The Topological Data Analysis of Time Series Failure Data in Software Evolution

Joao Pita Costa
Institute Jozef Stefan
joao.pitacosta@ijs.si

Tihana Galinac Grbac
Faculty of Engineering
University of Rijeka
tihana.galinac@riteh.hr

ABSTRACT

As evolving complex systems have become the central part of almost every human activity, their reliability is the key property for their successful application, especially with the emerging Internet of Services concept. There are many quantitative mathematical models, the so called reliability growth models, aiming to predict and estimate reliability of software systems based on the failure count time series. This paper suggests a novel and still unexplored qualitative approach to understand failure time series studying its topological features and their influence on failure distributions, thus affecting mission critical system properties, among which is reliability. To illustrate the new ideas, we analyse here the time series failure data of evolving software systems across the system versions for two open source software systems and one mission critical industrial software system, and discuss their topological relations and behaviour. We conclude that topological analysis might be useful for characterising software system behaviour early enough and for early characterization of system reliability that may contribute to software reliability modeling.

CCS Concepts

•**Software and its engineering** → **Software reliability;**
•**Networks** → *Network reliability;*

Keywords

Topological Data Analysis, Software Structure Evolution, System Defectiveness

1. INTRODUCTION

Evolving complex software systems (EVOSOFT) have become a central part of a rapidly growing range of applications, products and services supporting daily human activities from all economic sectors. These systems have been evolving gradually during the sequence of system releases. As these systems are often distributed, heterogeneous, decentralised and inter-dependent, and operating in dynamic and unpredictable environments, availability and reliability become key properties for its operation and future evolution. There are numerous models to model software reliability [14, 13, 16]. In the traditional EVOSOFT engineering the problem of early verification planning requires accurate predictions early enough. Tools and models used to simulate and predict reliability growth during verification process are extensively used in industrial practice to predict effort and time needed to bring the system to required reliability state [12]. Unfortunately, these models are not applicable in early stages of the verification process and usually it is a problem to identify the right model for accurate model and prediction [20, 1].

Modern development trends have changed this traditional system evolution principle by introducing new system development ecoenvironments, such as for example Internet of Services (IoS), where end user services are provided by chaining evolving services within these ecoenvironments to accommodate user needs. Reliability of such ecoenvironments remains the key issue and may influence the user satisfaction and the choice of service providers. Hence, the need for early determination of key reliability properties is getting even more important in the emerging service oriented context.

In this paper we approach the software reliability from an entirely novel perspective. All reliability models in the existing literature [14, 13, 16] consider the reliability by studying the quantitative properties of the failure time series. Their goal is to predict and estimate the quantitative behaviour of these time series. Our approach based on the topological data analysis is different.

Topological data analysis is a recent data science approach that looks into the insights on the data provided by its shape. It enables powerful tools to analyse time series data, within a wide variety of applications from gene expression data [19] to spreading of diseases [3]. In this paper we apply persistent homology, one of the topological data analysis methods, to study EVOSOFT reliability behaviour. Topological data analysis have been successfully used to model behaviour of wide variety of complex systems. Persistent homology provides mathematical algorithms that apply tools of computational topology to datasets where topology is derived from the connectivity information in dataset. Topological features are identified at different structure resolutions and the most persistent ones along the resolution space are said to be representative features of underlying space. We investigate

ICPE '17 Companion, April 22-26, 2017, L'Aquila, Italy
 ISBN 978-1-4503-4899-7/17/04…$15.00
DOI: http://dx.doi.org/10.1145/3053600.3053604

existence of these features in system reliability datasets and their relation to other system reliability properties as modeled by existing software reliability models.

The novel and still unexplored area of research addressed in this paper is to understand how topological features of failure time series data are related to system reliability. We study the failure count time series topologically, that is, we study its shape and qualitative properties regardless of the actual absolute values. In this way, we may find some hidden topological similarities between reliability properties of software systems whose failure counts (and other system properties) may be quantitatively very different. This may be useful, for example, in determination of reliability growth model family for particular software system.

