Data mining meets Network Analysis: Traffic Prediction Models

Teo Eterovic*, Sasa Mrdovic * , Dzenana Donko* and Zeljko Juric*
* Faculty of Electrical Engineering/ Department of CS and Informatics, Sarajevo, BiH
{teterovic, smrdovic, ddonko, zjuric} @etf.unsa.ba

Abstract - Most research on network traffic prediction has been done on small datasets based on statistical methodologies. This research analyzes an internet traffic dataset spanning multiple months using the data mining process. Each data mining phase was carefully fitted to the network analysis domain and systematized in context of data mining. The second part of the paper evaluates various seasonal time series prediction models (univariate), including ANN, ARIMA, Holt Winters etc., as a data mining phase on the given dataset. The experiments have shown that in most cases ANNs are superior to other algorithms for this purpose.

I. INTRODUCTION

Network traffic prediction is a crucial part of every serious network system. Network prediction enables network administrators to strategically plan and respond to network issues such as network infrastructure scaling, testing of new protocols/devices and to deal with network problems that already happened and could occur again in the future.

The construction of such a network traffic predictor could be the fundament of a network traffic generator that could be used for precise network traffic simulation and planning.

Lelend at al.(1994) has shown and proved [1] that the Ethernet traffic in LAN networks has an self-similarity nature. In 1998 *Feldman et. Al.* writes about the fractal nature of WAN traffic [2]. Their models have been verified in several experiments as in [3,4,5] on different datasets and network domains including the internet traffic[4] as a special class of network traffic. Other network traffic properties include multiscalarity and their nonlinear nature with long rage dependence as shown in [6]. This nature of network traffic enables us to do a complex and, with the available machine learning tools, a sufficiently accurate traffic prediction.

Although there are several papers that deal with the problem of network traffic prediction [7,8,9,10,11] most of them use simple - linear statistical models for time series prediction. Also most of the papers deal with only one dimension of the network traffic mostly the data they got from the SNMP logs. The problem with SNMP logs is that they miss the packet inter-arrival time parameter that is crucial for network generators. There are less papers that use machine learning models (in most cases simple ANN models or rarely SVM). Another problem is that most research has been done on relatively short time series that span usually over a few weeks focusing on short-term or medium-term time series prediction.

II. RELATED WORK

In the literature various models have been proposed in order to model the network traffic as a prediction problem. The methods vary from strictly statistical methods based on fitting statistical distributions to specific sources, models based on auto regression and moving average (AR, ARMA, ARIMA, SARIMA, GARCH)[8] or exponential smoothing methods such as Holt Winters that showed some superiority in daily predictions and computation time. Another approach are the machine learning models especially the non-linear ANN and kernel based SVM models[9][10], ensemble methods based on ANNs that give good results for short-term time series raging from five minutes or one hour[11]. There are also hybrid methods based on the combination of ANNs and ARIMA models[7].

III. DATA MINING - TIME SERIES ANALYSIS

For the purpose of this paper a new internet traffic dataset has been created on a web server. The dataset is recorded in *PCAP* format with *tcpdump* and spans from (10.10.2013. until 22.02.2014.) and it has been made publicly available at [12].

The data analyzed in this paper spans for several months but in order to simplify the graphs a focus has been set on several weeks to analyze the different time-series properties. The dataset includes several events like weekends, state and religious holydays and the daylight saving time switch on 25.10.2013. This switch can cause issues with some algorithms and the analysis. A data mining / KDD process methodology has been used to analyze the data as described by Fayyad et.al in [13].

A. Preprocessing

This part of the process reduces the dimensions of the dataset and extracts only the data that is useful to the prediction model. This is also the step where the training/ testing/validation datasets are created. Data preprocessing may include cleaning, normalization, transformation, feature extraction/selection [14]. Also a data warehouse could be constructed as shown in [15] that is used to analyze the web logs. A similar approach could be used for network traffic.

For the purpose of the analysis the pcap file has been preprocessed: features selected, cleaned, normalized and transformed to *CSV*. Than it was imported to a *RDBMS* in order to make fast aggregations and enable filters (groupby/where/partitions) for experiment dataset generation with different parameters.

The features chosen in the feature extraction/selection process where: absolute packet arrival time, packet size, delta time (packet interarrival time) and the protocol. The delta time and protocol features have been chosen because they are needed for a proper traffic generator implementation.

