The Regression Analysis of the Data to Determine the Buffer Size When Serving a Self-Similar Packets Flow

Gennadiy Linets [a], Roman Voronkin [a], Svetlana Govorova [a], Ilya Palkanov [a], Carlos Grilo [b]

[a] *North Caucasus Federal University, 2 Kulakova str, Stavropol, 355029, Russia*
[b] *Instituto Polit´écnico de Leiria, Rua General Norton de Matos, Apartado 4133, 2411-901 Leiria, Portugal*

Abstract

Using the methods of regression analysis on the basis of simulation data, a model for predicting the queue size of the input self-similar packet flow, distributed according to the Pareto law when it is transformed into a flow having an exponential distribution, is constructed. Since the amount of losses in the general case does not give any information about the efficiency of using the buffer memory space in the process of transforming a self-similar packet flow, a quality metric (penalty) was introduced to get the quality of the models after training, which is a complex score. This criterion considers both packet loss during functional transformations and ineffective use of the buffer space in switching nodes. The choice of the best model for predicting the queue size when servicing a self-similar packet flow was carried out using the following characteristics: the coefficient of determination; root-mean-square regression error; mean absolute error; the penalty score. The best in terms of the investigated characteristics are the models using the isotonic regression and the support vector regression.

Keywords [1]

Telecommunication network, self-similar traffic, Hurst exponent, Pareto distribution, packet loss, regression analysis, quality metrics, penalty score, machine learning.

1. Introduction

The main reason leading to a buffer overflow is the presence of a long-term dependence in network traffic due to its self-similarity, as a result of which the total cumulative effect in a wide range of delays can significantly differ from that observed in a short-term dependent process [1]. To eliminate self-similarity of network traffic, various models and traffic transformation devices are used, one of which is the asynchronous simulation model described in [2-4], for which there is a software implementation [5].

An important indicator of the operation of this model is the queue size used in the traffic transformation process. Since, due to limited computer resources, the queue cannot have an infinite size, the problem arises of predicting the queue size depending on the measure of self-similarity of the input traffic, which is the Hurst exponent.

The solution to the problem of finding the optimal buffer size for a given value of the Hurst exponent H can be found using the methods of regression analysis, based on simulation data obtained using the developed software [5].

2. Statement of the problem

Using machine learning methods, it is necessary to develop a model to predict the queue size depending on the Hurst exponent value based on the data obtained when performing the transformation of an input self-similar flow distributed according to the Pareto law into a flow having an exponential distribution.

YRID-2020: International Workshop on Data Mining and Knowledge Engineering, October 15-16, 2020, Stavropol, Russia
EMAIL: kbytw@mail.ru (Gennadiy Linets); roman.voronkin@gmail.com (Roman Voronkin); mitnik2@yandex.ru (Svetlana Govorova); ilya0693@yandex.ru (Ilya Palkanov); carlos.grilo@ipleiria.pt (Carlos Grilo)
ORCID: 0000-0002-2279-3887 (Gennadiy Linets); 0000-0002-7345-579X (Roman Voronkin); 0000-0002-3225-1088 (Svetlana Govorova); 0000-0003-0751-3928 (Ilya Palkanov); 0000-0001-9727-905X (Carlos Grilo)

Since machine learning includes many methods, at the initial stage, for further comparison with more complex models built, in particular, using deep learning methods, it is advisable to consider only methods of pairwise regression analysis, isotonic regression and support vector machines.

Let us involve a quality metric (penalty), which is a complex score and considers both packet loss during traffic transformation and inefficient use of buffer space.

Next, we choose the best model for predicting the queue size, depending on the Hurst exponent of the input flow, using the following quality metrics:

- coefficient of determination;
- root mean square error of regression;
- mean absolute error;
- penalty score value.

When setting the problem, special attention should be paid to testing the resulting models. In this case, the classical approach, which consists in dividing the entire data set into training and test samples, is not acceptable. Since on the test sample we will get the estimated number of lost packets and, therefore, the estimated penalty based on the difference between the predicted and actual buffer sizes, the obtained models must be tested by simulating traffic transformation with a queue size limitation, based on the results of applying the tested model for determining the size of the queue to the sequence being converted. This task is not trivial and will not be discussed in this article.

3. The solution of the problem

The simulation model presented in [2] provides transformation of the input flow of packets, which is obviously self-similar, into a given distribution law, in particular, into an exponential one. The object of transformation is a one-dimensional distribution density of time intervals between packets of the input flow. Using the developed model, 11,000 tests were carried out and data were obtained for statistical analysis.

