A Literature Review on Fault Expectation pursuance in Software Engineering

Akshat Agrawal

Amity University, haryana

Abstract: **The exact forecast of where issues are probably going to happen in code can help coordinate test exertion, lessen costs and enhance the nature of programming. We explore how the setting of models, the autonomous factors utilized and the displaying methods connected, impact the execution of blame expectation models. We utilized an orderly writing survey to distinguish 208 blame expectation concentrates distributed from January 2000 to December 2010. We combine the quantitative and subjective after effects of 36 studies which report adequate relevant and methodological data as per the criteria we create and apply. The models that perform well have a tendency to be founded on basic demonstrating systems, for example, Naïve Bayes or Logistic Regression. Mixes of autonomous factors have been utilized by models that perform well. Include choice has been connected to these mixes when models are performing especially well. The system used to construct models is by all accounts compelling to prescient execution. In spite of the fact that there are an arrangement of blame expectation thinks about in which certainty is conceivable, more reviews are required that utilization a solid system and which report their setting, system and execution exhaustively.**

Keywords: **Fault Expectation, Software Engineering**

I. INTRODUCTION

This Systematic Literature Review (SLR) expects to recognize furthermore, break down the models used to anticipate blames in source code in various reviews distributed between January 2000 and December 2010. Our examination researches how demonstrate execution is influenced by the setting in which the model was produced, the free factors utilized in the model and the procedure on which the model was fabricated. Our outcomes empower specialists to create forecast models in view of best learning and practice crosswise over numerous past reviews. Our outcomes likewise offer assistance specialists to settle on compelling choices on forecast models most suited to their specific situation.

1. Expectation displaying is an imperative zone of explore and the subject of numerous past reviews. These examines ordinarily create blame expectation models which permit programming specialists to center advancement exercises on blame inclined code, in this manner enhancing programming quality what's more, improving utilization of assets. The many blame forecast models distributed are unpredictable and different what's more, no up and coming extensive photo of the current condition of blame forecast exists. Two past surveys of the territory have been performed [1] and [2]

2. Our survey varies from these audits in the accompanying ways:

• Timeframes. Our survey is the most contemporary since it incorporates concentrates distributed from 2000- 2010. Fenton and Neil led a basic survey of programming deficiency expectation investigate up to 1999 [1]. Catal and Diri's [2] survey covers work distributed in the vicinity of 1990 and 2007.

• Systematic approach. We take after Kitchenham's [3] unique and thorough systems for directing efficient audits. Catal and Diri did not report on how they sourced their reviews expressing that they adjusted Jørgensen and Shepperd's [4] philosophy. Fenton and Neil did not make a difference the orderly approach presented by Kitchenham [3] as their review was distributed well before these rules were delivered.

• Comprehensiveness. We don't depend on web crawlers alone and, not at all like Catal and Diri, we read through important Journals and Conferences paper-by-paper. Accordingly, we investigated numerous more papers.

• Analysis. Catal and Diri concentrated on the specific situation of studies, including: where papers were distributed, year of distribution, sorts of measurements utilized, datasets utilized and demonstrating approach. Also, we report on the execution of models and orchestrate the discoveries of studies.

We make four noteworthy commitments by displaying:

1) An arrangement of varoius reviews tending to blame expectation in programming designing from January 2000 to December 2010. Scientists can utilize these reviews as the premise of future examinations concerning shortcoming forecast.

2) A subset of 36 blame forecast examines which report adequate relevant and methodological detail to empower these reviews to be dependably dissected by other analysts and assessed by model clients arranging to choose a suitable model for their specific circumstance.

3) An arrangement of criteria to survey that adequate logical also, methodological detail is accounted for in blame forecast thinks about. We have utilized these criteria to recognize the 36 contemplates said above. They can additionally be utilized to control different specialists to manufacture tenable new models that are justifiable, usable, replicable and in which specialists and clients can have a fundamental level of certainty. These criteria could likewise be utilized to guide diary and gathering analysts in verifying that a blame expectation paper has enough revealed a review.

4) A combination of the present cutting edge in programming blame expectation as revealed in the 36 ponders fulfilling our appraisal criteria. This combination is in light of extricating and consolidating: subjective data on the principle discoveries detailed by studies; quantitative information on the execution of these reviews; point by point quantitative examination of the 206 models (or, then again show variations) detailed in 19 ponders which report (or we can compute from what is accounted for) accuracy, review and f-measure execution information. This paper is sorted out as takes after. In the following segment, we show our deliberate writing audit philosophy. In Area 3, we introduce our criteria created to evaluate regardless of whether a review reports adequate logical also, methodological detail to empower us to combine a specific review. Segment 4 demonstrates the aftereffects of applying our appraisal criteria to 208 reviews. Area 5 reports the consequences of separating information from the 36 thinks about which fulfill our appraisal criteria. Area 6 combines our outcomes and Section 7 talks about the methodological issues related with blame expectation examines. Segment 8 recognizes the angers to legitimacy of this review. At last, in Segment 9 we compress and present our decisions.

