CYBERNETICS AND PHYSICS, VOL. 12, NO. 2, 2023, 129–135

FORECASTING THE STATE OF COMPLEX NETWORK SYSTEMS USING MACHINE LEARNING METHODS[1]

Nikita Knyazev
Saint Petersburg State University
Russia
nikitaknyazev@yandex.ru

Anton Pershin
Saint Petersburg State University
Russia
a.pershin@spbu.ru

Anna Golovkina
Saint Petersburg State University
Russia
a.golovkina@spbu.ru

Vladimir Kozynchenko
Saint Petersburg State University
Russia
v.kozynchenko@spbu.ru

Article history:
Received 18.05.2023, Accepted 15.09.2023

Abstract

The rapid development of information technology leads to the fact that the number of users in telecommunication networks is constantly growing along with the need to improve the quality of service. Network traffic prediction is an important task in this area, as it underlies network diagnostics and efficient use of its resources. Various linear and non-linear methods such as neural networks are actively used to analyze temporal and spatial relationships and forecast network traffic. However, most of them cannot accurately describe the dynamics of the network, as the importance of different nodes changes over time, which complicates the topology. This article proposes to modify autoregressive and gradient boosting models to detect spatial features and work with data with network structure in order to solve the above problem. Experimental results on three publicly available datasets with network traffic show that the proposed methods are superior to their one-dimensional counterparts and can compete with the most modern solutions. Additionally, it was found that the logarithmic transformation significantly increases the accuracy of the forecast, and models based on decision trees are superior to autoregressive ones. Also, increasing the size of the training sample does not always improve the accuracy of the forecast. Moreover, singular spectrum analysis is superior to exponential smoothing and moving average for Internet traffic. The performance of the proposed models achieved $MAPE$ values of 4.4% and 8.9% on the PeMSD7 dataset for gradient boosting and autoregression methods, respectively. Thus, using these models as a forecasting tool will help optimize complex network systems and improve the quality of their service.

Key words

Time series, network traffic forecasting, machine learning, complex network systems.

1 Introduction

With the advent and development of computer technology and the Internet, a huge number of devices are connected to the network, which leads to an increase in its scale and complexity, on the other hand, user requirements for communication quality are only growing. By properly allocating network resources, you can optimize routing, thereby reducing communication delays, preventing network congestion, and ensuring quality of service. However, network diagnostics, anomaly detection, and effective use of available tools require accurate prediction of key network metrics such as traffic, latency, calls, etc. Thus, network traffic prediction is the basis for improving the quality of network service.

From the point of view of telecommunications, network traffic is a matrix that describes the amount of data flow between all pairs of network nodes at a given time. The problem of network traffic prediction can be formulated as predicting a given matrix

[1]The work was partially supported by Saint Petersburg State University, project ID: 94029367

at a certain point in time in the future based on historical data.

A large number of works are devoted to the problem of predicting network traffic. Some of them use classical statistical methods ([Tikunov and Nishimura, 2007; Yang et al., 2018]), but currently there is a tendency to use machine learning algorithms ([Troia et al., 2018; Chen et al., 2016; Cortez et al., 2012]), among which graph neural networks stand out ([Yu et al., 2018; Lu et al., 2019; Jiang and Luo, 2022]). The main problem of state-of-the-art solutions is the complexity of the model: they usually have many layers with a huge number of connections, require a lot of time and computational resources for training, and tuning parameters can be quite complicated. Being black box models, they also lack interpretability and explainability compared to statistical methods. Most of the papers reviewed are related to road traffic ([Chen et al., 2016; Yu et al., 2018; Lu et al., 2019; Jiang and Luo, 2022]), while network traffic prediction remains understudied despite great interest and demand in branch. This can be explained by the greater complexity of network traffic time series: their dynamics is much more stochastic and noisy, and links between nodes can be discontinuous. However, network traffic prediction has a wider scope, including network optimization, control and fault detection, while traffic prediction is usually limited to solving transport problems.