B. Outlier detection

According to [16] an outlier can be defined as an observation that is significantly distanced from all the other observations. It is very important to identify all the outliers and to filter them out because in some cases they can produce massive issues when using statistical learning algorithms.

Outliers can be easily identified on a scatterplot as show in figure 1 for a multiday time series between the 10.10.2013 and 03.11.2013.

Several outlier detection algorithms have been tested including Distance Based Outlier Detection, Density outlier detection and LOF.

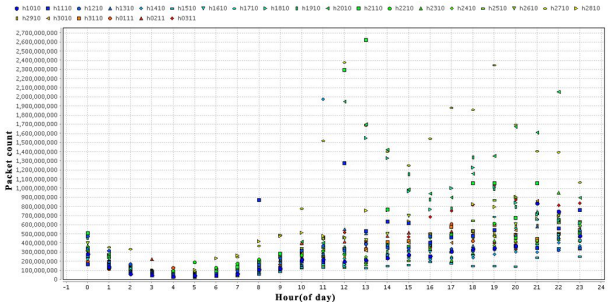

Figure 1 Outliers are visible on a time series scatterplot

Distance based outlier detection [17] searches for ***n*** outliers based on their distance from the ***k*** nearest neighbors (kNN)

Density based outlier detector DB(p,D) searches the global space for outliers. This algorithm takes an assumption that the outliers are those observations that have at least a ***p*** proportion from all other observations that are more distant than ***D***[18]. This algorithm differs from the previous one because it takes all observation into account and not just the k nearest.

The problem of this detector for this specific domain is that we have to specify the ***p*** and ***D*** parameters which is not very practical in context of the linear long time growth of network traffic and time series trends/seasonality. Although this algorithm has shown better results than the previous one by doing a global outlier detection, it is not the best choice in his plain form for this domain because of the parameter estimation problem.

Local outlier factor *(LOF)* ***algorithm*** [19] is similar to the previous ones. The LOF algorithm takes the distance from the k nearest neighbors and estimates the density.

The algorithm compares the local density of the observation with his all neighbor densities. Observations that have a much smaller density than they neighbors are considered outliers.

It's obvious that table 1 – left (distance based outlier detection) shows that there are anomalies at 11 o'clock because it's a big value(1098212) but it's hard to figure out what the anomalies are because we have to guess the number of outliers which is not very suitable for this domain.

hour	outlier	1410
0	false	304371
1	false	177879
2	true	97935
3	true	77836
4	true	63254
5	true	43206
6	true	51625
7	true	87473
8	true	115103
9	false	140210
10	false	184605
11	true	1098212
12	false	261378
13	false	234104
14	false	279164
15	false	275711
16	false	230653
17	false	248877
18	false	260348
19	false	300785
20	false	345683
21	false	326048
22	true	409741
23	true	398130

hour	outlier	1410
0	false	304371
1	false	177879
2	false	97935
3	false	77836
4	false	63254
5	false	43206
6	false	51625
7	false	87473
8	false	115103
9	false	140210
10	false	184605
11	true	1098212
12	false	261378
13	false	234104
14	false	279164
15	false	275711
16	false	230653
17	false	248877
18	false	260348
19	false	300785
20	false	345683
21	false	326048
22	false	409741
23	false	398130

hour	outlier	1410
11	14.972	1098212
22	2.259	409741
23	2.035	398130
20	1.430	345683
5	1.429	43206
21	1.339	326048
6	1.329	51625
1	1.321	177879
4	1.242	63254
10	1.235	184605
3	1.152	77836
9	1.141	140210
7	1.098	87473
8	1.060	115103
19	1.054	300785
2	1.051	97935
16	1.037	230653
0	1.035	304371
13	1.002	234104
14	0.958	279164
17	0.940	248877
15	0.937	275711
12	0.931	261378
18	0.931	260348

Table 1 Left-Distance based outlier detection(with euclide distance); Middle – Density outlier detctor; Right-Local outlier factor

If we set the outlier count parameter ***n*** to 1 the algorithm will successfully detect the big number but it will leave out the others.

Table 1 – middle shows the results of applying the Density based outlier detector. The parameters distance and proportion where set to 7000 and 0.9 respectively. The results are great but the problem is how to define the distance and proportion (the distance parameter is a constant but the problem is the network traffic is growing linear over time[20]).

