Analyzing excessive user feedback: A big data challenge

Faiza Allah Bukhsh
Department of Computer Science
University of Twente
Enschede, the Netherlands
Email f.a.bukhsh@utwente.nl

Jeewanie Jayasinghe Arachchige
Department of Computer Science
University of Ruhuna
Sri Lanka
Email jeewanie@dcs.ruh.ac.lk

Furqan Malik
Department of Computer Science
Khan Institute of CS &IT
Rawalpindi, Pakistan
Email fmaerospace@gmail.com

***Abstract*—User involvement in the process of discovering and shaping the product is the base of software systems. In recent years, however, a shift in the user feedback has been observed: repositories of user data have become increasingly more subjected to analysis for improvement purposes. Significant surge has been seen in feedback collected from users in the form of reviews and ratings along with app usage statistics. This led software engineering researchers to deploy big data analytics techniques in order to figure out the requirements that should be met in the future software system releases. While a variety of big data analytics methods exist, it is not clear which ones have been used and what are the benefits and disadvantages of these proposals. In this paper, we have aimed to outline the recently published proposals for big data analytics techniques for user feedback analysis. We found that the majority of the techniques rest on natural language processing concepts and visualization. Our findings also indicate that the majority of the proposals come from the United States, Germany and the United Kingdom. Moreover, we also found the proposed techniques perform well with the chosen datasets however the generalizability and scalability of these method raised concerns as these methods are not evaluated based on real-world cases.**

***Keywords*—Big data analytics, Feedback analysis, User Reviews**

I. INTRODUCTION

For software products to be highly usable and remain competitive in the ever changing world, enhancements and innovative features must be implemented on a regular basis. Understanding users' requirements is one of the challenging task in this process. Traditionally, requirements discovery has been contingent on the degree of users' involvement and participation and the collaboration between users and requirements engineers [1]. However, in the recent years requirement engineering (RE) is taking a shift towards becoming increasingly more a data-intensive activity [2]. With exponential growth of data, this shift has been seen to harness power of data for RE [3]. This shift is observed in a broad range of activities: from the identification of stakeholders, to exploring the problem solution domain, and to finally determine requirements are getting focused on data.

Several tools and techniques become available to collect, store and process large amount of data. Also, various methods have been proposed recently to drive requirements from the data, especially from user provided feedback. Such methods also help to quickly discover bugs and feature requests from very large and noisy datasets, which ultimately results in improved quality, rapid evolution and less time to ship the software product, without losing market share and user interest. These methods borrow techniques for big data analytics [4] from the field of very database management systems and information retrieval. While a variety of big data analytics methods exist, it is not clear which ones have been used in proposals for user feedback analysis and what are the benefits and disadvantages of these proposals. We look into this by conducting a state of the art review of big data analytics techniques proposed for user feedback analysis. Such a review helps us to understand the state of the art and identify the possible limitations and research gaps in the field.

The rest of the paper is structured as follows: Section II describes our review protocol. In Section III, big data analytics techniques for user feedback are presented. Section IV offers discussion, Section V is about the threats to validity to our findings and Section VI concludes our findings.

II. RESEARCH METHODOLOGY

In order to perform a systematic review we have followed the guidelines of Kitchenham [5] consisting of following steps,

- Objective:The objective of this study is to better understand and summarize the techniques, proposed in literature, for user feedback analysis to facilitate big

978-1-5386-9355-1/18/$31.00 ©2018 IEEE
DOI 10.1109/FIT.2018.00043

data driven requirement engineering. This review will identify the limitations of proposed studies as well as investigate any research gaps.

- Strategy: To extract the comprehensive list of all relevant studies, digital libraries mainly ‘Scopus’ and ‘Google Scholar’ were used. Moreover, we also performed the backward and forward reference search to explore any missing studies. In backward reference search, we explored the reference section of found studies. While in forward reference search, we go through the list of all the future studies that cite the particular study.
- Search String: To build the search terms synonyms are incorporated in search strings using Boolean OR whereas major terms are link together using Boolean AND operator, which resulted in the following search string:
  "big data analytics" OR techniques OR methods OR approaches AND "software user feedback"
- Study Selection:The search string resulted in a pool of 52 studies based on the defined search strings. To find the relevant studies that can be of high value, we scrutinize the studies titles as well as we have defined following quality assessment criteria.
  1) Does paper proposes a novel method/technique for analytic’s regarding requirement engineering?
  2) Does the proposed method/techniques has been validated?
  3) Does proposed method based on specific data analysis technique?