In general, the methods used to solve this problem can be divided into two groups: linear and nonlinear. Linear methods, such as autoregressive [Tikunov and Nishimura, 2007; Yang et al., 2018], model the characteristics of time series based on mathematical statistics. Nonlinear prediction models include wavelet analysis [Deineko, Zhanna, 2017; Rumaih et al., 2002], Bayesian networks [Maarten et al., 2014], neural networks [Troia et al., 2018; Chen et al., 2016; Cortez et al., 2012], convolutional neural network [Ghaderzadeh et al, 2021] and other methods of artificial intelligence [Ghaderzadeh and Aria, 2021] and deep learning [Gheisari et al, 2023]. Since one linear or nonlinear model cannot accurately describe the dynamics of network traffic, recently combined methods [Lu et al., 2019] based on neural networks capable of modeling spatial and temporal dependencies (for example, graph neural network) are increasingly used to solve such problems. However, these methods create a static adjacency matrix to model the network topology, while spatial dependencies can be dynamic, i.e. the importance of different nodes will change over time, which complicates the network structure. As a result, this approach can seriously limit the ability to model complex network traffic.

It is important to remember that network traffic can switch between different dynamic modes, so it is necessary that the predictive model can adapt to sudden changes. In other words, the prediction model must be supplemented with an online adaptation algorithm. Online gradient descent ([Yang et al., 2018; Chen et al., 2019]) can be used as an adaptation algorithm. Gradient descent is a first-order iterative optimization algorithm for objective function minimization that is widely used in machine learning. Online gradient descent is a variant based on batch or stochastic gradient descent, with the batch variant having more stable convergence. But gradient descent has a significant drawback: it can converge to a local minimum and saddle points (the global minimum is guaranteed to be found only if the objective function is convex). Another online adaptation method is the sliding window method [Mehmood et al., 2021]. The sliding window method is a transformation algorithm based on the shift of an interval with values that allows you to generate subsamples of a time series for training and testing a predictive model. The process of this method is as follows: the window is shifted along the sequence per unit of observation, where each position forms a new sample.

To solve the problem of predicting network traffic, this paper proposes to modify a number of linear and nonlinear methods, adding to them the ability to process spatial information. As an algorithm for online adaptation of models, we will use the sliding window method. The article also analyzes various data filtering methods to reduce the impact of noise on models. The contribution of this paper is summarized as follows:

1. We focus on network (Internet) traffic, as its forecasting is of greater interest and demand from the industry. At the same time, compared to transport problems, network traffic is more complex: the dynamics of time series is stochastic and very noisy, and connections between nodes can be discontinuous.
2. Unlike most works devoted to short-term forecasting of time series, we are not limited to one- or two-stage forecasting, but are trying to explore the predictive ability of our models for tens and even hundreds of time steps.
3. Numerous experiments with three sets of real network and road traffic data confirm that the implemented models that process spatial features are superior to their one-dimensional counterparts, and they are also able to compete with state-of-the-art solutions.

The rest of the article is organized as follows: Section 3 is devoted to selected prediction algorithms and their modifications for working with network traffic. The results of computational experiments and discussions are listed in Section 4. Finally, in Section 5 we draw a conclusion.

2 Algorithms description

In this paper, we restrict ourselves to two groups of time series forecasting models: statistical and decision tree based methods.

Statistical models such as ARIMA, autoregression, exponential smoothing, etc. are based on a large theoretical base and have been successfully used in a number of areas [Brockwell et al., 2002; Lutkepohl, Helmut, 2005] where prediction is part of decision making and the model must provide information for probabilistic estimation risk.

The second group of methods is very effective in solving time series forecasting problems, which is demonstrated on large datasets of various types. For example, 4 of the top 5 models in the «M5 Competition» were based on LightGBM [Makridakis et al., 2022]. Therefore, we will test their effectiveness for time series of network traffic.

To add spatial information, we will use traffic from nearest neighbors for both groups of models.

2.1 Autoregressive models

The autoregressive model, $AR(p)$, assumes that the X_t time series can be estimated as a linear combination of previous p lags (1).

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \epsilon_t, \tag{1}$$

where

X_t is time series value corresponding to a certain unit of traffic at time t, for more information, see Section 3.1.
$a = [c, \varphi_1, \ldots, \varphi_p]^T$ is model parameters,
ϵ_t is white noise.

The multivariate version of the model, $VAR(p)$, differs only in that it uses a multivariate time series, $\mathbb{X}_t = (X_t^1, X_t^2, ..., X_t^N)$, where N is the dimension of the network system.