Table 1 – right shows the results of the LOF algorithm ordered by the outlier factor. We can clearly see that it detected the 11 o'clock anomaly successfully but there is still the threshold problem. Nevertheless it's easier to figure it out than for the previous two methods.

A graphical method could also be used to detect the daily anomalies as seen in figure 2. Its obvious that there are anomalies in the days mentioned before and identified by the outlier detector.

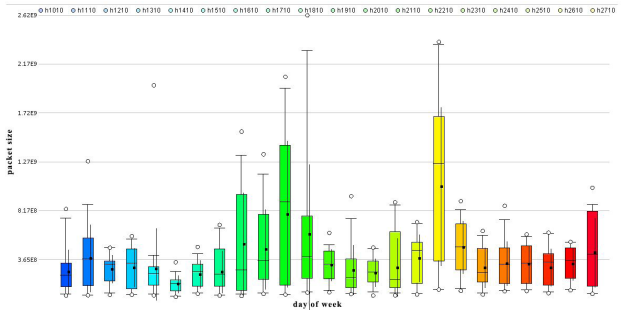

Figure 2 Quartile diagram used for anomaly detection

IV. TIME SERIES ANALYSIS OF THE DATASET

Figure 4(end of paper) shows the packet count rate of internet traffic for three and a half months on daily basis.

From figure 4 it is obvious that the traffic is similar between days and that the daily peek is from 7 till 11 PM while the off peek is from 3 to 5 AM which means that the traffic is mostly from the same time zone. It's also easy to detect the anomalies on the graph.

A. Time series components

To analyze the nature of the time series we have to decompose it to components: the trend component, seasonal component and the noise component.

In figure 5,6,7 we can see the specific components for a 10 days (240 hours) sample of the dataset. In figure 5 we see that the trend is fairly constant except the dropdown on the 100th hour (14.10.2013.) which was a day before a religious holyday (the students go home).

In figure 6 (the seasonal component) we see that the day is made of two periods. The first one from 0 to 50 has four waves, that fits to the previous observation that one day has two parts the (night) off peak and the peak period(day). In figure 7 we can clearly see the anomaly identified before.

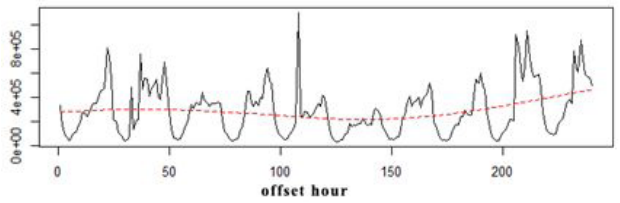

Figure 5 - The trend component

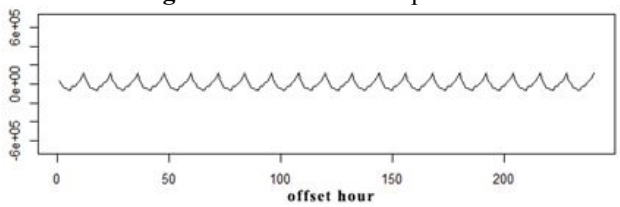

Figure 6 - The seasonal component

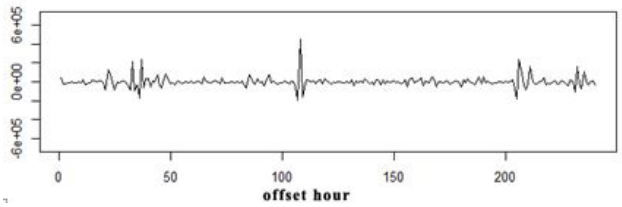

Figure 7 - The noise component

B. The Hurst parameter

By estimating the Hurst exponent we get a value that can indicate the long time memory of the time series [21] it also indicates if the time series is self-similar. An R/S analysis has been used on the given dataset to estimate the Hurst parameter. The results are given in table 2.

	Size	Delta	Count	ALL
Corrected R/ S Hurst exponent:	0.753247	0.6093524	0.7100859	0.9676726
Theoretical Hurst exponent:	0.5457227	0.5457227	0.5457227	0.5334291
Corrected empirical Hurst exponent:	0.7810718	0.4850182	0.6998232	0.9323203
Empirical Hurst exponent:	0.8152428	0.5140385	0.7312095	0.9528319

Table 2 - Hurst exponent estimation

If the Hurst parameter is smaller than 0.5(H<0.5) or greater than 0.5(H>0.5) then this is an indication that there is a trend. If the Hurst parameter is equal to 0.5 the time series is random. It's shown in [22] that time series with a greater value of the Hurst parameter give better results in ANN prediction. The results in table 2 show an H value greater than 0.5 which means there are signs of a trend.

