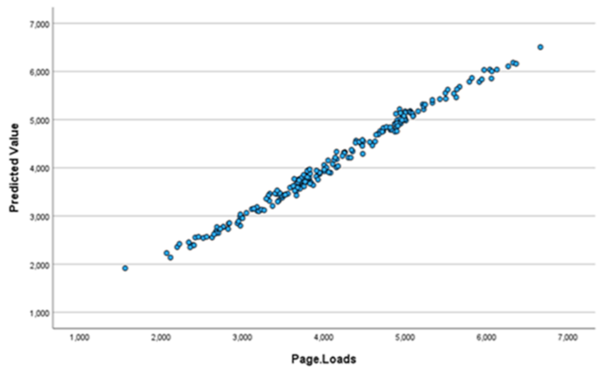

Figure 7: Residual Graph

displays the residual patterns that could point out model fit problems, which in turn helps to ascertain if the linear regression model has sufficiently represented the relationship between variables.

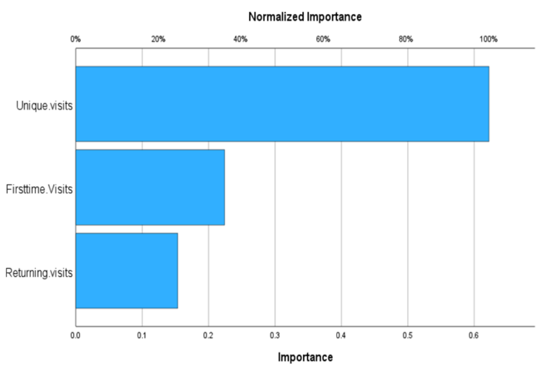

Figure 8: Normalized importance of website visitors

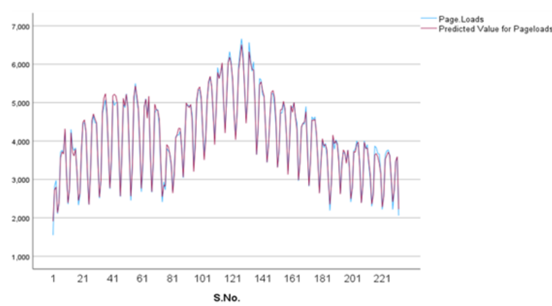

Figure 9: Prediction of page loads

The figure 8 exhibits the normalized relevancy of differing types of website visits. It should be noted that the unique visits are the most crucial factor with more than 80% of weight meaning that the main condition for a site's success is to attract internet users. The unique visits are the predominant factor in site growth at the same time, accounting for between 20% and 30% of the total. This means that getting existing visitors is still a very crucial task. The returning ones are the least valuable, with a score of less than 10%, which is to say that the reuse of humans is not so much as the acquisition of new and unique visitors.

Data on the real page loads shows peaks and troughs on the Page Loads line. The model's Predicted

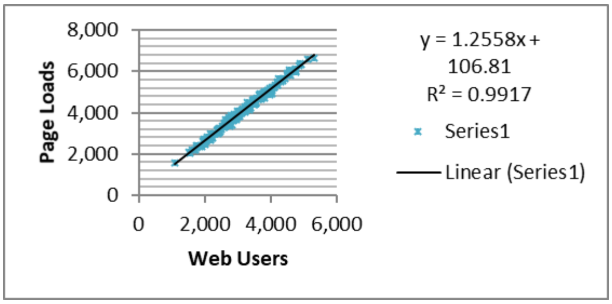

Figure 10: Linear regression graph

Values for Page Loads line has less volatility than real page loads. The predicted lines and the actual lines match, which means the prediction model is good at capturing data patterns. By employing this figure 9, we can perform a predictive model examination of their compatibility with actual data.

3.4. Linear Regression Method to Predict Web Users

figure 10 shows a linear relationship between web users and page loads. The trend is indicated by a black labelled line that fits these data points best. The equation of this line is ($y = 1.2558x + 106.81$), where y is page loads and x is web users. The (R^2) value is 0.9917 indicates a strong connection between web users and page loads, indicating a predictable rise in page loads with increased online users.