.

As a result, only 40 studies were selected, which are align with our objective and will be used for result synthesis. It is to be noted that we do not provide the complete list of selected paper in references list because of space limitations.

III. FINDINGS

An initial literature search for big data analytics for user feedback analysis shows that, most of the studies, found in literature, propose innovative frameworks and/or architecture to capture the user‘s feedback from data analysis. [6], [7], [8], [9], [10]. For instance, Sherief et. al. [10] presented an architecture for structured feedback acquisition based on two phase empirical study. Similarly, Lim et al. [9] proposed a method named StakeRare that uses social networks and collaborative filtering to identify and prioritize requirements.

Big data is known for its two divisions, data management and analytics. Data management is about supporting technologies to acquire, store and retrieve data for analysis. Analytics is about acquiring intelligence from big data. Gandomi et al [4]introduced a framework of data analytic by categorizing data into five division. As shown in the following Table I, we have applied this framework on our 40 studies and found that 13 studies doesn’t belong to any division of Gandomi’s framework [4].

We will keep highlighting these divisions in the following sections.

Further analysis of studies such as Fig 1 shows the list

Studies Venue	Count
Text Analytics	21
Audio Analytics	2
Video Analytics	0
Social Media Analytics	5
Predictive Analytics	0

TABLE I
STUDIES CATEGORIZATION

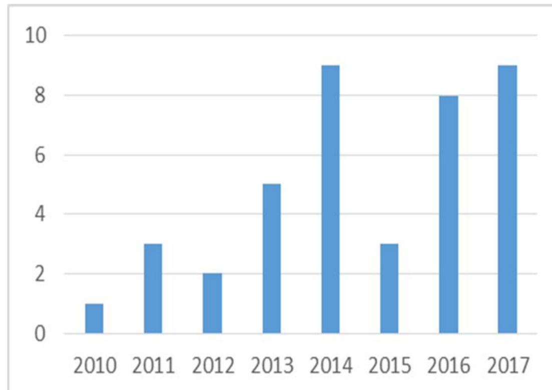

Fig. 1. Studies count by year

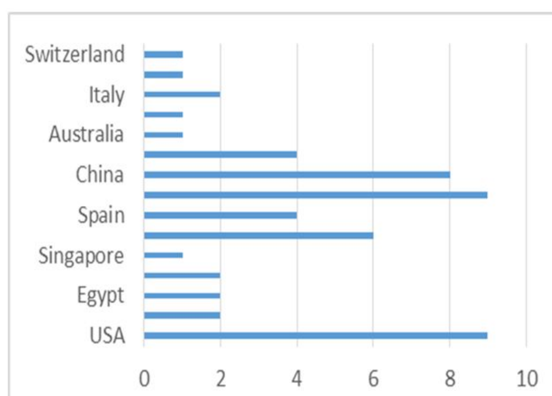

Fig. 2. Study count by country

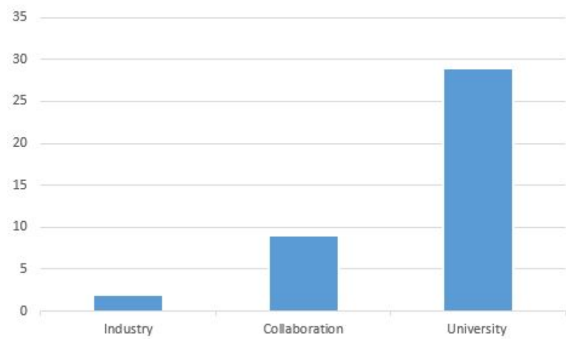

Fig. 3. Study count by author association.

of selected studies with respect to year of publication with a increased trend of publication. The possible reason for this could be the emergence of large amount of data and need of its analysis. However, it is also interesting to notice that only three studies are found that belong to year 2015. Furthermore, we have also analyzed the studies count with respect to country. Fig 2 shows that, USA, Germany and UK are found to be most active in the data analysis and contributing on overall of 18 studies in our review. Fig 3 shows that studies count based on author's association with the academia or industry. It is interesting to see that most of the studies are provided by academic researcher while only one study belong to the researchers from industry. Study count by authors' association also reveal the need of academia and industry collaboration to harness the full potential of data-driven user feedback analysis.