II. METHODOLOGY

We adopt a deliberate strategy to investigating the writing on the expectation of deficiencies in code. Precise writing surveys are settled in medicinal research what's more, progressively in programming designing. We take after the efficient writing survey approach distinguished by Kitchenham and Charters [3].

The assessment criteria

Our way to deal with distinguishing papers reasonable for combination is inspired by Kitchenham and Charter's [3] thought of a quality check. Our appraisal is centered particularly around recognizing just papers revealing adequate data to permit combination crosswise over reviews as far as replying our exploration questions. To permit this, an essential arrangement of data must be accounted for in papers. Without this it is hard to appropriately comprehend what has been done in a review and similarly hard to satisfactorily contextualize the discoveries detailed by a study. We have created and connected an arrangement of criteria concentrated on guaranteeing adequate logical and methodological data is accounted for in blame expectation examines. Our criteria are sorted out in four stages depicted underneath.

Stage 1: Establishing that the review is a forecast think about. In this SLR it is essential that we consider just models which really do some type of expectation. A few reviews which appear to be revealing forecast models really end up being doing almost no expectation. Huge numbers of these sorts of studies report connections amongst's measurements and deficiencies. Such reviews just demonstrate the inclination for building an expectation show. Moreover, a model is just doing any forecast in the event that it is tried on concealed information (i.e. information that was not utilized amid the preparation procedure) [[33]]. To be viewed as an expectation display it must be prepared and tried on various information [6]. Table 4 demonstrates the criteria we apply to evaluate whether a review is really a forecast contemplate. Table 4 demonstrates that a review can pass this basis as long as they have isolated their preparation and testing information. There are numerous courses in which this division can be finished. Holdout is presumably the least complex approach, where the first informational index is part into two gatherings involving: {training set, test set}. The model is developed oped utilizing the preparation set and its execution is at that point evaluated on the test set. The shortcoming of this approach is that outcomes can be one-sided on account of the way the information has been part. A more secure approach is frequently n-overlap cross approval, where the information is part into n bunches {g1..gn}. Ten times cross approval is extremely normal, where the information is arbitrarily part into ten gatherings, and ten tests done. For each of these tests, one of the gatherings is utilized as the testing set, and all others consolidated are utilized as the preparation set. Execution is then regularly detailed as a normal over each of the ten analyses. M-N overlay cross approval includes another progression by creating M distinctive N-crease cross approvals which builds the dependability of the outcomes and diminishes issues due to the request of things in the preparation set. Stratified cross approval is a change to this process, and keeps the circulation of flawed and nonfaulty information indicates around equivalent the by and large class dissemination in each of the n canisters. Despite the fact that there are more grounded and weaker systems accessible to isolated preparing and testing information we have not made a judgment on this and have acknowledged any type of detachment in this period of appraisal.

Stage 2: Ensuring adequate logical data is accounted for. We watch that essential logical data is introduced by studies to empower fitting understanding of discoveries. An absence of relevant information restrains the client's capacity to: translate a model's execution, apply the demonstrate suitably or rehash the review. For instance, a model may have been manufactured utilizing heritage frameworks with many discharges over quite a while period and has been shown to perform well on these frameworks. It might not then bode well to depend on this model for another framework where the code has just as of late been produced. This is on the grounds that the number and sort of shortcomings in a framework are thought to change as a framework develops [[29]]. On the off chance that the development of the framework on which the model was assembled is not detailed, this seriously constrains a model client's capacity to comprehend the conditions in which the model performed well and to choose this model particularly for inheritance frameworks. In this circumstance the model could be connected to recently created frameworks with baffling prescient execution. The relevant criteria we connected are appeared in Table 5 and are adjusted from the setting agenda created by Petersen and Wohlin [7]. Our specific circumstance agenda likewise covers with the 40 extend attributes proposed by Zimmermann et al. [[37]] as being pertinent to understanding a venture adequately for cross venture demonstrate building (it was unreasonable for us to execute each of the 40 qualities as none of our included reviews report every one of the 40). Setting information is especially vital in this SLR as it is utilized to answer Research Question 1 and translate our general discoveries on model execution. We as it were combine papers that report all the required setting data as recorded in Table 5. Take note of that reviews announcing a few models in view of various informational collections can pass the criteria in this stage if adequate relevant information is accounted for at least one of these models. For this situation, information may be removed from the paper in light of the legitimately contextualized show.