C. Correlation between days

An interesting factor for the time series analysis is the correlation between days in a week as well as between weeks and years. A correlation matrix between days for a 10 days sample is shown in table 3 shown at the end of the paper.

It's easy to spot that the days 14, 21 and 11 (holiday) are anomalies because they have a very low correlation with all the other days.

We also see small anomalies on Friday the 25.10; 18.10 and the Saturday 12.10; 19.10 and 26.10. These are the days when most of the students go out (they are the targeted users of the websites hosted on this server).

day1	day2	corr	day1	day2	corr
h1010	h3010	0.970	h2410	h3010	0.916
h1710	h2510	0.967	h2210	h0311	0.912
h1010	h2310	0.956	h2610	h3010	0.912
h2910	h0311	0.949	h2510	h2910	0.909
h2910	h3010	0.946	h2510	h0311	0.906
h2210	h3110	0.940	h1310	h2610	0.899
h2410	h2910	0.939	h1710	h3010	0.899
h1610	h2610	0.934	h1010	h2910	0.896
h1710	h0311	0.929	h2610	h2910	0.896
h1310	h1610	0.926	h2410	h2610	0.896
h2210	h0111	0.926	h3110	h0111	0.895
h1710	h2910	0.926	h1310	h2010	0.895
h0111	h0311	0.926	h3110	h0311	0.894
h2310	h3010	0.917			

Table 4 Correlation between days sorted

An interesting thing occurred when we analyzed the most correlated days as shown in table 4. Although it's logical that the day of week should have the highest correlation with the same day next week the analysis has

shown different. The days that are near each other have the highest correlation during the week.

A correlation histogram for the days of week has been made as shown in figure 8. The histogram shows that the correlation for the sum of packet size per hour per day of week has a big value (around 0.8) and that the peaks are around the central value of 0.9. This fits a Beta(8,2) distribution.

The results have shown that the analysis for the delta parameter and packet size is very similar to the analysis made for the packet size.

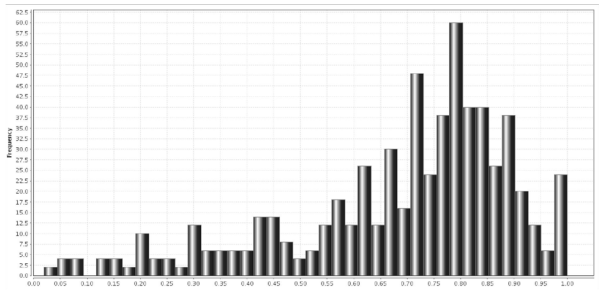

Figure 8 – Histogram - packet size correlation per hour per day per week(vertical axis frequency)

V. UNIVARIATE PREDICTION MODELS

The most popular time series prediction models are ARIMA and Holt Winer for linear prediction and ANNs for nonlinear predictions.

In this experiment we focused on a univariate time series predictions based on individual domain features.

A. Holt Winter

The Holt Winter model is a simple exponential smoothing model for TS prediction. The results of a 24 hour prediction of the time series is shown in figure 9. From the graph it's obvious that the prediction is just an average of the time series.

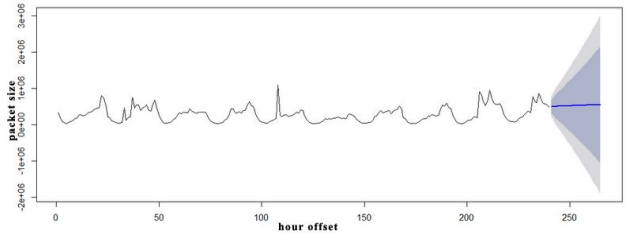

Figure 9 - 24h Time series prediction with Holt Winter

B. ARIMA

Autoregressive integrated moving average has shown good results for self-similarity internet traffic prediction[23] and with predicting time series that show a trend. During the experiment several ARIMA have been used and the model that best fitted the traffic was the ARIMA(2,0,3) model with p=2; d=0 and q=3. The 24 hour prediction on the given dataset is shown in figure 10.