The first author is currently working on topological epidemiological data analysis. The aim of his ongoing research is to model spreading behaviour of diseases (e.g. influenza) with the help of topological data analysis. The second author is working on modelling the behaviour of complex software systems. The aim of their research within the ongoing projects is to empirically and analytically investigate behaviour of complex software systems and develop solid ground for their simulation and modelling of their behaviour [10, 9]. Here, we analyse time series data based on software failures in the several releases of two open source software products from Eclipse community (JDT and PDE) and several releases of an industrial system from telecommunication core network product provider (MSC). The discussion in this paper reveals new opportunities for research in two directions. One is in extending the application of the underlying mathematical models to explain the behaviour of complex software systems and compare so obtained reliability properties with existing reliability models. The other is to find in that use-case new motivations to discuss the topological analysis of time series data. This paper will discuss the main research goals being: (i) to explore application of persistent homology on reliability behaviour of software system; (ii) to observe topology similarities with other complex systems.

2. THE PROBLEM, THE DATA AND THE APPROACH

One of the key system properties for mission critical software systems is reliability. Roughly speaking, reliability of a system is a measure of probability of its failure in time. It is usually assessed using the fault or failure count over time. There are various analytical models of reliability, the so-called reliability growth models (RGM). In many cases they are applied to predict failures of software systems in operation, as well as the remaining number of faults in the system during its development. The latter is of great importance for software development companies, because the correct prediction of remaining faults in the system can significantly reduce cost of testing and verification effort without risking the expensive failures during operation at the customer's site.

In applications of RGM in practice, there are two main difficulties that should be addressed. The first difficulty is the choice of the family of models which to apply. This is quite often a serious issue even for systems that are not evolutionary developed. Since reliability modeling is used for verification planning it is very important to determine the right model early enough within development process, while we can still act cost effectively to verification activities. The second difficulty is the possible change of models during evolution, that is, from release to release of a software system. Although the software system in evolution is only changing partially, it is often the case that the properties of software, such as quality or reliability, change from release to release. The reasons for that are in many factors that influence the software properties, which may be all referred to as environmental factors (e.g. development organization, software structure, size change). Hence, it is quite rare that the reliability models can be directly applied from release to release. The idea behind this work is that it is quite possible that, although the quantitative values of the reliability measures may vary during evolution, their shape may be preserved. This hidden shape of reliability data is studied using the recent topological method in data science, the so-called topological data analysis. Topology is the field of mathematics dealing with qualitative, instead of quantitative phenomena, in particular, it is concerned with shape of objects rather than their measures. Therefore, here we want to explore if we can use topological data analysis to understand a system failure time series. This may be helpful for identifying some hidden similarities between software systems failure time series and these may be further used for characterizing their reliability behaviour. In the next section we present toplogical data analysis measures that we use to find out if we can find these hidden similarities by using these measures in the failure time series.

The purpose of this preliminary study is to assess the feasibility of this approach by considering three evolutionary developed software systems; two open source and one closed source. We used the failure data reported in project failure repositories. The failure data were counted within the one week period, with removed duplicates, and represented as time series in weeks as is presented in Figure 1. The first two figures represent time series of open source Eclipse projects, Java Development tools (JDT) and Plug-in Development Environment (PDE) respectively, for which we collected data from the Bugzilla repository. The last one is a dataset from a mission critical system in telecommunication network, application software of Mobile Switching Centre (MSC) from an Ericsson core network product line [8]. We choose industrial and open source projects with the aim to understand if there are differences in their topological behaviour, as one may expect, since this difference has been reported in many other empirical studies. These development environments are very much different and in most of the studies is mentioned as cause of different results stressing that same models may not be applicable for both environments. From the figure their different behaviour is visible, although we may not conclude that open source projects are more similar to each other than to the industrial project. We have chosen three projects with quite different failure fault time series with the intention to compare their topology.