Since the amount of losses in the general case does not give any information about the efficiency of using the queue in the process of the transformation traffic, to assess the quality of the resulting model, we introduce a quality metric - the penalty score, which takes into account not only the amount of losses, but also not rational use of buffer memory.

Let define y_i as the true value of the queue size in the sample, $\hat{y}_i$ is the predicted value of the queue size in the sample corresponding to the true value y_i. If $y_i > \hat{y}_i$, we will penalize the learning system by $\beta_+ \cdot (y_i - \hat{y}_i)$. If $y_i \geq \hat{y}_i$, the amount of the penalty will depend on the value of the difference $\Delta_i = y_i - \hat{y}_i$, with $\Delta_i \geq \Delta$ the amount of the penalty will be $\beta_- \cdot (\Delta_i - \Delta)$ and 0 – otherwise. Let us illustrate this with an example (Figure 1).

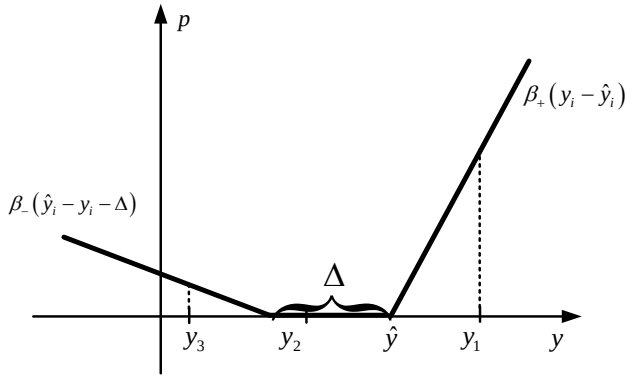

Figure 1: Graph of the dependence of the amount of the penalty on the buffer volume

Consider three cases, each of which corresponds to the true values of the queue sizes y_1, y_2 and y_3. Suppose the predicted values of the queue sizes coincide in each of the three cases, in other words $\hat{y}_1 = \hat{y}_2 = \hat{y}_3 = \hat{y}$. Then, in the first case, $y_1 > \hat{y}$ and the amount of the penalty is determined as $\beta_+ \cdot (y_1 - \hat{y})$. In the second case $\hat{y} - \Delta \leq y_2 \leq \Delta$ and the penalty is 0, it is assumed that $\hat{y}$ is the preferred queue size for y_2. In the third case $y_3 < \hat{y} - \Delta$, the amount of the penalty is determined from the expression $\beta_- \cdot (\hat{y} - y_3 - \Delta)$.

Thus, the amount of the penalty score will be determined from the equation:

$$p_i = \begin{cases} \beta_+ \cdot (y_i - \hat{y}), & \text{if } y_i > \hat{y}_i, \\ \beta_+ \cdot (\hat{y}_i - y_i - \Delta), & \text{if } \hat{y}_i - y_i > \Delta, \\ 0, \text{otherwise.} \end{cases}$$

The total penalty for all trials is determined as the arithmetic mean between the penalties for each trial:

$$p = \frac{1}{n} \sum_{i=1}^{n} p_i$$

where n is the number of tests. In the process of training the model, it is necessary to ensure the minimum value of the penalty for all tests, in other words $p \to \min$.

The presented system of penalties provides for the introduction of three hyperparameters: β_+, β_- and Δ, where $\beta_+ > \beta_- > 0$ and $\Delta > 0$. Let's set the hyperparameter values as follows: $\beta_- = 0.3,\ \beta_+ = 1,\ \Delta = 50$.

3.1. The initial data analysis

Figure 2 shows a scatter plot of queue size in dependence of Hurst exponent. The figure clearly shows that there is a certain correlation between the Hurst exponent and the buffer size [4].

Let us first group the tests by the value of the Hurst exponent and then select 30 groups to estimate the spread of the queue size.