In section III-A, the discussion of user feedback analytics strategies on data analysis are presented. Sub-section III-B discussing the extent, to which the proposed techniques are evaluated in real life context. Finally, section III-C outlines the short comings of discussed big data driven methods.

A. Techniques for User feedback analysis

In this section, we will discuss only those studies that suggest novel methods to extract user information from data provided by users either in form of feedback, software logs, rating, etc. We regard those studies as novel that suggest original method/framework for user feedback analysis. Requirements engineering techniques exploit the immense amount of data provided in a form of reviews, comments, etc by the users. The main objective of these techniques is to extract information from users' feedback in order to understand the applications' bugs and to motivate the future release of application. In the following, we have outlined the few techniques of user feedback analysis that are mainly based on text analysis which is most known type of analysis(see Table I) Text analysis of users feedback will allow us to categorize application reviews into bug reports and feature requests.

- Johann et al. [11] present a simple approach for feature extraction (SAFE) from app descriptions and app reviews. Features were manually extracted from 10 apps and reviewed. Results depicts that for well-maintained app pages such as for Google Drive suggested approach has a precision of 87% and on average 56% for 10 evaluated apps. The proposed approach gives analysts a feature-based perspective on their apps.
- Chen et al, [12] contributes an architecture-centric methodology to address technical, organizational, and rapid technology change challenges of both big data system development and agile delivery. Proposed methodology reason about trade-off, for big data value discovery, planning and estimating continuous delivery of value.
- Maalej et. al. [13] proposed a probabilistic classification technique to categorize app reviews into four types: bugs, feature request, user experiences and rating based on meta-data. They concluded that, classification resulted in poor performance when only done on the basis of meta-data. Improved results were achieved by combining natural language processing techniques with multiple binary classifiers.
- Wei et. al. [14] proposed techniques that use opinion mining and clustering to extract opinion expressions from on-line reviews to capture meaningful feedback and determine user satisfaction level based on the captured feedback to derive evolutionary requirements with association of software revenue. For results evaluation of their proposed techniques they used on-line reviews of popular apps from app store and amazon.com. Moreover, the proposed approaches performed well even with large amounts of review data.
- Guzman et. al. [15] proposed a feedback visualization approach to show app reviews from different viewpoints including: general, review, feature and topic-feature based. They found the approach useful for software evolution and participants came to same conclusion regarding user reviews using 'feedback visualization tool'.
- Fu et. al [16] proposed a tool named 'WisCom' that can be used to analyze thousands of app reviews to determine why user dislike a particular app

and inconsistencies in user reviews and ratings. They were able to identify inconsistencies between user comments and ratings how users' complaints changed over time and high level knowledge of the market place like global trends.

- Dumitru et. al. [17] proposed a recommendation system that extract feature from software specifications available online. They employed text mining and incremental diffusive clustering algorithm to discover domain-specific features along with that they also used association rule mining to capture affinities among software features. They concluded that, the proposed recommender system is able to reduce labour intensive task of domain analysis and result in increased reuse and reduced time to market.

There exist many (e.g [18], [19], [20] [21]) but due to space constraints, we have summarized only few most recent methods.

B. Evaluation of the Techniques

We explore the literature in order to find those studies that deals with the evaluation of data techniques. However, we haven't found many literature studies that evaluate the existing big data techniques. There could be two main reasons for such research a gaps: First, most of these analytic techniques, discussed in previous section, are published very recently as shown in figure 2. Second, none of the analytic techniques have got enough popularity to be used and evaluated empirically. To argue the usefulness of user feedback analysis, we found very relevant big data studies. Both of these studies assess the user feedback empirically to investigate their usefulness.