3. ENCODING THE TOPOLOGICAL INFORMATION

Topological Data Analysis (TDA) applies the qualitative methods of topology to problems of machine learning, data mining and computer vision. Persistent homology identifies a global structure by inferring high-dimensional structure from low-dimensional representations and studying proper-

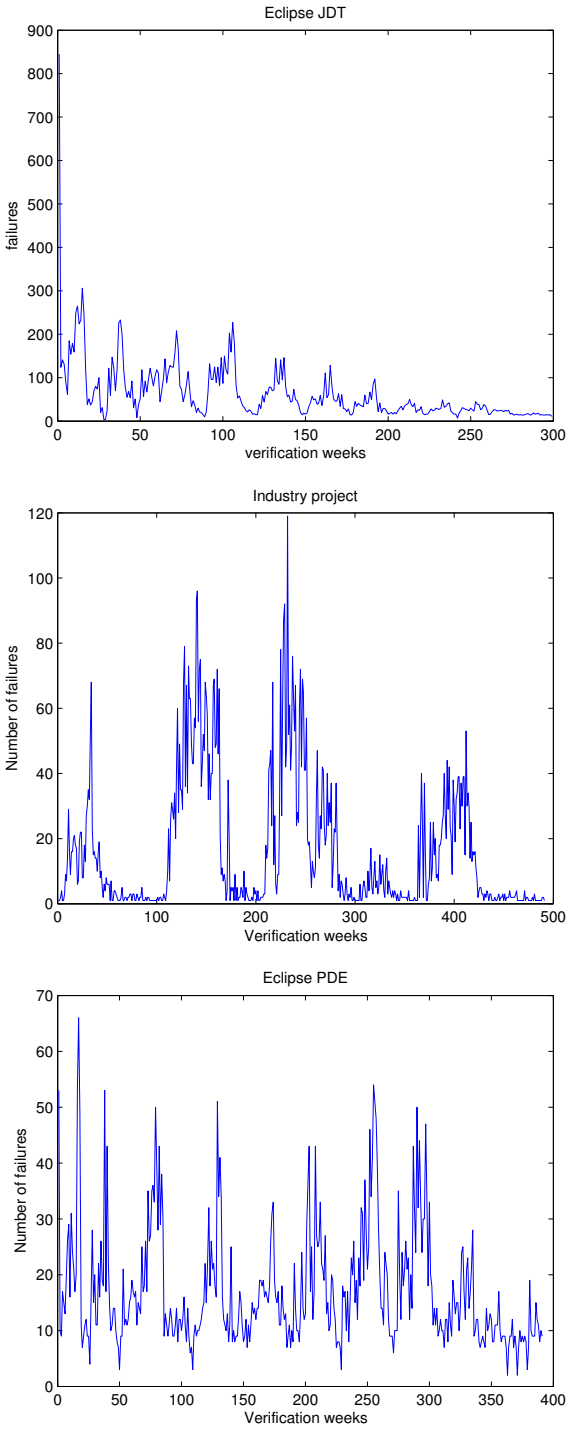


Figure 1: Time series for failure data in three software projects: the open source JDT (above) and PDE (below), and the industrial MSC (center).

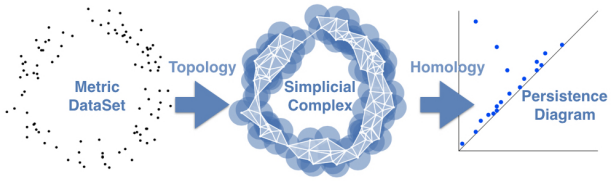


Figure 2: Encoding persistent topological features of a pointcloud (on the left) into a persistence diagram (on the right) by approximating the space through simplicial complexes (in the middle) [11].