Real time series are characterized by some seasonality with one or more periods, so it is necessary to be able to include this part of the information in the model. Seasonal autoregressive models can be obtained in two alternative ways: either using seasonal order (2) or using seasonal dummies (3).

$$\mathbb{X}_t = c + \sum_{i=1}^{p} \varphi_i \mathbb{X}_{t-i} + \sum_{i=1}^{l} \nu_i \mathbb{X}_{t-iT} + \epsilon_t, \tag{2}$$

where l is the seasonal order, which determines how many seasons we have to look back in period $T \in \mathbb{Z}$.

$$\mathbb{X}_t = c + \sum_{i=1}^{p} \varphi_i \mathbb{X}_{t-i} + \sum_{i=1}^{T} n_{it} v_i + \epsilon_t. \tag{3}$$

where n_{it} is a binary variable that equals 1 if $t \, mod \, T = i$, otherwise $n_{it} = 0$.

2.2 Gradient boosting methods

Boosting is an ensemble algorithm that builds a strong model using a set of weak ones, while gradient boosting is a special case, treating it as an optimization problem in a function space. Well-known boosting algorithms such as XGBoost, LightGBM and CatBoost use f_k decision trees as weak models (4).

$$\hat{X}_t^{(K)} = \sum_{k=1}^{K} f_k(X_{t-1}, ..., X_{t-i}), \quad f_k \in \mathcal{F}, \tag{4}$$

where

$\hat{X}_t^{(K)}$ is the value of the time series predicted by the ensemble with K weak models;
K is a number of trees;
$\mathcal{F}$ is the set of all possible decision trees;
f_k is a function associated with a decision tree in the function space $\mathcal{F}$.

3 Experiments

We tested a number of combinations of smoothing algorithms (logarithmic, exponential, singular spectrum analysis [Hassani, Hossein, 2007]) and prediction algorithms on three available datasets: Abilene, Totem, PeMSD7. In addition, we aimed to study the influence of spatial information on the accuracy of the models. For all algorithms and datasets, we predict $T_{test} = 500$ steps ahead.

3.1 Data description

To train and test our models, we need datasets that contain a sufficient number of observations and have a graph structure. In this work, we have studied three open datasets related to traffic:

1. Abilene [Zhang, 2014] is a 12-router backbone network in North America that collected traffic volumes (bytes/s) of origin-destination flows aggregated at five-minute intervals over a 24-week period in 2004.
2. Totem GEANT [Direction Generale des Technologies, 2008] is a pan-European research network that handles all traffic from National Research and Education Networks connecting universities and research institutes. The network

consists of 23 nodes connected to each other by 74 channels, covers about 4 months (January-April 2005) with a step of 15 minutes and corresponds to the volume of input traffic, measured in kbps.
3. PeMSD7 [Yin, 2019] is a set of 228 stations in California's 7th District Congestion Measurement System, corresponding to miles per hour, and covering weekdays in May and June 2012.

The topologies of these datasets are given in the Appendix A.

3.2 Univariate models

In this section, we provide the best validation results among univariate models for the considered datasets.

3.2.1 Abilene

As you can see from the Table 1, with an equal window size, models based on decision trees show better results compared to autoregression. Moreover, the LightGBM model with SSA and logarithm achieved the best results for the Abilene dataset with a large window size. Despite the use of smoothing algorithms, the models have a relatively large MAE, which can be explained by the strong noise of the data and a lot of outliers. With an average value of 191.474 Mb, the standard deviation is 66699.796, which confirms the high noise level of the dataset.

Table 1. Best univariate models validation results for Abilene. The table is organized as follows: first, the best results for autoregression, XGB and LightGBM will be indicated on an average window of 3000, then the model with the best results for the entire dataset will be indicated.

Model	Smoothing	Train size	MAPE	MAE
AutoReg	SSA & Log	3000	0.178	12.381
XGB	SSA & Log	3000	0.182	12.030
LightGBM	SSA & Log	3000	0.180	11.662
LightGBM	SSA & Log	5000	0.174	11.287

3.2.2 Totem

According to the Table 2, the LightGBM and XGB models again give better results than autoregression on the average window size. In addition, as in the case of the previous dataset, Totem is very noisy and has areas with a sudden change in dynamics: with an average value of 687.023 Mb, the standard deviation is 63875.902. Thus, the models demonstrate large MAE and MAPE metrics. However, the LightGBM model achieves the best results ($MAE = 35.602$) for the Totem set with a large window and with SSA and logarithm smoothing.

