Evaluation of link bandwidth using Levenshtein distance

Dragan Jevtic and Sanja Jevtic

***Abstract* — This paper aims to review the activity of the computer network, i.e., the bandwidth and to observe deviations from the usual values for a certain predefined period. For this purpose, the bandwidth was taken for one working day a week. All Mondays in one month were considered. The bandwidth is considered and the Levenshtein distance is calculated and discussed.**

***Key words* — bandwidth, Levenshtein distance.**

I. Introduction

Bandwidth monitoring in computer networks is an activity that serves to analyze and search for traffic that may be caused by viruses or dangerous activities. These activities, with time, became very important for business and business processes as well as for saving business data. There is various software that analyze the bandwidth and alarms if the normal values are exceeded, which can often be a trigger for noting the irregular actions.

As it is stated in abstract the aim of this paper is to review deviations from the usual values. It is assumed that the analysis can be done for a smaller time range as well as for near-real-time recording. One working day in a week is chosen (Monday) and was reviewed for the whole month.

The usual max/min comparison of data sets is a main stream approach used as a first angle for data set evaluation. The novelty in this paper is a process of assigning a letter (letter should be treated as a level or grade of bandwidth) to bandwidth as further treated these letters as strings. Levenshtein distance [1] is used to compare these strings among themselves comparing them as usual text. Being the means for comparison of two strings (of letters), Levenshtein distance is widely used [1-5]. It shows how similar two string are, and represent a minimum number of operations required to make the two inputs equal. But, in this paper no tasks like tokenization, lemmatization/ stemming, sentence segmentation are envisaged.

II. Materials and Methods

For every Monday, a graph is defined that represents the bandwidth, that is, the amount of transferred data at a given time. Orange data modeling software [6] was used to analyze and display the corresponding bandwidths per unit of time. The bandwidth is represented per minute for 24 h. Figures 1 to 5 show the bandwidths, by consecutive Mondays based on the bandwidth data collected from the PRTG program [7], for the selected days.

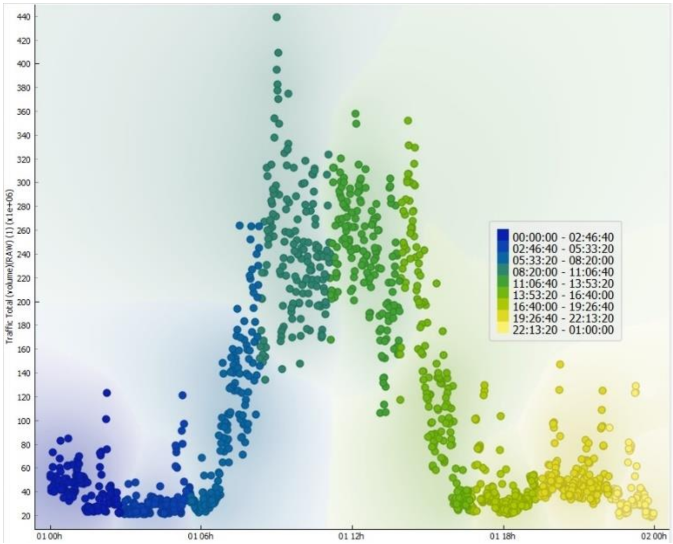
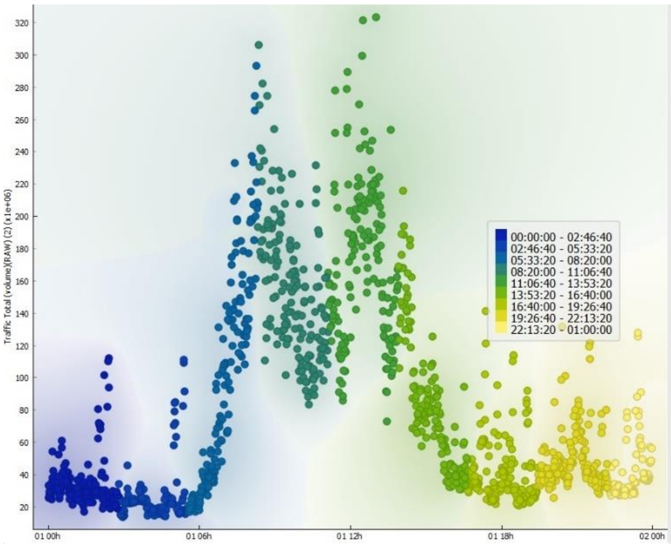


Fig. 1. Bandwidth in time (24h/1min) for the first Monday.