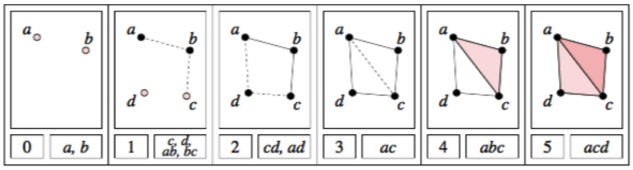


Figure 3: Persistent homology permits to consider the homology of the filtered simplicial complex at all times during the filtration [21].

ties of a (often) continuous space by the analysis of a discrete sample of it, assembling discrete points into global structure [2]. Topological data analysis is interested in problems relating to nonlinear systems, large scale data and development of more accurate models. The study of time series is a great source of problems that focus aspects of that nature. It has applications in digital disease detection [3] which have motivated this study.

TDA is used to infer topological structure in data sets while variations on the method can be applied to study the shape of point clouds (see Figure 2). Instead of choosing one particular threshold, persistent homology considers a continuous height function that permits us to observe the connectivity on our data at all times (see Figure 3). By considering all possible scales, one can infer the correct scale at which to look at the point cloud simply by looking for scales where the persistent homology is stable. To encode the lifetime of that connectivity, we plot birth times and death times (as starting/ending times) in a plane to what we call a *persistence diagram*. In particular, persistence diagrams are a clear and practical tool that allows us the detection of outliers and to capture the dynamics of the system.

In the recent years the method of sliding window persistence has been successful in the study of gene expression time series data (see [19], [18] and [4]). We shall apply this method to the study of reliability during the software

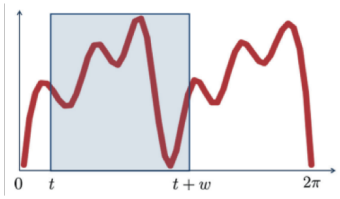


Figure 4: The sliding window persistence method applied to time series data, where the window size is w [18].

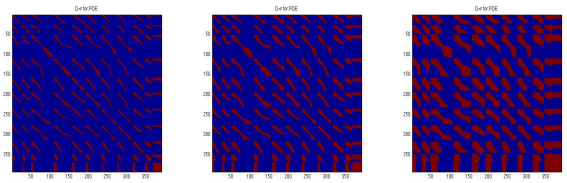


Figure 5: Heat maps representing the distance matrix between the points in the 5-dimensional point cloud (generated by the sliding window of size 5) that are at a distance higher then a certain threshold r, for r=150 (above), r=250 (center), and r=500 (below).

evolution in this paper. Given a time series $g_0, g_1, \ldots, g_S$ measured at times $t_0, t_1, \ldots, t_S$, we consider the graph of g restricted to the interval $[t_i, t_{i+w}]$, where $i = 0, \ldots, S - w$ and w is the length of the window, and we consider the pointcloud

$$\{(g_{t_i}, \ldots, g_{t_{i+w}}) \,|\, i = 0, \ldots, Sw\}.$$

This method is illustrated in Figure 4.

The heat maps in Figure 5 represent the distances between points in the data cloud at several threshold levels, permitting us to track the relevant parameters for the construction of simplicial complexes.

The input structure is given as the pairwise distance matrix – using Euclidean distance – between points in a given point cloud. In Figure 6 we can see three steps of the construction of the Vietoris-Rips complex providing us with the persistence diagram encoding the topological information of the software failures over time and across new versions during its lifetime.

The simplicial complexes originated in the filtration of the space at several levels varying according to a parameter r for the input time series with parameters: $r = 150$, $r = 250$, and $r = 500$ (see Figure 6). These parameters were chosen randomly to illustrate the construction of the simplicial complex at each step while increasing the distance threshold.