Stage 3: Establishing that adequate model building data is accounted for For a review to have the capacity to help us to answer our examination questions it must report its fundamental model building components. Without clear data about the autonomous what's more, ward factors utilized and additionally the demonstrating method, we can't extricate adequate information to permit combination. Table 6 portrays the criteria we apply.

Stage 4: Checking the model building information Information utilized is essential to the unwavering quality of models. The criteria we apply to guarantee that ponders report fundamental data on the information they utilized. Notwithstanding the criteria we connected in Phases 1 to 4, we likewise grew more stringent criteria that we didn't apply. These extra criteria identify with the nature of the information utilized and the route in which prescient execution is measured. Despite the fact that we at first planned to apply these, this was not viable in light of the fact that the territory is most certainly not adequately develop. Applying these criteria would have brought about just a modest bunch of studies being blended. We incorporate these criteria in Appendix C as they recognize advance essential criteria that future scientists ought to consider when building models.

Applying the assessment criteria

Our criteria have been connected to our included arrangement of different blame forecast ponders. This recognized a subset of 36 at last included reviews from which we separated information also, on which our union is based. The underlying arrangement included papers was separated between the five creators. Each paper was evaluated by two creators freely (with each creator being matched with no less than three other creators). Each creator connected the evaluation criteria to in the vicinity of 70 and 80 papers. Any contradictions on the evaluation result of a paper were talked about between the two creators and, where conceivable, understanding built up between them. Understanding couldn't be come to by the two creators in 15 cases. These papers were at that point given to another individual from the creator group for balance. The arbitrator settled on an official choice on the evaluation result of that paper. We connected our four stage evaluation to each of the included reviews. The stages are connected successively. In the event that a review does not fulfill the majority of the criteria in a stage at that point the assessment is ceased and no resulting stages are connected to the review. This is to enhance the proficiency of the procedure as there is no reason for surveying resulting criteria if the review has as of now fizzled the evaluation. This has the impediment that we didn't gather data on how a paper performed in connection to all evaluation criteria. So if a paper falls flat Phase One we have no data on how that paper would have performed in Phase Four. This appraisal procedure was steered four times. Each pilot included three of the creators applying the appraisal to 10 included papers. The appraisal procedure was refined accordingly of each pilot. We built up our own particular MySQL database framework to deal with this SLR. The framework recorded full reference points of interest what's more, references to pdf's for all papers we recognized as waiting be perused in full. The framework kept up the status of those papers and also giving an on the web procedure to bolster our evaluations of different papers. The framework gathered information from all creators performing appraisals. It likewise gave a balance procedure to encourage recognizing and settling contradictions between sets of assessors. The framework facilitated the organization of the appraisal procedure and the investigation of evaluation results. All information that was extricated from the 36 papers which passed the evaluation is likewise recorded on our framework. A diagram of the framework is accessible from [9] furthermore, full subtle elements are accessible from the third creator.

Extracting data from papers

Information tending to our three research inquiries was separated from each of the 36 at last included reviews which passed all evaluation criteria. Our point was to assemble information that would enable us to break down prescient execution inside individual reviews and over all thinks about. To encourage this, three arrangements of information were separated from each review:

1) Context information. Information demonstrating the setting of each study was removed by one of the creators. This information gives the setting as far as: the wellspring of information contemplated and the development, estimate, application zone and programming dialect of the system(s) contemplated.

2) Qualitative information. Information identified with our exploration questions was removed from the discoveries and conclusions of each review. This was as far as what the papers revealed as opposed to all alone translation of their review. This information supplemented our quantitative information to create a rich picture of results inside individual reviews. Two creators separated subjective information from each of the 36 ponders. Each creator extricated information freely furthermore, contrasted their discoveries with those of the other creator. Contradictions and exclusions were examined inside the match and a last arrangement of information settled upon.