Table 2. Best univariate models validation results for Totem. For more details, see Table 1.

Model	Smoothing	Train size	MAPE	MAE
AutoReg	SSA & Log	3000	0.309	45.463
XGB	SSA & Log	3000	0.265	40.356
LightGBM	SSA & Log	3000	0.259	39.362
LightGBM	SSA & Log	5000	0.267	35.602

3.2.3 PeMSD7

The results of the Table 3 show that in this case autoregression is superior to gradient boosting models. The PeMSD7 dataset is not so noisy and has almost no outliers compared to the first two sets, which confirms its standard deviation of 13.483 with an average of 58.889 mph. The best values of metrics were achieved by autoregression on a large window with smoothing in the form of a logarithm.

Table 3. Best univariate models validation results for PeMSD7. For more details, see Table 1.

Model	Smoothing	Train size	MAPE	MAE
AutoReg	Log	3000	0.100	4.617
XGB	SSA & Log	3000	0.114	5.132
LightGBM	SSA & Log	3000	0.111	4.954
AutoReg	Log	5000	0.098	4.500

3.3 Multidimensional models

In this section, we provide the best validation results among multivariate models for the considered datasets.

3.3.1 Abilene

Analyzing Tables 2 and 4, you can see that the models that were added the ability to process spatial information turned out to be better than one-dimensional ones. In addition, as in the one-dimensional case, models based on decision trees outperformed autoregression on a medium-sized window. The best results were achieved by the XGB model on a large window using SSA and logarithm. As for relatively large metrics, see section 3.2.1 for more details.

Table 4. Best multivariate models validation results for Abilene. For more details, see Table 1.

Model	Smoothing	Train size	MAPE	MAE
VAR	Log	3000	0.180	12.269
XGB	SSA & Log	3000	0.157	10.098
LightGBM	SSA & Log	3000	0.152	9.957
XGB	SSA & Log	5000	0.150	9.665

3.3.2 Totem

Comparing Tables 2 and 5, it can be concluded that multidimensional variants of boosting models showed better results compared to one-dimensional analogues. On the other hand, vector autoregression did not give the expected result. As with the previous dataset, decision tree-based models outperformed autoregression on the middle window. The LightGBM model showed the best results on the entire dataset with a large window and with SSA and logarithm. As for relatively large metrics, see section 3.2.2 for more details.

Table 5. Best multivariate models validation results for Totem. For more details, see Table 1.

Model	Smoothing	Train size	MAPE	MAE
VAR	SSA & Log	3000	0.311	47.064
XGB	SSA & Log	3000	0.254	38.407
LightGBM	SSA & Log	3000	0.246	38.062
LightGBM	SSA & Log	5000	0.243	35.316

3.3.3 PeMSD7

We got the following results: Table 6. If we compare them with the results of one-dimensional models (Table 3), we will again see that the multidimensional variants of the methods turned out to be better. Moreover, gradient boosting methods are about twice as good as autoregression. The best results were achieved by the LightGBM model with SSA smoothing and logarithm, but not at the maximum window size, but at 4000. If we compare the obtained results ($MAPE = 4.4\%$, $MAE = 2.064$) with existing state-of-the-art approaches, such as graph neural network [Yu et al., 2018], then our models are close to them in accuracy, and perhaps even better.

Table 6. Best multivariate models validation results for PeMSD7. For more details, see Table 1.