Bano et. al. [22] highlight the use of user feedback analysis for the selection of a services among others providing similar functionalities. Among others, 92 services were considered, which provide functionality i.e. to send SMS with minimal cost within Australia. To extract the useful feedback provided by users for each service, a sentiment analysis tool was used. Among 92 services, 5 services were selected which are popular on social media and have gain maximum positive comments.All these results were obtained after utilizing data analysis techniques.

Pagano et. al. [23] inspected the one million reviews on AppStore in order to inspect feedback content and its impact of software engineering teams. Review data is classified into 22 categories belonging to free and paid applications. Frequency of feedback, length of feedback and relative length of feedback with respect to time are the key characteristics that are assessed. To investigate the content of feedback, iterative content analysis technique was used. Content of feedback is classified into 17 topics e.g. praise, helpfulness, recommendation, feature request, etc. Based on each topic, nature of reviews are identified as positive or negative nature. Finally, the empirical study argue that the use of large amount of reviews as a tool for developers to understand the users's evolving need.

C. Challenges

During the review process, several challenges are identified. Most important of them is the generalizability of the proposed techniques. Various techniques are evaluated only against reviews of certain kind of apps from particular app store and platform. Similarly, majority of the techniques are evaluated against specific datasets, which raise concerns about their scalability. Moreover, the implicit app usage data and meta data of the apps from stores are not used, which can be useful in determining critical flaws and user experience provided by certain app features. It is also noticed that the Natural language processing (NLP) and Machine learning (ML) techniques applied for feedback analytic have their own limitations e.g. understanding sentence structure and determining quality of clustering or topic modelling methods. We believe that, the improvement in NLP and ML methods will lead to improve analysis of user feedback. For increasing the generalizability of the techniques, diverse dataset should be used. Collection of app reviews from different app stores, platforms and in different languages need to be considered. To increase the scalability of the techniques frameworks like Apache Hadoop and Apache Spark should be used as they provide not only efficient storage of big data but scalable implementation of popular machine learning algorithms. Lastly, the proposed techniques should be validated against real industry cases to determine their reliability. Therefore, close industry academia collaboration is required.

IV. DISCUSSION

This review outlines the philosophy behind analytic's research and presents existing techniques that can be used by developers and analysts to better understand users' satisfaction about software application. User feedback and app usage statistics, in various forms e.g. reviews, ratings and logs etc., provide developers/analysts the useful insights about app and impact subjective business value [24]. The insights from data yields robust mechanism directly from the end users' needs. The general user feedback about particular app feature can also be identified. With number of inline requirements gathered from users' feedback, most important requirements are required to be prioritized to identify most valuable requirements. Garnet et. al. [25] proposed a tool named 'ConTexter' to automatically prioritize end-users feedback using keywords based on information retrieval techniques.

The literature studies cover a broad area of interest ranging from continuous feedback collection, run time software evaluation and acquisition of high quality requirement. Similarly, Almaliki et. al., [26] proposed an adaptive feedback acquisition mechanism that model different user preferences. They employed the concept of Persona (a descriptive model of the user, encircling information such as user characteristics) to aid developers/analysts understand the various users' behaviour and increase their ability to design feedback acquisition systems more efficiently. The runtime evaluation of software by users is considered a powerful way to capture richer information about software usage. It could be used by developers on taking evolution and maintenance decisions. For this reason, Sherief et. al.[27], proposed devising a framework for users' feedback evaluation at runtime. Today's software systems are highly complex and consisted of various complicated and conflicting requirements. In order to help requirement engineers gather high quality requirements and overcome information overload.