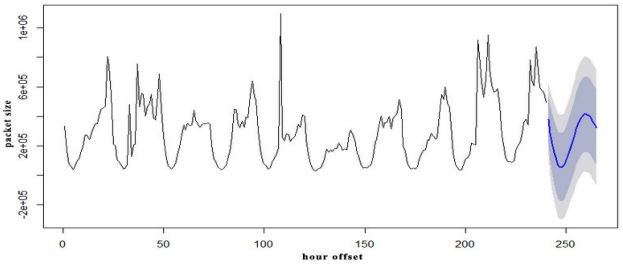

Figure 10 ARIMA(2,0,3) 24h prediction with non-zero mean

The result isn't bad but if we try to expand the prediction to five more days (120 hours) days we get the result shown in figure 11. So ARIMA in it's plain form is good for short term predictions but fails on long term predictions.

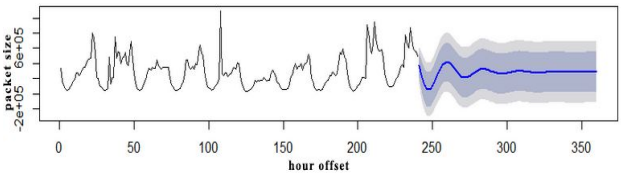

Figure 11 ARIMA(2,0,3) 5 days prediction with non-zero mean

C. ANN

Several experiments[7,11,13] have shown that the prediction with ANNs a nonlinear statistical learning model gives the best prediction results.

In order to analyze the time series a resolution of the time series sampling rate has to be chosen. The ITU rec-E.492 recommends to take a 15 min. or one hour aggregation frame [24]. Our analysis has shown that the best results for this particular dataset and the selected algorithms is a one hour frame.

An ANN with only 22 seasonal lags in the layers and with time series frequency 3 for the window has been used during the experiments. The 24h prediction can been seen in figure 12. As we can see the results are much better fitted than ARIMA so it fits better for the construction of a network traffic generator.

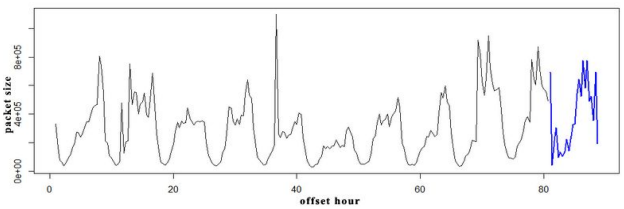

Figure 12 - ANN 24h prediction with 22 seasonal lags and frequency 3

The problem with this approach is that we use only one dimension the packet size and the neural network is adapting to that dimension. One solution is to use an ANN multivariate model in order to do a much longer prediction without the memory effect.

For the second experiment an 27 seasonal lag ANN configuration with 3 days prediction has been used as shown in figure 13.

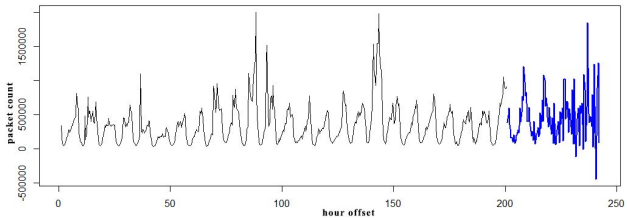

Figure 13 ANN(27,1) 3 days prediction with 27 seasonal lags and frequency 3

VI. RESULTS

A. *Fitting errors and performance*

Several error measurements have been used to evaluate the different models including ME, RMSE, MAE, MPE, MAPE, ACFI. Because of the univariate limitation of this work every dimension of the domain had to be evaluated separately. The fitting error results are shown in table 5,6,7.

	ANN	ARIMA	HoltWinter
ME	-345.6	-246.2	177387.5
RMSE	77451.7	144005.4	333465
MAE	45998.2	79901.6	217246.6
MPE	-9.8043	-17.6479	49.8455
MAPE	18.6512	31.0240	65.7479
MASE	0.25865	0.32807	1.21914
ACF1	0.059133	-0.00108	0.045426

Table 5 - Fitting errors for packet count per hour

	ANN	ARIMA	HoltWinter
ME	-205956	10955995	194601375
RMSE	125857101	226003847	417423245
MAE	65042165	11124690	24943242
MPE	-15.576	-26.010	50.054
MAPE	26.478	43.630	70.327
MASE	0.29317	0.50143	1.113607
ACF1	0.0489	0.0300	-0.0644