Model	Smoothing	Train size	MAPE	MAE
VAR	Log	3000	0.089	3.919
XGB	Log	3000	0.047	2.168
LightGBM	Log	3000	0.045	2.131
LightGBM	Log	4000	0.044	2.064

4 Conclusion

We have adapted a number of time series prediction algorithms (AR, VAR, XGBoost, LightGBM) for network traffic prediction. We also further tested the effect of smoothing (logarithmic, exponential smoothing, SSA) on the accuracy of these algorithms. In order to conduct a thorough assessment of the quality of the forecast, we defined a validation protocol based on a sliding window. To test for the presence of spatial dependencies and exploit them, we developed simple multivariate versions of all the univariate prediction algorithms used in the study. To study the sensitivity of the forecast accuracy to the length of the training time series, we tested the algorithms on window sizes from 1000 to 5000. Illustrations of the results are shown in Appendix B (Figures 4 and 5). Based on the results obtained, the following conclusions can be drawn:

1. The logarithmic transformation greatly improves the accuracy of the forecast.
2. Models based on decision trees usually outperform others.
3. AR, the simplest statistical model we used, is a competitive algorithm in the one-dimensional case with additional support for the seasonality of Internet traffic.
4. Increasing the learning window does not necessarily improve accuracy. Smaller training sample size may be preferable as it allows the prediction algorithm to be adaptive.
5. SSA outperforms exponential smoothing and moving average for internet traffic.
6. Multivariate algorithms are superior to univariate ones.
7. Our best results are comparable to state-of-the-art solutions.

Thus, if the task is simple time series forecasting, XGBoost or LightGBM algorithms should be used. On the other hand, although autoregressive methods are inferior in accuracy, they can be used to infer dependencies between system components, making them useful for network reconstruction and causality inference problems.

References

D. Tikunov and T. Nishimura (2007). Traffic prediction for mobile network using Holt-Winters's exponential smoothing. 15th International Conference on Software, Telecommunications and Computer Networks, pp. 1–5.

Cortez, Paulo and Rio, Miguel and Rocha, Miguel and Sousa, Pedro (2012). Multi-scale Internet traffic forecasting using neural networks and time series methods. Expert Systems, pp. 143–155.

Deineko, Zhanna (2017). Wavelet coherence as a tool for visualization of complex physical processes.

Rumaih M. Alrumaih and Mohammad A. Al-Fawzan (2002). Time Series Forecasting Using Wavelet Denoising an Application to Saudi Stock Index. Journal of King Saud University - Engineering Sciences, pp. 221–233.

Maarten van der Heijden and Marina Velikova and Peter J.F. Lucas (2014). Learning Bayesian networks for clinical time series analysis. Journal of Biomedical Informatics, pp. 94–105.

Hassani, Hossein (2014). Singular Spectrum Analysis: Methodology and Comparison. University Library of Munich, Germany, MPRA Paper.

Ghaderzadeh, M., Asadi, F., Jafari, R., Bashash, D., Abolghasemi, H., Aria, M., (2021). Deep Convolutional Neural Network-Based Computer-Aided Detection System for COVID-19 Using Multiple Lung Scans: Design and Implementation Study. Journal of medical Internet research.

Ghaderzadeh, M. and Aria, M. and Hosseini, A. and Asadi, F. and Bashash, D. and Abolghasemi, H., (2021). A fast and efficient CNN model for B-ALL diagnosis and its subtypes classification using peripheral blood smear images. International Journal of Intelligent Systems.

Ghaderzadeh, Mustafa and Aria, Mehrad, (2021). Management of Covid-19 Detection Using Artificial Intelligence in 2020 Pandemic. Association for Computing Machinery.

M. Gheisari, F. Ebrahimzadeh, M. Rahimi, M. Moazzamigodarzi, Y. Liu, A. Mehbodniya, M. Ghaderzadeh, et al, (2023). Deep learning: Applications, architectures, models, tools, and frameworks: A comprehensive survey. CAAI Transactions on Intelligence Technology.

Troia, Sebastian and Alvizu, Rodolfo and Zhou, Youduo and Maier, Guido and Pattavina, Achille (2018). Deep Learning-Based Traffic Prediction for Network Optimization. 20th International Conference on Transparent Optical Networks (ICTON), pp. 1–4.

Chen, Yuan-yuan and Yisheng, Lv and Li, Zhenjiang (2016). Long short-term memory model for traffic congestion prediction with online open data. pp. 132–137.

Lu, Zhilong and Lv, Weifeng and Xie, Zhipu and Du, Bowen and Huang, Runhe (2019). Leveraging Graph Neural Network with LSTM For Traffic Speed Prediction. IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovation, pp. 74–81.

Haimin Yang and Zhisong Pan and Qing Tao and Junyang Qiu (2018). Online learning for vector autoregressive moving-average time series prediction. Neurocomputing, pp. 9–17.