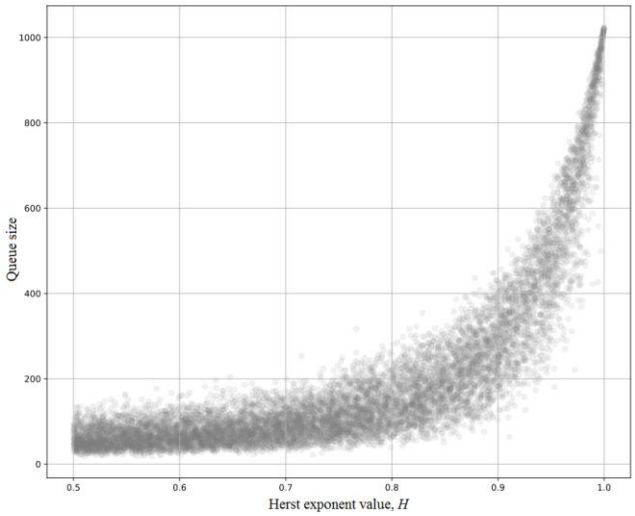

Figure 2: The scatter plot of queue size in dependence of the Hurst exponent

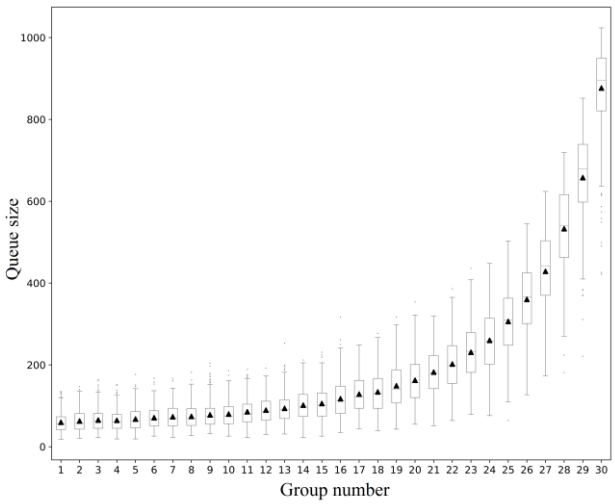

Figure 3: The box-plot of the 30 analyzed groups

Next, we can build a box-plot for each group. It follows from Figure 3 that the largest amount of outliers from above is observed for the first 10 groups, which corresponds to the Hurst exponent value close to 0.5. Consequently, at these values of the Hurst exponent, losses may occur due to the fact that the required buffer size will be greater than the predicted one.

For groups from 28 to 30, there are significant outliers from the bottom, which leads to inefficient use of buffer memory.

3.2. The regression analysis

Machine learning is a subset of artificial intelligence that studies and explores algorithms that can learn without direct programming. Linear regression is a typical representative of machine learning algorithms [7].

There are the following tasks solved by machine learning: supervised learning, unsupervised learning, reinforcement learning, active learning, knowledge transfer, etc. Regression (as well as classification) belongs to the class of supervised learning problems, when a certain target variable must be predicted for a given set of features of the observed object. As a rule, in supervised learning problems, experience E is

represented as a set of pairs of features and target variables: $D=\{(x_i, y_i)\}_i = 1...n$. In the case of linear regression, the feature description of an object is a real vector $x \in R^m$, where R is the set of real numbers and the target variable is a scalar $y \in R$. The simplest measure of the quality L for the regression problem is

$$L(y, \hat{y}) = (y - \hat{y})^2,$$

where $\hat{y}$ is an estimate of the real value of the target variable [7, 8].

Let us restore the dependence shown in Figure 2 using the methods of regression analysis.

The basis of regression analysis is the method of least squares (OLS), according to which the function $y = f(x)$ is taken as the regression equation such that the sum of the squares of the differences would satisfy

$$S = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \to \min.$$

Using the methods of pairwise regression analysis, we will carry out a statistical analysis of the data obtained by transforming an input self-similar flow distributed according to the Pareto law into a flow having an exponential distribution. Let us examine the methods widely used in practice, which allow finding the buffer size for the input flow with a given Hurst exponent.

3.3. The linear regression analysis

In this case, the relationship between the Hurst exponent H and queue size $\hat{y}$ is determined according to the linear equation:

$$\hat{y} = b_0 + b_1 H.$$

We obtain the regression equation using the least squares method:

$$\hat{y} = -611.182 + 1077.442H \quad (1)$$

The result of the fitting for the current model is shown on Figure 4.

OLS Regression Results

Dep. Variable:	enqueued_max	R-squared:	0.585
Model:	OLS	Adj. R-squared:	0.585
Method:	Least Squares	F-statistic:	1.549e+04
Date:	Tue, 06 Oct 2020	Prob (F-statistic):	0.00
Time:	16:10:34	Log-Likelihood:	-69228.
No. Observations:	11000	AIC:	1.385e+05
Df Residuals:	10998	BIC:	1.385e+05
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-611.1826	6.619	-92.331	0.000	-624.158	-598.207
hurst	1077.4428	8.657	124.459	0.000	1060.473	1094.412

Omnibus:	2822.757	Durbin-Watson:	1.989
Prob(Omnibus):	0.000	Jarque-Bera (JB):	7508.136
Skew:	1.378	Prob(JB):	0.00
Kurtosis:	5.964	Cond. No.	10.9

Figure 4: The linear model report is built using the statsmodels package of the Python programming language

Thus we obtained the statistically significant result. In Table 1 the quality metrics values are shown for the obtained linear regression model.