Dragan Jevtic – Infrastructure of Serbian Railways, Nemanjina 6, 11 000 Belgrade, Serbia; University of Criminal Investigation and Police Studies, Cara Dusana 196, 11 080 Belgrade, Serbia (e-mail: dragan.jevtic@srbrail.rs, jevtic.dragan@gmail.com).

Sanja Jevtic – Academy of Technical and Art Applied Studies Belgrade, Zdravka Celara 14, 11000 Belgrade, Serbia (e-mail: sanja.jevtic@vzs.edu.rs, jevtic.sanja@gmail.com).

Fig.2. Bandwidth in time (24h/1min) for the second Monday.

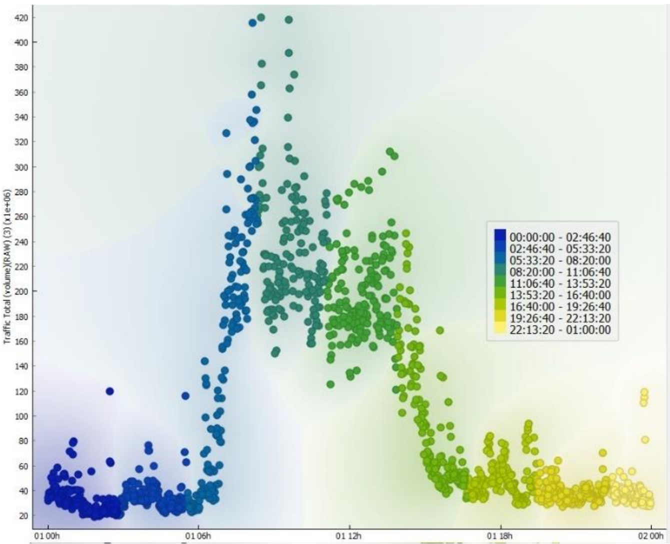


Fig.3. Bandwidth in time (24h/1min) for the third Monday.

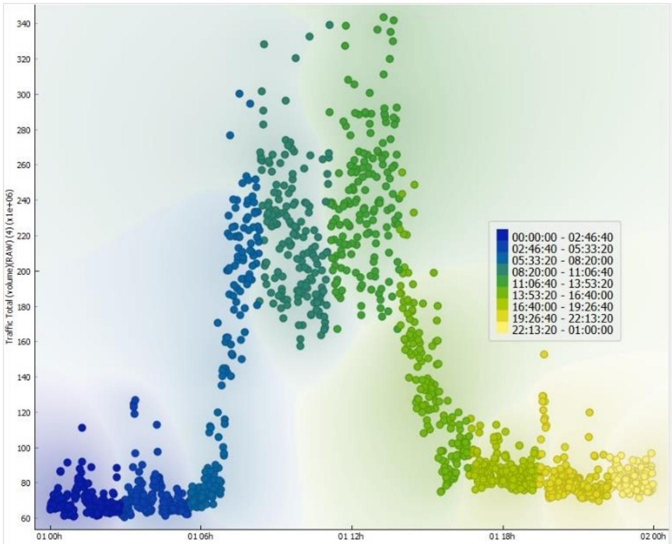


Fig.4. Bandwidth in time (24h/1min) for the fourth Monday.

The next step in the analysis is the search for local minima and maxima for defined graphics, i.e., functions. Local minima and maxima are calculated via the first derivatives of the functions.

The function *f* has a local maximum at the point x_0 if there exists a neighborhood $U(x_0) \subset A$ such that $f(x) \leqslant f(x_0)$, $x \in U(x_0)$ if $f(x) < f(x_0)$, $x \in U(x_0) \setminus \{x_0\}$ then at the point x_0 the function *f* has a local maximum.

The function *f* has a local minimum at the point x_0 if there is a neighborhood $U(x_0) \subset A$ such that $f(x) \geqslant f(x_0)$, $x \in U(x_0)$ if $f(x) > f(x_0)$, $x \in U(x_0) \setminus \{x_0\}$ then at the point x_0 the function *f* has a local maximum.

The other approach to evaluation of bandwidth is comparison of quantized bandwidth. This seams as a good method for comparison of two strings thus giving us a useful tool for determining if two data sets are similar and the measure of this similarity.