Table 6 - Fitting errors for packet size per hour

	ANN	ARIMA	HoltWinter
ME	1.8e-05	3.0e-05	0.00968482
RMSE	0.00403604	0.00656261	0.01967587
MAE	0.00253094	0.0045137	0.01317839
MPE	-4.489	-16.192	47.346
MAPE	16.14	33.22	73.21
MASE	0.21070	0.37578	1.096781
ACF1	-0.02278	0.006316	0.397162

Table 7 - Fitting errors for interarrival delta per hour

The results show that ANNs outperform the other models in model fitting performance but this could be also an indication for overfitting.

The indication that the model fits the data correctly is not the only factor that is important for the prediction.

Cross validation is a statistical method that could help to better evaluate the model but the method is a little bit more complex to be used on time series. A common name in the literature for time series cross validation is "forecast evaluation with rolling origin".

B. *Cross validation*

A cross validation for the time series could be constructed by splitting the dataset into several parts. One part could be the training set (in our case 80% of the dataset) and the other one the validation set (20%).

One could take 80% of the training set and then predict the next 20% and then compare the predicted ones with the real ones from the validation set that wasn't used for the training.

For the verification if the prediction was successful a correlation between the 20% predicted and the 20% validated could be done (better result would be with windowing).

The validation results cross-correlation coefficients are shown in table 8. It is obvious that ANNs outperform the ARIMA model

	Size	Count	Delta
ANN	0.6046795	0.6115911	0.7085994
ARIMA	-0.5489083	-0.3409841	0.7394064
HW	-0.6820613	0.7975048	0.6904266

Table 8 Cross validation 24 hours scope. The values are cross-correlation coefficients

It's interesting that Holt Winter outperforms the ANN and ARIMA, this is due the fact that we use cross-correlation without a time window and the result of the holt winters prediction is a straight line(avg of the time series). This shows the importance to use multiple parameters in order to evaluate the model.

VII. CONCLUSION

In this paper we show that the network traffic time series prediction fits the data mining process model. We have gone through different aspects of the analysis. This is important because most of the papers use traditional process models and small dataset while the network traffic time series prediction problem could be easily classified as a Big Data problem. It was shown that neural network outperform other linear models in the context of precision and that they could be a good model for a network traffic generator based on historical data on top of a data warehouse.

REFERENCES

[1] W. E. Leland, M. S. Taqqu, W. Willinger, D.V. Wilson. 1994. On the self-similar nature of Ethernet traffic (extended version). IEEE/ACM Trans. Netw. 2, 1 (February 1994), 1-15.

[2] A. Feldmann, A.C. Gilbert, and W. Willinger. Data networks as cascades: investigating the multi fractal nature of internet wan traffic. In P roc. ACM SIG COMM, pages 42-55, 1998.

[3] Crovella, M.E.; Bestavros, A., Self-similarity in World Wide Web traffic: evidence and possible causes, Networking, IEEE/ACM Transactions on , vol.5, no.6, pp.835,846, Dec 1997

[4] D. Chakraborty, A. Ashir, T. Suganuma, G. Mansfield Keeni, T. K. Roy, and N. Shiratori. 2004. Self-similar and fractal nature of internet traffic. Int. J. Netw. Manag. 14, 2 (March 2004), 119-129.

[5] Marie, R.R.; Blackledge, J.M.; Helmut E Bez, "On the fractal characteristics of Internet network traffic and its utilization in covert communications," Internet Technology and Secured

Transactions, 2009. ICITST 2009. International Conference for , vol., no., pp.1,6, 9-12 Nov. 2009

[6] Dharmadhikari, V. B.; Gavade, J. D., An NN approach for MPEG video traffic prediction, Software Technology and Engineering (ICSTE), 2010 2nd International Conference on , vol.1, no., pp.V1-57,V1-61, 3-5 Oct. 2010

[7] G.Peter Zhang, Time series forecasting using a hybrid ARIMA and neural network model, Neurocomputing, Volume 50, January 2003, Pages 159–175

[8] S Jung, C Kim, Y Chung, A Prediction Method of Network Traffic Using Time Series Models,Computational Science and Its Applications - ICCSA 2006