Table 1

Linear regression model quality metrics

Quality Metric	Value
Coefficient of determination R2	0.584
Root mean square error of regression RMSE	130.908
Mean absolute error MAE	96.808
Penalty score p	55.710

The obtained value of the coefficient of determination suggests that only about 58% of cases of changes in the Hurst exponent lead to a change in the size of the queue within the framework of the linear model. The obtained result is unsatisfactory for practice, therefore, in the simplest case, it makes sense to consider other methods using the methods of linearization of nonlinear dependencies. As a result, the nonlinear dependence can be reduced to linear, and then, the least squares method can be used.

3.4. The hyperbolic regression

For the hyperbolic regression, the relationship between H and $\hat{y}$ can be described as follows:

$$\hat{y} = b_0 + \frac{b_1}{H}.$$

The linearization of the hyperbolic equation is achieved by replacing $\frac{1}{H}$ with a new variable, which we denote by z [6]. Then the hyperbolic regression equation takes the form $\hat{y} = b_0 + b_1 z$. We obtain the regression equation using the least squares method:

$$\hat{y} = 875.438 - \frac{489.379}{H} \quad (2)$$

The result of the fitting for the current model is shown on Figure 5.

OLS Regression Results

Dep. Variable:	enqueued_max	R-squared:	0.453
Model:	OLS	Adj. R-squared:	0.453
Method:	Least Squares	F-statistic:	9118.
Date:	Tue, 06 Oct 2020	Prob (F-statistic):	0.00
Time:	16:10:34	Log-Likelihood:	-70741.
No. Observations:	11000	AIC:	1.415e+05
Df Residuals:	10998	BIC:	1.415e+05
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	875.4380	7.239	120.935	0.000	861.249	889.628
hurst	-489.3798	5.125	-95.487	0.000	-499.426	-479.334

Omnibus:	3395.600	Durbin-Watson:	1.984
Prob(Omnibus):	0.000	Jarque-Bera (JB):	9877.538
Skew:	1.627	Prob(JB):	0.00
Kurtosis:	6.312	Cond. No.	10.6

Figure 5: The first hyperbolic model report is built using the statsmodels package of the Python programming language

Thereby, we obtained the statistically significant result. In Table 2 the quality metrics values are shown for the obtained first hyperbolic regression model.

Table 2
Quality metrics of a hyperbolic regression model

Quality Metric	Value
Coefficient of determination R2	0.453
Root mean square error of regression RMSE	150.218
Mean absolute error MAE	110.511
Penalty score p	63.841

The obtained value of the coefficient of determination suggests that about 45% of cases of changes in the Hurst exponent lead to a change in the size of the queue. This is much worse than the value of the coefficient of determination of the linear model. For this reason, it makes sense to consider a different hyperbolic regression model:

$$\hat{y} = \frac{1}{b_0 + b_1 H}.$$

Using the least squares method, we obtain the regression equation for this model:

$$\hat{y} = \frac{1}{0.0399 - 0.0397 \cdot H} \tag{3}$$

The result of the fitting for the current model is shown on Figure 6.

OLS Regression Results

Dep. Variable:	enqueued_max	R-squared:	0.591
Model:	OLS	Adj. R-squared:	0.591
Method:	Least Squares	F-statistic:	1.591e+04
Date:	Tue, 06 Oct 2020	Prob (F-statistic):	0.00
Time:	16:10:34	Log-Likelihood:	43210.
No. Observations:	11000	AIC:	-8.642e+04
Df Residuals:	10998	BIC:	-8.640e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0400	0.000	166.113	0.000	0.040	0.040
hurst	-0.0397	0.000	-126.140	0.000	-0.040	-0.039

Omnibus:	3416.148	Durbin-Watson:	2.016
Prob(Omnibus):	0.000	Jarque-Bera (JB):	15829.080
Skew:	1.444	Prob(JB):	0.00
Kurtosis:	8.118	Cond. No.	10.9

Figure 6: The second hyperbolic model report is built using statsmodels package of the Python programming language

Accordingly we obtained the statistically significant result. In Table 3 the quality metrics values are shown for the obtained second hyperbolic regression model.