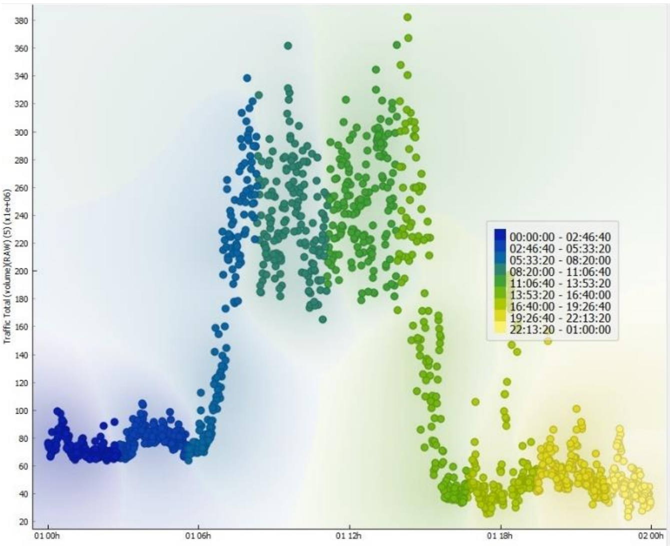


Fig.5. Bandwidth in time (24h/1min) for the fifth Monday.

Used with text it points out if the different strings refer to the same entity (coreference). On the other hand, edit distance also quantify similarity of the string (minimal edit distance between two strings is defined as the number of editing operations).

III. Experiment and Results

We use the definition of local extremes (minimum and maximum) in the tables with the values 0, for the local minimum, and 1, for the local maximum. The last, final stage of comparing the bandwidths of five consecutive Mondays is to compare the local extremes via string conversion and using the Levenshtein string distance. Levenstein's distance of strings refers to the mutual comparison of two strings (in our case each with each) and looking at the transformation of one string into another, i.e., how many operations are necessary to perform in order to transform one string into another. The smaller the minimum distance between two strings, the more similar the strings are. In this process, five different strings are formed.

Trough evaluating local maxima and minima at time coordinates it was chosen to make strings from their consecutive appearance, no matter the time stamp. These strings are presented in Table I. Only first 15 elements in strings are presented because of the length of the strings (781, 810, 842 833,812 for consecutive Mondays).

Another string evaluation was made. Strings are formed based on the bandwidth per minute and each value was assigned a letter Table II, Table III.

The first 15 minutes converted to strings, for all 5 consecutive Mondays, with the names APon, BPon, CPon, DPon and EPon were shown (Table III). Strings in full have 1439 elements each.

 ISBN 978-86-7466-969-3

TABLE I
STRINGS FORMED FROM CONSECUTIVE MAXIMA AND MINIMA (0-MINIMA, 1-MAXIMA)

Strings formed for consecutive maxima and minima for 5 Mondays				
APon	BPon	CPon	DPon	EPon
1	1	1	0	0
0	0	0	1	1
1	1	1	0	0
0	0	0	1	1
1	1	1	0	0
0	0	0	1	1
1	1	1	0	0
0	0	0	1	1
1	1	1	0	0
0	0	0	1	1
1	1	1	0	0
0	0	0	1	1
1	1	1	0	0
0	0	0	1	1
1	1	1	0	0

TABLE II
RULE FOR ASSIGNING LETTERS TO BANDWIDTH

Bandwidth B Traffic total ($x10^6$)	Assigned letter
B < 50	A
50 ≤ B < 100	B
100 ≤ B < 150	C
150 ≤ B < 200	D
200 ≤ B < 250	E
250 ≤ B < 300	F
300 ≤ B < 350	G
350 ≤ B < 400	H
400 ≤ B < 450	I

In case of calculating Levenshtein distance between strings there are two steps. First step is forming a matrix *(m+1) x (n+1)*, where *m* and *n* are lengths of two strings. Then the minimum edit distance is calculated between the elements of the strings. Insertion and deletion have a value of 1 and substitution a value of 1 or 2 (1) [1].

$$(\forall\, 1 \le i \le m,\ 1 \le j \le n)$$
$$D(\mathrm{i},\mathrm{j}) = min\begin{cases} D(i-1,j)+1 \\ D(i,j-1)+1 \\ D(i-1,j-1)+\begin{cases}1\ or\ 2, substitution\ diff.char. \\ 0,\ identical\ characters\end{cases}\end{cases} \tag{1}$$

The value of the element D(m,n) is the Levenshtein distance between the strings *s1* and *s2* [8].

In this paper a Python code was used to generate the Levenshtein distance. The existing code example was used to define the distance between the strings using the value of 1 for substitution [9].

As a result of the program code, the following reports were generated, presented in the form of matrix, Table IV and Table V. Table IV presents Levenshtein distances for the first method of evaluation (max-min) and in Table V distances for the second method (quantized levels). The distances are symmetrical.