Persistence diagrams encode the birth and death time of a topological feature of the data. Much of the applications within the recent research require manipulation and comparison of persistence diagrams. We will focus on the degree 1 persistence diagrams as they are the ones that provide us with more information on the topological features of the input data (see Figure 7). The outlier on the top right corner in all the diagrams represents a qualitative feature that is born late but has a longer lifetime. On the other hand, most of the features that are born early are so close to the diagonal (have a 'short life') that can be considered noise. These persistence diagrams were computed using the open source freely available tool perseus [17] that is based on [15].

4. COMPARING SOFTWARE EVOLUTION

The bottleneck distance is still today the standard method to compare two persistence diagrams. The bottleneck distance is based on a bijection between the points and is therefore always at least the Hausdorff distance between the two diagrams. For points $p = (p_1, p_2)$ and $q = (q_1, q_2)$ in $\mathbb{R}^2$, let $\|p - q\|_\infty$ be the maximum of $|p_1 - q_1|$ and $|p_2 - q_2|$. Let X and Y be multisets of points. Every persistence diagram is

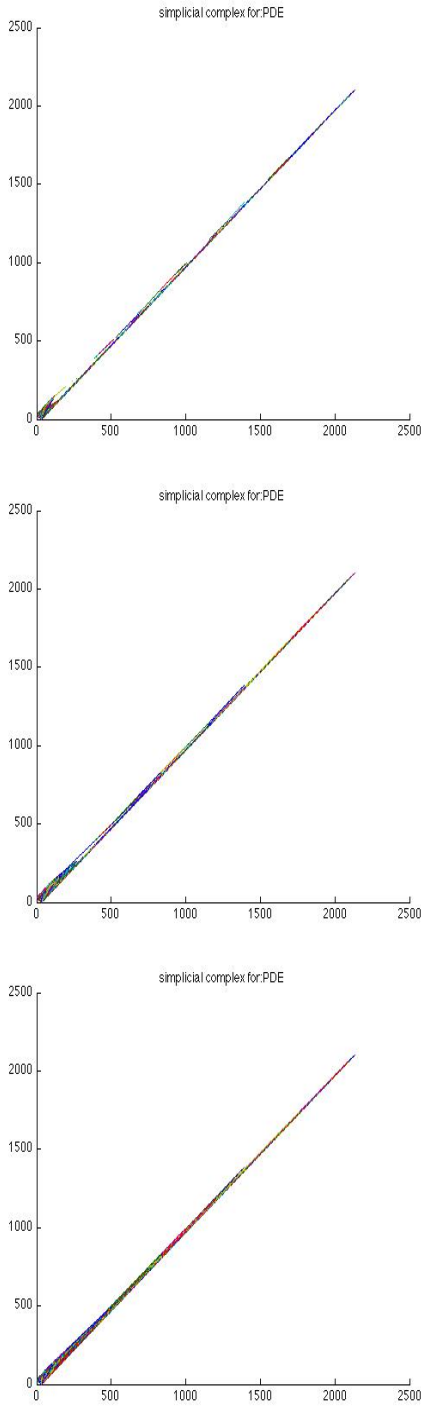


Figure 6: Simplicial complexes approximating the shape of the 5 dimension point cloud (generated by the sliding window of size 5) that are at a distance higher then a certain threshold r, for $r = 150$ (above), $r = 250$ (in the center), and $r = 500$ (below).