Table 3
Quality metrics of the modified hyperbolic model

Quality Metric	Value
Coefficient of determination R2	0.591
Root mean square error of regression RMSE	223.798
Mean absolute error MAE	77.543
Penalty score p	39.537

The obtained value of the coefficient of determination is about 59%, which is slightly better than the linear model.

3.5. The power regression

In the case of the power regression, the relationship between H and $\hat{y}$ is:

$$\hat{y} = b_0 H^{b_1}.$$

This equation is nonlinear in the coefficient b_1 and belongs to the class of regression models that can be reduced to linear form using transformations [6]

$$\ln y = \ln b_0 + b_1 \ln H.$$

The exponential function is internally linear, therefore, estimates of the unknown parameters of its linearized form can be calculated using the classical least squares method. The regression equation is:

$$\hat{y} = 401.143\ H^{3.596} \tag{4}$$

The result of the fitting for the current model is shown on Figure 7.

OLS Regression Results

Dep. Variable:	enqueued_max	R-squared:	0.699
Model:	OLS	Adj. R-squared:	0.699
Method:	Least Squares	F-statistic:	2.556e+04
Date:	Tue, 06 Oct 2020	Prob (F-statistic):	0.00
Time:	16:10:34	Log-Likelihood:	-7211.0
No. Observations:	11000	AIC:	1.443e+04
Df Residuals:	10998	BIC:	1.444e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	5.9943	0.008	732.235	0.000	5.978	6.010
hurst	3.5966	0.022	159.866	0.000	3.553	3.641

Omnibus:	69.944	Durbin-Watson:	1.994
Prob(Omnibus):	0.000	Jarque-Bera (JB):	54.313
Skew:	-0.086	Prob(JB):	1.61e-12
Kurtosis:	2.702	Cond. No.	5.55

Figure 7: The power model report is built using statsmodels package of the Python programming language

Thus we obtained the statistically significant result. In Table 4 the quality metrics values are shown for the obtained power regression model.

Table 4
Extent regression model quality metrics

Quality Metric	Value
Coefficient of determination R2	0.699
Root mean square error of regression RMSE	128.675
Mean absolute error MAE	72.823
Penalty score p	53.042

The obtained value of the coefficient of determination is 70%, which is much better than the coefficient of determination of the linear model.

3.6. The exponential regression

For the exponential regression, the relationship between H and $\hat{y}$ is:

$$\hat{y} = b_0 e^{b_1 H}.$$

This equation is non-linear with respect to the coefficient b_1 and belongs to the class of regression models, which are reduced to a linear form using transformations [6]:

$$\ln \hat{y} = \ln b_0 + H \ln b_1.$$

The exponential function is internally linear; therefore, estimates of the unknown parameters of its linearized form can be calculated using the classical least squares method. The regression equation is:

$$\hat{y} = 2.926 \cdot e^{5.089H} \quad (5)$$

The result of the fitting for the current model is shown on the Figure 8.

, In this way we obtained the statistically significant result. In Table 4 the quality metrics values are shown for the obtained exponential regression model.

OLS Regression Results

Dep. Variable:	enqueued_max	R-squared:	0.746
Model:	OLS	Adj. R-squared:	0.746
Method:	Least Squares	F-statistic:	3.226e+04
Date:	Tue, 06 Oct 2020	Prob (F-statistic):	0.00
Time:	16:10:35	Log-Likelihood:	-6284.5
No. Observations:	11000	AIC:	1.257e+04
Df Residuals:	10998	BIC:	1.259e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.0738	0.022	49.564	0.000	1.031	1.116
hurst	5.0891	0.028	179.621	0.000	5.034	5.145

Omnibus:	92.967	Durbin-Watson:	1.998
Prob(Omnibus):	0.000	Jarque-Bera (JB):	89.695
Skew:	-0.196	Prob(JB):	3.33e-20
Kurtosis:	2.794	Cond. No.	10.9

Figure 8: The exponential model report is built using statsmodels package of the Python programming language

Table 5

Exponential regression model quality metrics

Quality Metric	Value
Coefficient of determination R2	0.745
Root mean square error of regression RMSE	112.443
Mean absolute error MAE	65.199
Penalty score p	46.768

The obtained value of the coefficient of determination indicates that about 74% of cases of changes in the Hurst exponent lead to a change in the size of the queue in the framework of the exponential model, which is the best result when using the methods of paired regression analysis. An analysis of the amount of the penalty gives the same result.

