describe how close are the reliability evolution paths of each software instance. This may reveal hidden topological similarities between software products and help us to characterise its reliability behaviour. In this preliminary study we illustrated the idea how we can use topological data analysis for that purpose that give us some promising results that is motivation for our future work.

In future work we want to compare obtained distances between projects to the traditional reliability growth models [14, 16, 13]. From this future work we expect to gain deeper understanding of different reliability behaviour among software products and relate it to theory of reliability growth modelling. This knowledge may be useful in early characterisation of software system reliability behaviour that can lead to early determination of the best reliability growth modeling approach. As we show from analysed examples of software systems this approach may be very useful to provide a measure of development environment with respect to various system properties. This information may be very useful in early determination of the best mathematical model for system behaviour.

The choice of distance used to construct the simplicial complexes that originate the persistence diagrams can have influence on the information extracted from them. This is a topic of discussion in further work, where we shall also study in depth the interpretation of the information provided by the persistence diagram in the context of the original problem.

Moreover, the study of the dynamics of the reliability evolution itself, and the identification of patterns, can be accessed by these methods and will also be subject of further research. Also the distinctions between the different levels of persistence landscapes can complement the comparison information provided, and thus contribute to this research.

6. ACKNOWLEDGMENTS

The authors would like to thank to Primož Škraba for the discussions on the applications of topological data analysis to time series data; to Mateusz Juda for his support with the computation of the bottleneck distance between diagrams, to the funding of the EU project TOPOSYS (FP7-ICT-318493- STREP) of the first author, and to the support of Croatian Science Foundation's funding of the project EVOSOFT (UIP-2014-09-7945) and by the University of Rijeka Research Grant 13.09.2.2.16 funding of the second author.

7. REFERENCES

[1] C. Andersson. A replicated empirical study of a selection method for software reliability growth models. *Empirical Softw. Engg.*, 12(2):161–182, Apr. 2007.

[2] G. Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[3] J. P. Costa and P. Škraba. A topological data analysis approach to the epidemiology of influenza. In *SIKDD15 Conference Proceedings*, 2015.

[4] V. de Silva, P. Škraba, and M. Vejdemo-Johansson. Topological analysis of recurrent systems. *Workshop on Algebraic Topology and Machine Learning, NIPS*, 2012.

[5] H. Edelsbrunner and J. Harer. Persistent homology-a survey. *Contemporary mathematics*, 453:257–282, 2008.

[6] B. T. Fasy, J. Kim, F. Lecci, and C. Maria. Introduction to the R package TDA. *arXiv:1411.1830*, 2014.

[7] B. T. Fasy, J. Kim, F. Lecci, C. Maria, and V. Rouvreau. TDA package for R. *cran.r-project.org/web/packages/TDA, Accessed 18/12/2016.*

[8] T. Galinac and S. Golubić. Project overlapping and its influence on the product quality. In *Proceedings of the 8th International Conference on Telecommunications, 2005. ConTEL 2005.*, volume 2, pages 655–662, June 2005.

[9] T. Galinac Grbac and D. Huljenić. On the probability distribution of faults in complex software systems. *Information and Software Technology*, 58:250–258, 2015.

[10] T. Galinac Grbac, P. Runeson, and D. Huljenić. A second replicated quantitative analysis of fault distributions in complex software systems. *IEEE Transactions on Software Engineering*, 39(4):462–476, 2013.

[11] R. Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

[12] D. R. Jeske and X. Zhang. Some successful approaches to software reliability modeling in industry. *J. Syst. Softw.*, 74(1):85–99, Jan. 2005.

[13] D. Kececioglu, editor. *Reliability engineering handbook, vol. 2*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1991.

[14] M. R. Lyu, editor. *Handbook of Software Reliability Engineering*. McGraw-Hill, Inc., Hightstown, NJ, USA, 1996.

[15] K. Mischaikow and V. Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete & Computational Geometry*, 50(2):330–353, 2013.

[16] J. D. Musa. *Software Reliability Engineering: More Reliable Software Faster and Cheaper*. Authorhouse, 2004.

[17] V. Nanda. Perseus, the persistent homology software. *http://www.sas.upenn.edu/ vnanda/perseus, Accessed 18/12/2016.*

[18] J. A. Perea and J. Harer. Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, 15(3):799–838, 2015.

[19] J. A. Perea et al. Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC bioinformatics*, 16(1):257, 2013.

[20] C. Stringfellow and A. A. Andrews. An empirical method for selecting software reliability growth models. *Empirical Softw. Engg.*, 7(4):319–343, Dec. 2002.

[21] A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.