Table 9: Descriptive statistics

Measures	**Descriptive Statistics**
Mean	3173.839827
Median	3047
Mode	3982
Standard Deviation	798.0712485
Sample Variance	636917.7177
Range	3796
Minimum	1526
Maximum	5322
Skewness	0.158585162
Kurtosis	0.519014894
Count	231
Sum	733157

Table 10: F-test two-sample for variance

Mean	3173.839827	4092.376623
Variance	636917.7177	1014430.079
Observations	231	231
	230	230
F	0.627857682	
P(F¡=f) one-tail	0.000224181	
F Critical one-tail	0.804635469	

Table 9 represents the descriptive statistics of the data set, with an average value of 3173.839827, indicating the central tendency. The estimate of the standard deviation of the sample mean (standard error) is 52.50923307, the median of the ordered data set is 3047, while the standard deviation and variance are 798.07, 636917.712 indicates the measure of the amount of dispersion or spread of the data set. These statistics collectively provide a summary of the central tendency, spread, and shape of the dataset. The small p-value (0.000224181) in table 10 indicates a significant disparity in variances between the two groups. In this case, since the calculated F=0.627 is less than the critical F=0.8046, indicates rejection of the null hypothesis.

4. Conclusion

In this paper, real-time internet user data arrivals at web centres are shown to be self-similar. R/S and the correlogram methods have been used to investigate self-similarity. This results in performance decay in network traffic. Also, multi-linear perceptron in ANN is to be best fitted for the data to forecast the web users. Also, an artificial neural network (ANN) model, utilizing a multi-layer perceptron (MLP) and a multi-linear regression model, is employed to analyze internet traffic. We utilized the input and outcome data to characterize ANN model and examined the efficacy of the training algorithms employed to estimate the weights of the neurons, comparing several training procedures. Consequently, the obtained model using ANN successfully be used as an adequate model for the identification and management of internet traffic. Multi Linear perceptron in ANN is to be best fitted for the data to forecast the web users. This serves as an effective foundational instrument for managing internet traffic at various intervals; all implementations were executed utilizing the neural network toolbox within the MATLAB software package. Consequently, this work employed MATLAB version R2019b on a 64-bit PC operating Windows 10 to execute the MLR and ANN models.