Let us carry out a comparative analysis of the results obtained and then build graphs of the regression equations (1-5) (Figure 9). It is obvious that exponential regression most closely fits the relationship between the Hurst exponent and the buffer size.

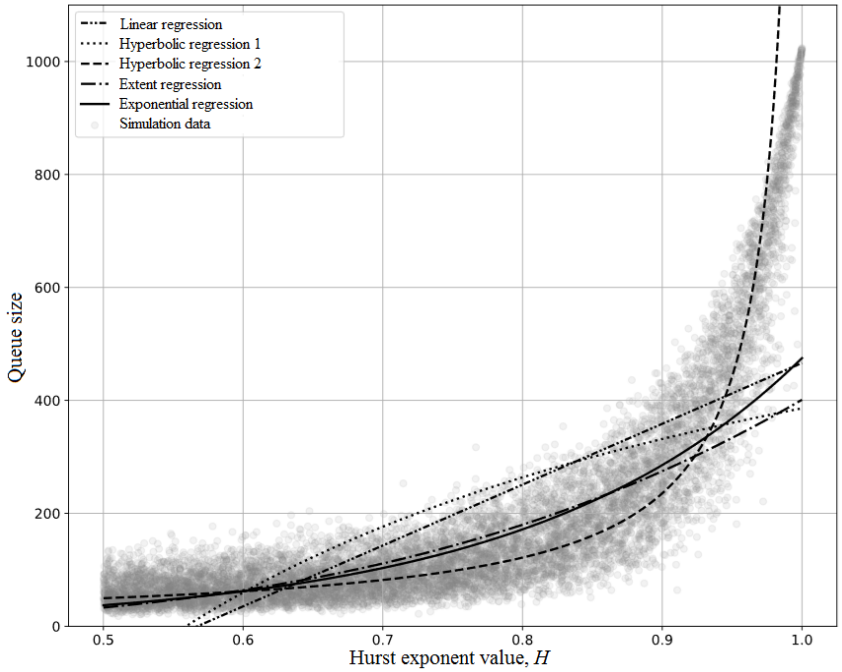

Figure 9: The comparative analysis of the results of paired regression analysis

The trivial paired regression models described above do not adequately describe the dependence of the queue size on the Hurst exponent, so we complicate the model. One possible way is the isotonic regression usage.

3.7. The isotonic regression

In statistics, isotonic regression or monotonic regression is a method of fitting a free-form line to a sequence of observations under the following constraints: the fitted free-form line should be non-decreasing (or not increasing) over the domain, and should lie as close as possible to the observations [13]. In the process of constructing an isotonic curve, the following problem is solved [13]:

$$\sum_i w_i \left(y_i - \hat{y}_i \right)^2 \to \min,$$

where the value of the weighting factor is $w_i > 0$. This gives a vector that consists of the non-decreasing elements that are closest in terms of the root mean square error. In practice, this list of elements forms a piecewise linear function.

Let us train the isotonic regression model using the scikit-learn package of the Python 3 programming language [9] and build a graph corresponding to the model built using isotonic regression (Figure 10)

In Table 6 the quality metrics values are shown for the obtained isotonic regression model.

Table 6

Isotonic regression quality metrics

Quality Metric	Value
Coefficient of determination R2	0.928
Root mean square error of regression RMSE	54.437
Mean absolute error MAE	39.501
Penalty score p	21.269

The obtained value of the coefficient of determination suggests that about 92% of cases of changes in the Hurst exponent lead to a change in the size of the queue within the framework of this model, which is much better than the models built on the basis of pair regression methods. Moreover, the value of the penalty for isotonic regression is two times less than the corresponding value for paired regression.