TABLE III
STRING FORMED FOR 5 CONSECUTIVE MONDAYS, ASSIGNING A LETTER FOR BANDWIDTH PER MINUTE

Time	Strings formed for consecutive Mondays for first 15 minutes in a day				
0:00:00	APon	BPon	CPon	DPon	EPon
0:01:00	B	A	A	B	B
0:02:00	B	A	A	B	B
0:03:00	A	A	A	B	B
0:04:00	A	A	A	B	B
0:05:00	A	A	A	B	B
0:06:00	B	A	A	B	B
0:07:00	B	A	A	B	B
0:08:00	A	A	B	B	B
0:09:00	A	A	A	B	B
0:10:00	A	B	A	B	B
0:11:00	A	A	A	B	B
0:12:00	A	A	A	B	B
0:13:00	A	A	A	B	B
0:14:00	B	A	B	B	B
0:15:00	A	A	A	B	B

TABLE IV
LEVENSHTEIN DISTANCES BETWEEN STRINGS FORMED FOR MAXIMA (1) AND MINIMA (0)

Levenshtein distances between strings	APon	BPon	CPon	DPon	EPon
APon		29	52	61	31
BPon			23	32	2
CPon				9	21
DPon					30
EPon					

TABLE V
LEVENSTEIN DISTANCES BETWEEN STRINGS FORMED FOR QUANTIZED LEVELS OF BANDWIDTH

Levenshtein distances between strings	APon	BPon	CPon	DPon	EPon
APon		583	542	1071	829
BPon			539	1118	969
CPon				1053	903
DPon					637
EPon					

 ISBN 978-86-7466-969-3

IV. Discussion and Conclusion

Two evaluations based on the strings were performed. First was driven by engineering mindset, looking for maxima and minima. The second one was based on treating the amount of bandwidth as a letter i.e., sort of quantization.

After the first string processing method was used, we conclude that minimal edit distances are in the range of 9-61.

Second method of string evaluation made for the levels of bandwidth (quantization) did give much greater differences.

Although maxima and minima in bandwidth curves did not have the same time stamps in different Mondays, this method showed smaller differences among the data for consecutive Mondays. The other method gave lager differences and an assumption could be made that this method could be used for further *hunting* of Mondays with different bandwidths i.e., anomalies in bandwidth.

The third type of string could be formed and comparison to obtained data could be made. This third process could treat increase/decrease of bandwidth in present time moment compared to previous moment as an +1/-1.

Forming the words of strings is yet to be adopted for this kind of evaluation, if or not it has any meaning.

This method (using of Levenshtein) has yet to be tested in real-time environment and with multiple scenarios of change in bandwidth.

Thus, we conclude that anomalies in bandwidth could be detected using this method. This method could also help in detecting deviations in data transmission for chosen time periods.

References

.
[1] Daniel Jurafsky & James H. Martin, *Speech and Language Processing*. Stanford, 2023, https://web.stanford.edu/~jurafsky/slp3/
[2] Riesen, K., Schmidt, R. "Online signature verification based on string edit distance". IJDAR 22, 41–54, 2019
[3] C. Vielhauer, S. Schimke and J. Dittmann, "Using Adapted Levenshtein Distance for On-Line Signature Authentication," in Pattern Recognition, International Conference on, Cambridge UK, pp. 931-934, 2004 doi: 10.1109/ICPR.2004.1334412
[4] S. Chowdhury, U. Bhattacharya and S. Parui, "Online Handwriting Recognition Using Levenshtein Distance Metric," in 2013 12th International Conference on Document Analysis and Recognition (ICDAR), Washington, DC, USA, pp. 79-83. 2013 doi: 10.1109/ICDAR.2013.24
[5] Milan Gnjatović, Vojkan Nikolić, Dušan Joksimović, Nemanja Maček, Nebojša Budimirović "An approach to human activity clustering using inertial measurement data", in Proc. of the Archibald Reiss Days 2020, vol. 10, pp. 547-555, 2020,
[6] https://orangedatamining.com/
[7] https://www.paessler.com/bandwidth_monitoring
[8] M.Gnjatović, *Uvod u pronalaženje informacija na vebu,* VIŠER, 2017, Beograd, Srbija. 2017.
[9] https://python-course.eu/applications-python/levenshtein-distance.php

Apstrakt

Ovaj rad ima za cilj da sagleda aktivnost računarske mreže, odnosno propusni opseg i da uoči odstupanja od uobičajenih vrednosti za određeni unapred definisani period. U tu svrhu uzet je propusni opseg za jedan radni dan u nedelji. Smatrani su svi ponedeljci u jednom mesecu. Razmatra se širina opsega i izračunava se i raspravlja o Levenštajnovom rastojanju.

Procena propusnog opsega linka Levenštajnovim rastojanjem

Dragan Jevtić i Sanja Jevtić

 ISBN 978-86-7466-969-3