[9] X. Mu, N. Tang, W. Gao, L. Li, Y. Zhou, A One-Step Network Traffic Prediction, Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, Lecture Notes in Computer Science Volume 5227, 2008, pp 616-621

[10] Weidong Luo, Xingwei Liu, Jian Zhang, SVM-based analysis and prediction on network traffic, ISKE-2007 Proceedings, 2007

[11] Cortez, P., Rio, M., Rocha, M. and Sousa, P. (2012), Multi-scale Internet traffic forecasting using neural networks and time series methods. Expert Systems, 29: 143–155. doi: 10.1111/j.1468-0394.2010.00568.x

[12] Dataset URL: http://etf.ba/netdataset

[13] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, From data mining to knowledge discovery: an overview., Advances in knowledge discovery and data mining., 1996

[14] Kotsiantis, D. Kanellopoulos, P. Pintelas, Data Preprocessing for Supervised Leaning, International Journal of Computer Science, 2006, Vol 1 N. 2, pp 111–117

[15] K. P. Joshi, A. Joshi, Y. Yesha, On Using a Warehouse to Analyze Web Log, Distributed and Parallel Databases March 2003, Volume 13, Issue 2, pp 161-180

[16] F.E. Grubbs, Procedures for detecting outlying observations in samples, Technometrics 11 (1): 1–21, February 1969

[17] S. Ramaswamy, R. Rastogi, and K.Shim, Efficient algorithms for mining outliers from large data sets. SIGMOD Rec. 29, May 2000

[18] E. Knorr, R. Ng: Algorithms for Mining Distancebased Outliers in Large Datasets , VLDB, 1998, pp392 - 403

[19] V.A., Local outlier factor, Wikipedia / Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; Sander, J. (2000). "LOF: Identifying Density-based Local Outliers". Proceedings of the 2000 ACM SIGMOD international conference on Management of data. SIGMOD '00: 93–104.

[20] Gardner, E. S., Jr. "Exponential smoothing: The state of the art." Journal of Forecasting 4, 1985, 1–28

[21] H.E. Hurst, 1951, Long-term storage of reservoirs: an experimental study, Transactions of the American Society of Civil Engineers, Vol. 116, pp. 770-799.

[22] B. Qian, K. Rasheed, 2004, “Hurst Exponent and financial market predictability,” IASTED conference on “Financial Engineering and Applications”(FEA 2004), pp. 203-209,

[23] R.F.Nau, “Introduction to ARIMA”, Duke University Lectures

[24] ITU, “Recommendation E.492 Traffic reference period”, ITU, 1996

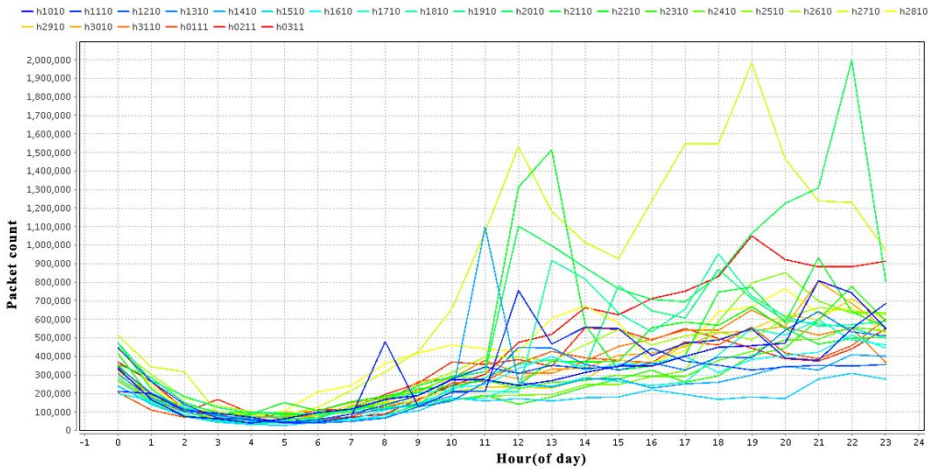

Figure 4 - Packet count aggregation based on hour basis the lines show the different days