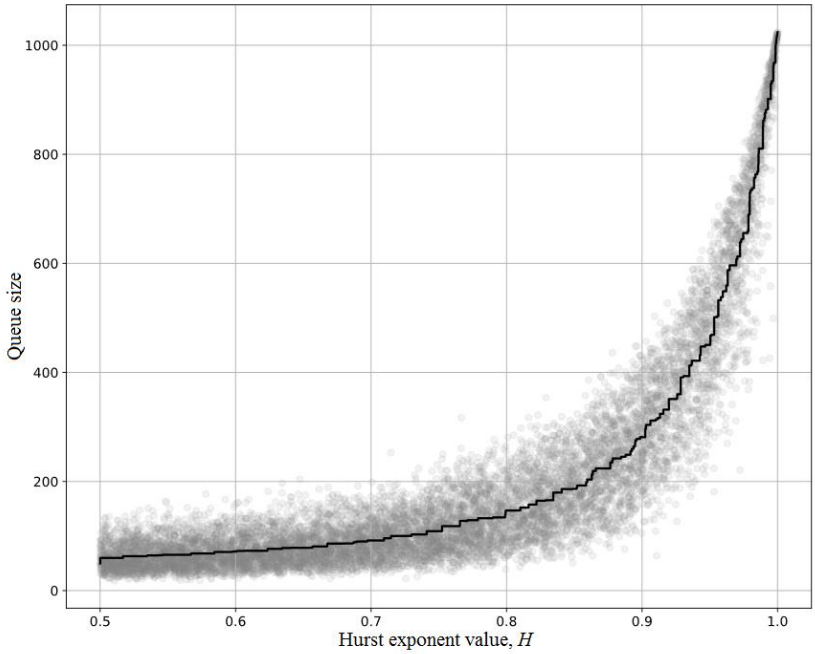

Figure 10: Plotting an isotonic curve to a dataset

3.8. The support vector regression

Support Vector Machines (SVM) is a linear algorithm used in classification and regression problems (for regression problems it is called SVR - Support Vector Regression). The main idea of the method is to construct a hyperplane that separates the sampled objects in an optimal way [10-12].

Support vector machines maximize the padding of objects, which is closely related to minimizing the likelihood of overfitting. Moreover, it makes it very easy to go to the construction of a nonlinear dividing surface due to the nuclear transition [10, 119].

Let us train the model based on SVR. The nonlinear nature of the relationship between the Hurst exponent value and the queue size indicates the need to choose a radial basis kernel for the SVR model. This model was trained using the scikit-learn package of the Python 3 programming language [12]. In Figure 11 the graph of the relationship between queue size and Hurst exponent is shown.

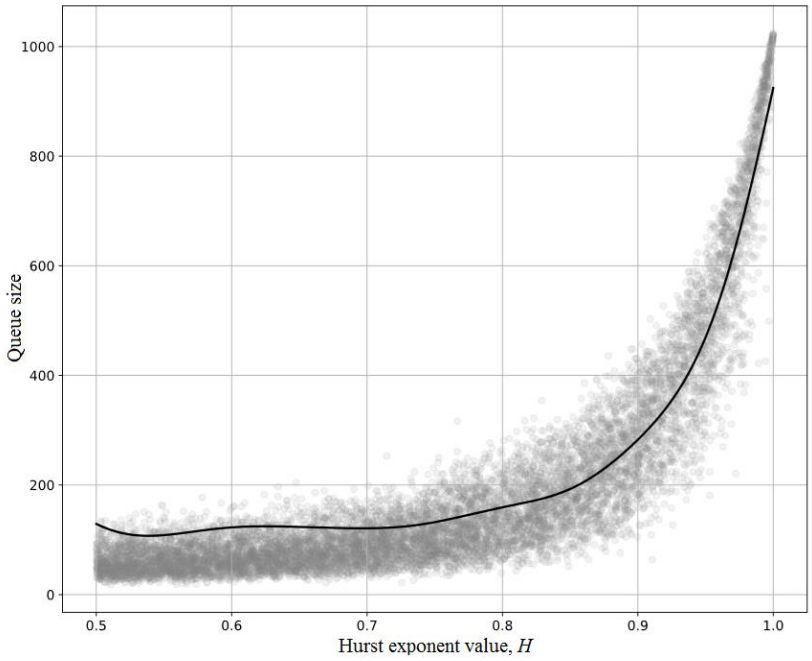

Figure 11: Plot corresponding to trained support vector machine

In Table 7 the quality metrics values are shown for the obtained support vector regression model.

Table 7
Support vector model quality metrics

Quality Metric	Value
Coefficient of determination R2	0.901
Root mean square error of regression RMSE	63.868
Mean absolute error MAE	52.506
Penalty score p	18.374

The obtained value of the coefficient of determination is about 90%, which is slightly worse than that of the method using isotonic regression. However, the penalty for this method is less than for isotonic regression.

3.9. Comparative analysis of models

The research results are presented in Table 8 for estimating and choosing the best method for predicting queue size from the Hurst exponent.

Based on the data of the pivot table, it can be concluded that the best predictive ability based on the introduced quality metric is a model built using the support vector machine. Within the framework of this study, it can be concluded that the complication of SVR by transition to the rectifying space does not lead to an improvement in the quality of learning.