Among the number of proposed techniques and frameworks [26], [27] [27] , a significant research gap has been noticed regarding usage of big data. Very few of the proposed techniques are evaluated against large datasets, which raise concerns about the scalability of those methods. Limitations are also identified regarding noisy datasets and incorporation of domain specific words (such as 'good' or 'needs' or 'please fix!'), that occur in many user reviews but ignored by the suggested approaches. Another critical concern is regarding the generalizability of the some of the proposed techniques, as these techniques are evaluated only against reviews of certain kind of apps from particular store and platform. Hence, It is unclear that whether those techniques can attain similar good results when being applied to other kinds of apps on different platforms (e.g. iOS). Furthermore, the proposed techniques tend to use datasets instead of reviews. There is rich meta-data available about applications on app stores like number of downloads, number of app crashes, device name that downloaded the apps. Also, there is implicit data collected by the applications about usage (e.g. time spent by user, while using certain feature). The effect of using such datasets is unknown as majority of the studies use and evaluate their techniques using datasets consisting of text and ratings, only. Furthermore, with respect to topic modelling, the quality assessment criteria to evaluate the quality of generated topics (e.g. whether the topics identified should considered as 'topics' or not) is limited in its nature. Visualization tools and techniques can be further improved by providing functionality to detect conflicting user opinions and reviews can be analyzed along with user demographic characteristics. Likewise, limitations of the lexical sentiment analysis for detecting sarcasm and context can be improved by including the reviews rating in the computation of the sentiment score, which is further helpful in identifying weak areas of the application. Finally, real industry cases are required to be used to fully assess the robustness and reliability of the proposed techniques.

V. THREATS TO VALIDITY

The main threat to validity lies in the study search and study selection procedure. Even with the defined set of search string, there are chances of relevant studies to be left out. We have mitigated this limitation but iteratively executing the search strings and by going through the reference list of selected studies. The possible biases in study selection procedure is mitigated by applying the study selection criteria and quality assessment questions. Finally, the categorization of selected studies as novel and empirical can be argued. Being a relatively new field of study, at one hand, there are very few empirical studies while on the other hand the novel method proposed in studies are usually evaluated by small app/data.

VI. CONCLUSION

Analytics encourages the use of data to derive new insights. We have found there is a large number of methods proposed to analyse user reviews, each type of proposed methods has a unique way to achieve the goal of discovering user opinion. We found that the majority of the techniques are based on natural language processing concepts and visualization and majority of the proposals come from the United States, Germany and the United Kingdom. Moreover, we also found the proposed techniques perform well with the chosen datasets however the generalizability and scalability of these method are still question of concerns. Second, we found that, despite of the high number of found proposals, very few of them are empirically evaluated in real-world situations and we have no way to know to what extent the methods would scale up. In turn, this implies that we need more evaluation of the proposed techniques on real industry cases to fully determine their generalizability and effectiveness from big data perspective. Moreover, it is noticed that as the majority of the techniques are using NLP methods, they have reported shortcomings which are translated to open research areas for the future e.g. quality assessment of topic modelling methods. In addition to we suspect that Industry does a lot of big data with some kind of improvement goals but they rarely to publish papers about them.

VII. Acknowledgement

We would like to thank Dr. Maya Daneva, Assistant Professor, University of Twente, The Netherlands, for her valuable input.

References

[1] S. Lauesen, *Software requirements: styles and techniques*. Pearson Education, 2002.

[2] N. Genc-Nayebi and A. Abran, "A systematic literature review: Opinion mining studies from mobile app store user reviews," *Journal of Systems and Software*, vol. 125, pp. 207–219, 2017.

[3] E. C. Groen, N. Seyff, R. Ali, F. Dalpiaz, J. Doerr, E. Guzman, M. Hosseini, J. Marco, M. Oriol, A. Perini *et al.*, "The crowd in requirements engineering: The landscape and challenges," *IEEE software*, vol. 34, no. 2, pp. 44–52, 2017.

[4] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *International Journal of Information Management*, vol. 35, no. 2, pp. 137–144, 2015.

[5] B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-based software engineering and systematic reviews*. CRC Press, 2015, vol. 4.

[6] M. Almaliki, C. Ncube, and R. Ali, "The design of adaptive acquisition of users feedback: An empirical study," in *Research Challenges in Information Science (RCIS), 2014 IEEE Eighth International Conference on*. IEEE, 2014, pp. 1–12.

[7] D. Bajic and K. Lyons, "Leveraging social media to gather user feedback for software development," in *Proceedings of the 2nd international workshop on Web 2.0 for software engineering*. ACM, 2011, pp. 1–6.

[8] S. L. Lim, D. Damian, and A. Finkelstein, "Stakesource2. 0: using social networks of stakeholders to identify and prioritise requirements," in *Software Engineering (ICSE), 2011 33rd International Conference on*. IEEE, 2011, pp. 1022–1024.