Attributes	h1010	h1110	h1210	h1310	h1410	h1510	h1610	h1710	h1810	h1910	h2010	h2110	h2210	h2310	h2410	h2510	h2610
h1010	1	0.353	0.569	0.771	0.196	0.866	0.834	0.846	0.476	0.592	0.720	0.305	0.669	0.956	0.877	0.781	0.843
h1110	0.353	1	0.433	0.634	0.068	0.422	0.568	0.422	0.420	0.398	0.672	0.621	0.492	0.320	0.441	0.309	0.547
h1210	0.569	0.433	1	0.790	0.369	0.782	0.771	0.555	0.605	0.623	0.620	0.466	0.746	0.540	0.727	0.481	0.813
h1310	0.771	0.634	0.790	1	0.250	0.785	0.926	0.767	0.672	0.623	0.895	0.706	0.749	0.720	0.814	0.681	0.899
h1410	0.196	0.068	0.369	0.250	1	0.235	0.256	0.131	0.020	0.082	0.119	0.066	0.210	0.165	0.290	0.093	0.319
h1510	0.866	0.422	0.782	0.785	0.235	1	0.862	0.676	0.461	0.579	0.664	0.292	0.683	0.887	0.845	0.623	0.853
h1610	0.834	0.568	0.771	0.926	0.256	0.862	1	0.799	0.720	0.678	0.884	0.571	0.828	0.822	0.884	0.727	0.934
h1710	0.846	0.422	0.555	0.767	0.131	0.676	0.799	1	0.614	0.732	0.765	0.406	0.812	0.823	0.884	0.967	0.803
h1810	0.476	0.420	0.605	0.672	0.020	0.461	0.720	0.614	1	0.722	0.722	0.569	0.798	0.425	0.575	0.557	0.699
h1910	0.592	0.398	0.623	0.623	0.082	0.579	0.678	0.732	0.722	1	0.611	0.303	0.820	0.565	0.669	0.723	0.783
h2010	0.720	0.672	0.620	0.895	0.119	0.664	0.884	0.765	0.722	0.611	1	0.757	0.733	0.679	0.728	0.668	0.820
h2110	0.305	0.621	0.466	0.706	0.066	0.292	0.571	0.406	0.569	0.303	0.757	1	0.451	0.202	0.361	0.283	0.445
h2210	0.669	0.492	0.746	0.749	0.210	0.683	0.828	0.812	0.798	0.820	0.733	0.451	1	0.651	0.816	0.776	0.820
h2310	0.956	0.320	0.540	0.720	0.165	0.887	0.822	0.823	0.425	0.565	0.679	0.202	0.651	1	0.870	0.793	0.808
h2410	0.877	0.441	0.727	0.814	0.290	0.845	0.884	0.884	0.575	0.669	0.728	0.361	0.816	0.870	1	0.850	0.896
h2510	0.781	0.309	0.481	0.681	0.093	0.623	0.727	0.967	0.557	0.723	0.668	0.283	0.776	0.793	0.850	1	0.724
h2610	0.843	0.547	0.813	0.899	0.319	0.853	0.934	0.803	0.699	0.783	0.820	0.445	0.820	0.808	0.896	0.724	1
h2710	0.538	0.669	0.695	0.803	0.340	0.535	0.791	0.709	0.719	0.722	0.849	0.709	0.842	0.457	0.663	0.620	0.768
h2810	0.648	0.519	0.689	0.758	0.215	0.591	0.777	0.802	0.797	0.711	0.793	0.562	0.842	0.608	0.839	0.749	0.822
h2910	0.896	0.463	0.629	0.804	0.195	0.807	0.885	0.926	0.622	0.767	0.769	0.326	0.855	0.878	0.939	0.909	0.896
h3010	0.970	0.415	0.651	0.828	0.232	0.835	0.880	0.899	0.608	0.724	0.790	0.375	0.790	0.917	0.916	0.838	0.912
h3110	0.690	0.453	0.801	0.761	0.379	0.719	0.831	0.778	0.745	0.846	0.744	0.411	0.940	0.645	0.803	0.729	0.873
h0111	0.700	0.589	0.675	0.802	0.184	0.686	0.886	0.809	0.796	0.756	0.789	0.524	0.926	0.650	0.807	0.753	0.831
h0211	0.611	0.598	0.798	0.780	0.299	0.744	0.839	0.608	0.722	0.746	0.700	0.424	0.819	0.570	0.758	0.532	0.874
h0311	0.797	0.509	0.667	0.829	0.145	0.738	0.880	0.929	0.707	0.838	0.804	0.446	0.912	0.775	0.865	0.906	0.860

Table 3 - Aggregate packet size per hour: Correlation matrix