References

1. Hurst, H. E. (1951). Long-term storage of reservoirs: An experimental study. Transactions of the American Society of Civil Engineers, 116, 770–799.
2. Hurst, H. E. (1951). Long-term storage capacity of reservoirs (With discussion). Transactions of the American Society of Civil Engineers, 116, 770–799.
3. Mandelbrot, B. (1967). How long is the coast of Britain? Statistical self-similarity and fractional dimension science, 156(3775), https://doi.org/10.1126/science.156.3775.636
4. Breslin, M. C., & Belward, J. A. (1999). Fractal dimensions for rainfall time series. Mathematics and Computers in Simulation, 48(4–5), 437–446.https://doi.org/10.1016/S0378-4754(98)00225-5
5. Pushpalatha Sarla, D., & Mallikarjuna Reddy, D. (2017). Linear regression model fitting and implication to self-similar behavior traffic arrival data pattern at web centers. IOSR Journal of Computer Engineering (IOSR-JCE), 19(1), 1–5.
6. Fontugne, R., Abry, P., Fukuda, K., Veitch, D., Cho, K., Borgnat, P., & Wendt, H. (2017). Scaling in Internet Traffic: A 14-Year and 3-Day Longitudinal Study, with Multiscale Analyses and Random Projections. arXiv preprint arXiv:1703.02005.
7. Millán, G. (2021). Estimation of Hurst Exponent in Self-Similar Traffic Flows. RIAI - Revista Iberoamericana de Automática e Informática Industrial.
8. Reddy, D. M., & Sarla, P. (2017). Linear Regression Model Fitting and Implication to Self-Similar Behaviour Traffic Arrival Data Pattern at Web Centres. IOSR Journal of Computer Engineering, 19(1), 01–05.
9. Filus, K., Domański, A., Domańska, J., & Szyguła, J. (2020). Long-Range Dependent Traffic Classification with Convolutional Neural Networks Based on Hurst Exponent Analysis. Entropy, 22(10), 1159.
10. Shi, J., Leau, Y. B., Li, K., & Obit, J. H. (2021). A Comprehensive Review on Hybrid Network Traffic Prediction Models. International Journal of Electrical and Computer Engineering (IJECE), 11(2), 1450–1459
11. Agatonovic-Kustrin, S., & Beresford, R. (2000). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. Journal of Pharmaceutical and Biomedical Analysis, 22(5), 717–727. https://doi.org/10.1016/S0731-7085(99)00272-1
12. Anderson, D., & McNeill, G. (1992). Artificial neural networks technology. Kaman Science Corporation, 6, 1–83.
13. Zain, A. M., Haron, H., & Sharif, S. (2010). Prediction of surface roughness in the end milling machining using artificial neural network. Expert Systems with Applications, 37(3), 1755–1768. https://doi.org/10.1016/j.eswa.2009.07.033
14. Birdi, Y., Aurora, T., & Arora, P. (2013). Study of artificial neural networks and neural implants. International Journal on Recent and Innovation Trends in Computing and Communication, 1(4), 1–6.
15. Basheer, I. A., & Hajmeer, M. (2000). Artificial neural networks: Fundamentals, computing, design, and application. Journal of Microbiological Methods, 43(1), 3–31. https://doi.org/10.1016/S0167-7012(00)00201-3
16. Chen, G., Fu, K., Liang, Z., Sema, T., Li, C., Tontiwachwuthikul, P., & Idem, R. (2014). The genetic algorithm based back propagation neural network for MMP prediction in CO_2-EOR process. Fuel, 126, 202–212. https://doi.org/10.1016/j.fuel.2014.02.034
17. Pentoś, K., Mbah, J. T., Pieczarka, K., Niedbała, G., & Wojciechowski, T. (2022). Evaluation of Multiple Linear Regression and Machine Learning Approaches to Predict Soil Compaction and Shear Stress Based on Electrical Parameters. Applied Sciences, 12(17), 8791. https://doi.org/10.3390/app12178791
18. Jomthanachai, S., Wong, W. P., & Khaw, K. W. (2022). An application of machine learning regression to feature selection: A study of logistics performance and economic attribute. Neural Computing and Applications, 34, 15781–15805. https://doi.org/10.1007/s00521-022-07266-6
19. Pravalika, D., Sarla, P., & Reddy, R. A. (2025). Comparative analysis of M/G/K queueing model with artificial neural networks for service system dynamics. Discover Applied Sciences, 7(5). https://doi.org/10.1007/s42452-025-06591-4

20. Krishna, T. V., & Reddy, D. M. (2020). Self-similar behavior of CPI headline inflation and the role of Hurst exponent in determining CPI core inflation. Advances in Mathematics: Scientific Journal, 9(11), 9377–9392. https://doi.org/10.37418/amsj.9.11.43

21. Khinchin, A. Y. (1934). Korrelationstheorie der stationären stochastischen Prozesse. Mathematische Annalen, 109(1), 604–615. https://doi.org/10.1007/BF01449156

Pravalika Dhadi,
Department of Mathematics,
SR University,
Telangana, India,506371.
E-mail address: pravalikadhadi47@gmail.com

and

Archana Reddy.R,
Department of Mathematics,
SR University,
Telangana, India,506371.
E-mail address: r.archanareddy@sru.edu.in

and

Pushpalatha Sarla,
Department of Mathematics,
Sumathi Reddy Institute of Technology for Women,
Telangana, India.
E-mail address: pushpa.sarla@gmail.com

and

Ramesh Alluguvelli,
Department of Mathematics,
Geethanjali College of Engineering and Technology,
Telangana, India.
E-mail address: alluramesh1@gmail.com