Table 8
The comparison between considered regression methods for $0.5<H<1$

	Coefficient of determination R2	Root mean square error of regression RMSE	Mean absolute error MAE	Penalty score p
Linear regression	0.584	130.908	96.808	55.710
Hyperbolic regression 1	0.453	150.218	110.511	63.841
Hyperbolic regression 2	0.591	223.798	77.543	39.537
Extent regression	0.699	128.675	72.823	53.042
Exponential regression	0.745	112.443	65.199	46.768
Isotonic regression	0.928	54.437	39.501	21.269
Support vector machine SVR	0.901	63.868	52.506	18.374

4. Conclusions

Thus, we investigated seven models that allow to predict the size of the queue when transforming an input flow with a Pareto distribution into an output flow with an exponential distribution depending on the Hurst exponent of the input flow, built on the basis of regression analysis methods.

The unacceptability of the classical approach, which consists in dividing the entire dataset into training and test samples, is shown. Since, within the framework of the set task, the obtained models must be tested by simulation modeling of traffic transformation with a limit on the queue size, based on the results of applying the tested model for determining the queue size to the sequence being converted.

Since the amount of losses in the general case does not give any information about the efficiency of using the queue in the process of converting traffic, to assess the quality of the resulting model, a penalty was introduced that takes into account not only the amount of losses, but also not rational use of buffer memory.

The best for the selected quality metrics are the isotonic regression and the support vector regression. It managed to to reduce the penalty score for these models by more than two times in comparison with the trivial linear model. The use of these models will make it possible to more efficiently use the buffer space of the RAM of telecommunication network switching nodes. The usage of these models will make it possible to more efficiently use the buffer space of the RAM in telecommunication network switching nodes. Nevertheless, the obtained models do not belong to strong machine learning models, therefore, the additional researches are required using decision trees ensembles and neural networks.

5. References

[1] Shelukhin O. I., Fractal processes in telecommunications / O.I. Shelukhin, A.M. Tenyakshev, A.V. Aspen; Ed. O.I. Shelukhin. - M.: Radiotekhnika, 2003 - 479 p.

[2] Linets G.I., Govorova S.V., Voronkin R.A, Mochalov V.P., Simulation model of asynchronous transformation of self-similar traffic in switching nodes using a queue // Infocommunication technologies. 2019. T.17. №3. pp. 293-303.

[3] Basan, A.S., Basan, E.S., Lapina, M.A., Kormakova, V.N., Lapin, V.G. Security methods for a group of mobile robots according to the requirements of Russian and foreign legislation: IOP Conference Series: Materials Science and Engineering, 2020, 873(1), 012031 DOI: 10.1088/1757-899X/873/1/012031.

[4] Linets G.I., Melnikov S.V., Govorova S.V., Medenec V.V., Lapina M.A. Decrease energy consumption of transport telecommunication networks due to the usage of stage-by-stage controlling procedure: CEUR Workshop Proceedings REMS 2018 – Proceedings of the 2018 Multidisciplinary Symposium on Computer Science and ICT. 2018. Pp. 181-190.

[5] Linets G.I., Govorova S.V., Voronkin R.A, A program for generating a dataset to study the statistical characteristics of a self-similar traffic transformation model. Certificate of state registration of a computer program № 2019619275. Register date 15.07.19.

[6] Handbook of Mathematics. Sixth Edition / I.N. Bronshtein, K.A. Semendyayev, G. Musiol, H. Mühlig.URL: https://doi.org/10.1007/978-3-662-46221-8
[7] Basic principles of machine learning on the example of linear regression URL: https://habr.com/ru/company/ods/ blog /322076/
[8] Luis Pedro Coelho, Willie Richart. Building machine learning systems in Python. 2nd edition / translate from English. Slinkin A.A. M.: DMK Press, 2016, 302 p.
[9] Isotonic regression. URL: https://scikit-learn.org/stable/modules/iso tonic.html (дата обращения 01.06.2020).
[10] Chardin B., Massaron L., Boschetti A. Large-scale machine learning with Python / translate from English. A.V. Logunova. M.: DMK Press, 2018.358 p. (in Russian).
[11] Raska S. Python and machine learning / translate from English. A.V. Logunova. M.: DMK Press, 2017.418 p. (in Russian)
[12] Support Vector Regression (SVR) using linear and non-linear kernels. URL: https://scikit-learn.org/stable/auto_exam ples/svm/plot_svm_regression.html?highlight=svr.
[13] Westling T., Gilbert P., Carone M. Causal isotonic regression. URL: http://arxiv.org/abs/1810.03269.