Yu, Bing and Yin, Haoteng and Zhu, Zhanxing (2018). Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting, pp. 3634–3640.

Weiwei Jiang and Jiayun Luo (2022). Graph neural network for traffic forecasting: A survey. Expert Systems with Applications.

Xi Chen and Hongzhi Wang and Yanjie Wei and Jianzhong Li and Hong Gao (2019). Autoregressive-Model-Based Methods for Online Time Series Prediction with Missing Values: an Experimental Evaluation. ArXiv.

Mehmood, Hassan and Kostakos, Panos and Cortes, Marta and Anagnostopoulos, Theodoros and Pirttikangas, Susanna and Gilman, Ekaterina (2021). Concept Drift Adaptation Techniques in Distributed Environment for Real-World Data Streams. Smart Cities, pp. 349–371.

Brockwell, Peter J and Davis, Richard A. (2002). Introduction to time series and forecasting.

Lütkepohl, Helmut (2005). New introduction to multiple time series analysis. Springer Science & Business Media.

Makridakis, Spyros and Spiliotis, Evangelos and Assimakopoulos, Vassilios (2022). M5 accuracy competition: Results, findings, and conclusions. International Journal of Forecasting.

Veritas Yin (2019). STGCN_IJCAI-18, PeMSD7 dataset. https://github.com/VeritasYin/STGCN_IJCAI-18/blob/master/dataset/PeMSD7_Full.zip.

Yin Zhang (2014). Abilene dataset. https://www.cs.utexas.edu/ yzhang/research/ AbileneTM/.

Direction Generale des Technologies, de la Recherche et de l'Energie of the Walloon government (2008). Totem dataset. https://totem.run.montefiore.uliege.be/ datatools.html.

Appendix A Dataset topologies

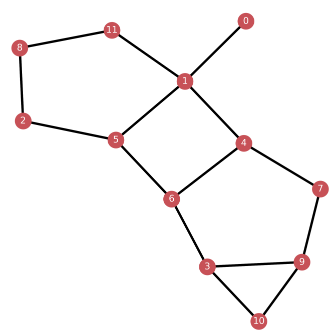

Figure 1. Abilene topology

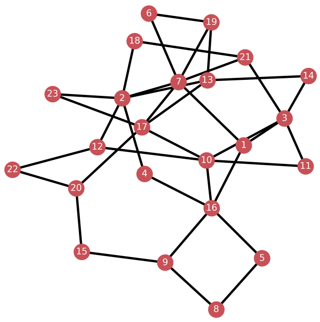

Figure 2. Totem topology

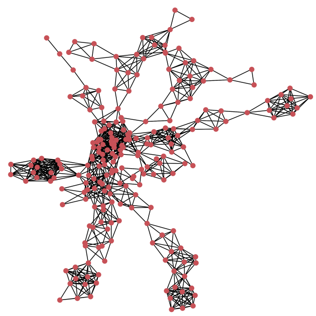

Figure 3. PeMSD7 topology

Appendix B Illustrations of model predictions

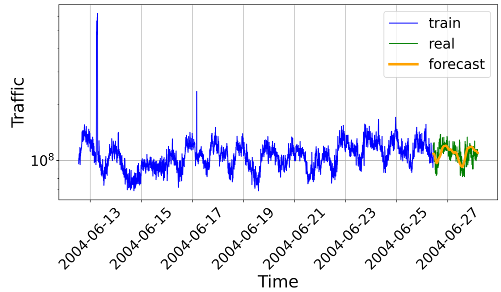

Figure 4. VAR forecast with SSA for Abilene dataset

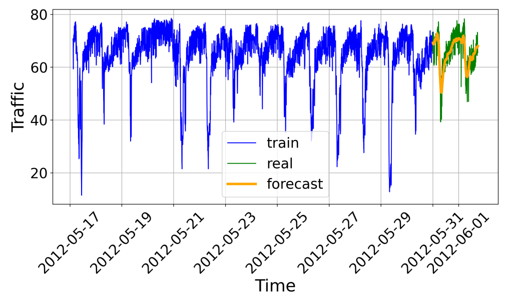

Figure 5. LightGBM forecast with SSA for PeMSD7 dataset