[9] S. L. Lim and A. Finkelstein, "Stakerare: using social networks and collaborative filtering for large-scale requirements elicitation," *IEEE Transactions on Software Engineering*, vol. 38, no. 3, pp. 707–735, 2012.

[10] N. Sherief, W. Abdelmoez, K. Phalp, and R. Ali, "Modelling users feedback in crowd-based requirements engineering: An empirical study," in *IFIP Working Conference on The Practice of Enterprise Modeling*. Springer, 2015, pp. 174–190.

[11] T. Johann, C. Stanik, W. Maalej *et al.*, "Safe: A simple approach for feature extraction from app descriptions and app reviews," in *Requirements Engineering Conference (RE), 2017 IEEE 25th International*. IEEE, 2017, pp. 21–30.

[12] H.-M. Chen, R. Kazman, and S. Haziyev, "Agile big data analytics for web-based systems: An architecture-centric approach," *IEEE Transactions on Big Data*, vol. 2, no. 3, pp. 234–248, 2016.

[13] W. Maalej and H. Nabil, "Bug report, feature request, or simply praise? on automatically classifying app reviews," in *Requirements Engineering Conference (RE), 2015 IEEE 23rd International*. IEEE, 2015, pp. 116–125.

[14] W. Jiang, H. Ruan, L. Zhang, P. Lew, and J. Jiang, "For user-driven software evolution: requirements elicitation derived from mining online reviews," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2014, pp. 584–595.

[15] E. Guzman, P. Bhuvanagiri, and B. Bruegge, "Fave: Visualizing user feedback for software evolution," in *Software Visualization (VISSOFT), 2014 Second IEEE Working Conference on*. IEEE, 2014, pp. 167–171.

[16] B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, and N. Sadeh, "Why people hate your app: Making sense of user feedback in a mobile app store," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1276–1284.

[17] H. Dumitru, M. Gibiec, N. Hariri, J. Cleland-Huang, B. Mobasher, C. Castro-Herrera, and M. Mirakhorli, "On-demand feature recommendations derived from mining public product descriptions," in *Proceedings of the 33rd International Conference on Software Engineering*. ACM, 2011, pp. 181–190.

[18] C. Iacob and R. Harrison, "Retrieving and analyzing mobile apps feature requests from online reviews," in *Proceedings of the 10th Working Conference on Mining Software Repositories*. IEEE Press, 2013, pp. 41–44.

[19] A. K. Massey, J. Eisenstein, A. I. Antón, and P. P. Swire, "Automated text mining for requirements analysis of policy documents," in *Requirements Engineering Conference (RE), 2013 21st IEEE International*. IEEE, 2013, pp. 4–13.

[20] A. Casamayor, D. Godoy, and M. Campo, "Identification of non-functional requirements in textual specifications: A semi-supervised learning approach," *Information and Software Technology*, vol. 52, no. 4, pp. 436–445, 2010.

[21] L. Liu, Q. Zhou, J. Liu, and Z. Cao, "Requirements cybernetics: Elicitation based on user behavioral data," *Journal of Systems and Software*, vol. 124, pp. 187–194, 2017.

[22] M. Bano and D. Zowghi, "Users' voice and service selection: An empirical study," in *Empirical Requirements Engineering (EmpiRE), 2014 IEEE Fourth International Workshop on*. IEEE, 2014, pp. 76–79.

[23] D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," in *Requirements Engineering Conference (RE), 2013 21st IEEE International*. IEEE, 2013, pp. 125–134.

[24] J. da Silva Reis, P. de Alencar Silva, F. A. Bukhsh, and A. F. de Castro, "Configuring value networks based on subjective business values."

[25] S. Gärtner and K. Schneider, "A method for prioritizing end-user feedback for requirements engineering," in *Cooperative and Human Aspects of Software Engineering (CHASE), 2012 5th International Workshop on*. IEEE, 2012, pp. 47–49.

[26] M. Almaliki, C. Ncube, and R. Ali, "Adaptive software-based feedback acquisition: A persona-based design," in *Research Challenges in Information Science (RCIS), 2015 IEEE 9th International Conference on*. IEEE, 2015, pp. 100–111.

[27] N. Sherief, "Software evaluation via users' feedback at runtime," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. ACM, 2014, p. 58.