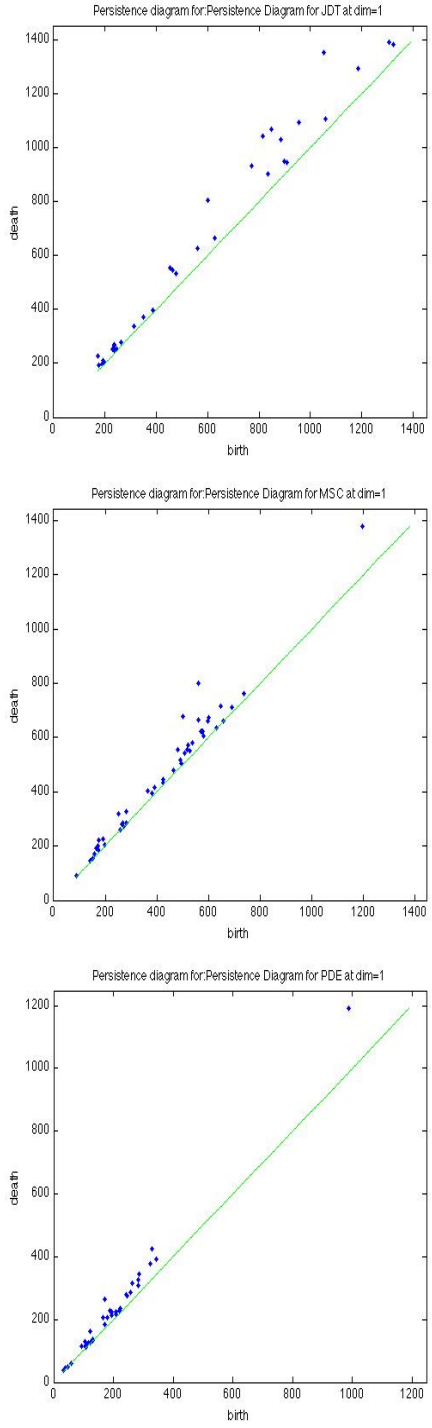
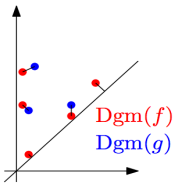


Figure 7: Persistence diagrams of dimension 1 for the topological data analysis of the software faults data in the several versions and updates of the software products JDT (above), MSC (in the center) and PDE (below).



	JDT	MSC	PDE
JDT	0	95.5	72.5
MSC	95.5	0	95.5
PDE	72.5	95.5	0

Figure 8: The bottleneck distance comparing two persistence diagrams [5] (on the left), and the bottleneck distances between persistence diagrams of degree 1, representing the software products JDT, MSC and PDE, using the package TDA for R [7, 6] (on the right).

such a multiset. The bottleneck distance between X and Y is defined as

$$d_B(X,Y) = \inf_{\eta} \sup_{x \in X} \|x - \eta(x)\|_\infty,$$

where the infimum is taken over all bijections η from X to Y. Each point with multiplicity k in a multiset is interpreted as k individual points, and the bijection is interpreted between the resulting sets.

As illustrated in Figure 8 for the case of persistent diagrams $Dgm(f)$ and $Dgm(g)$, the bottleneck identifies the closest elements of each diagram and determines the global distance based on that [5]. We have used the TDA package of the statistical application R [7, 6] to compute these distances, included in the table of Figure 8.

The computation of the bottleneck distances between the persistence diagrams in degree 1, each of them corresponding to one software product, permit us to access the distance between the evolution of reliability behavior similarities of these software products themselves.

By direct analysis of the table in Figure 8, the closest persistence diagrams in degree 1 are JDT and PDE while the most distant are MSC and JDT or MSC and PDE, distances that cannot be distinguished by the basic bottleneck distance. Interesting here is to observe that JDT and PDE are both open source Eclipse projects with quite different development practices from the industrial MSC software product. This seems quite promising for further research, as it shows that topological properties of the reliability time series may be used to as a measure to characterise development environment in respect to various system properties.

5. CONCLUSION AND FURTHER WORK

In this paper we present an initial investigation towards new approach to predict reliability of large software systems by using the topological data analysis approach. Software system reliability is related to failure time series and software reliability models aim to estimate the remaining number of faults that may lead to failure in software system under the test. Our approach was to use topological data analysis of failure time series data during software evolution that may provide us with a complementary understanding of the reliability behavior of that evolution. This is qualitative information on the topological properties of the pointcloud generated by the failure time series data within the sliding window approach used in this study. In that, we show how to extract the topological measure from the derived higher dimension pointcloud, and how to use that information to

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.