3) Quantitative information. Prescient execution information was separated for each individual model (or model variation) revealed in a review. The execution information we separated shifted by whether the review announced their outcomes through absolute or consistent subordinate factors. A few reviews announced both straight out and consistent outcomes. We separated just a single of these arrangements of results relying upon the route in which the lion's share of results were exhibited by those reviews. The accompanying is a diagram of how we extricated information from straight out and constant contemplates. All out reviews. There are 23 examines detailing clear cut subordinate factors. Absolute reviews report their outcomes as far as foreseeing whether a code unit is probably going to be blame inclined or not blame inclined. Where conceivable we report the prescient execution of these thinks about utilizing accuracy, review and f-measure (the same number of ponders report both accuracy and review, from which an f-measure can be figured). F-measure is regularly characterized as the consonant mean of exactness and review, also, for the most part gives a decent general picture of prescient performance4. We utilized these three measures to look at results crosswise over reviews, and where essential we compute and get these measures from those. Institutionalizing on the execution measures revealed permits correlation of prescient exhibitions crosswise over reviews. Lessmann et al. [[30]] suggest the utilization of predictable execution measures for cross review correlation; specifically, they suggest utilization of Area Under the Curve (AUC).We too extricate AUC where thinks about report this. Index D outlines the estimation of prescient execution. We show the execution of straight out models in box plots. Box plots are valuable for graphically appearing the contrasts between populaces. They are helpful for our outcomes as they make no presumptions about the conveyance of the information exhibited. These case plots exhibit the accuracy, review and f-measure of studies as per a scope of model variables. These variables are identified with the inquire about inquiries exhibited toward the start of Section 2, a case is a case plot indicating model execution in respect to the demonstrating method utilized. Constant reviews. There are 13 thinks about detailing constant subordinate factors. These reviews report their brings about terms of the quantity of flaws anticipated in a unit of code. It was unrealistic to change over the information displayed in these reviews into a typical relative measure; we report the individual measures that they utilize. Most measures revealed by constant reviews are in view of detailing a blunder measure (e.g. Mean Standard Mistake (MSE)), or measures of contrast between expected and watched comes about (e.g. Chi Square). A few constant reviews report their outcomes in positioning structure (e.g. best 20% of broken units). We remove the execution of models utilizing whatever measure each review utilized. Two creators removed quantitative information from every one of the 36 considers. A couple approach was taken to removing this information since it was a perplexing and itemized assignment. This implied that the match of creators sat together recognizing and separating information from a similar paper at the same time.

Synthesizing data across studies

Blending discoveries crosswise over reviews is famously troublesome what's more, numerous product building SLRs have been appeared to introduce no amalgamation [13]. In this paper, we have additionally discovered orchestrating over an arrangement of dissimilar concentrates exceptionally difficult. We removed both quantitative also, subjective information from studies. We proposed to meta analyse our quantitative information crosswise over reviews by joining accuracy and review execution information. However the studies are exceptionally divergent as far as both setting furthermore, models. Meta-dissecting this quantitative information may produce dangerous outcomes. Such a meta-investigation would experience the ill effects of a hefty portion of the impediments in SLRs distributed in different controls [14]. We consolidated our subjective and quantitative information to create a rich picture of blame forecast. We did this by sorting out our information into subjects based around our three investigate questions (i.e. setting, free factors furthermore, displaying procedures). We at that point consolidated the information on each subject to answer our examination questions.

III. RESULTS OF OUR ASSESSMENT

This segment shows the outcomes from applying our appraisal criteria to build up regardless of whether a paper reports adequate logical and methodological detail to be combined. The evaluation result for each review is appeared at the end of its reference in the rundown of included reviews. This demonstrates that lone 36 of our at first included reviews passed all evaluation criteria5. Of these 36 at long last included reviews, three are generally short [[35]], [[32]] and [[36]]. This implies it is conceivable to report important relevant and methodological detail briefly without a huge overhead in paper length. Additionally demonstrates that different papers fizzled at stage 1 of the appraisal since they didn't report forecast models thusly. This incorporates concentrates that lone present connection studies or models that were not tried on information concealed amid preparing. This is a critical finding as it recommends that a generally high number of papers detailing flaw expectation are not by any stretch of the imagination doing any forecast (this finding is additionally revealed by [6]). Table 8 likewise demonstrates that 13 thinks about gave deficient data about their information. Without this it is troublesome to set up the unwavering quality of the information on which the model is based. Table 8 likewise demonstrates that a high number of studies (34) revealed deficient data on the setting of their review. This makes it troublesome to decipher the outcomes announced in these reviews and to choose a fitting model for a specific setting. A few reviews passing the majority of our criteria anonymised their logical information, for instance [[31]] and [[32]]. In spite of the fact that these reviews gave full logical subtle elements of the frameworks they utilized, the outcomes related with each were anonymised. This implied it was difficult to relate particular blame data to particular frameworks. While a level of business secrecy was kept up, this restricted our capacity to break down the execution of these models. This recommends that a scope of developments may likewise be spoken to in these datasets. No reasonable knowledge is given into whether specific informational indexes depend on frameworks created from un tested, recently discharged or inheritance code in light of many discharges. The main three reviews utilizing NASA information which passed the setting period of the appraisal were those which additionally utilized other informational collections for which full setting information is accessible (the NASA based models were not extricated from these reviews). Regardless of whether a review employments NASA information (sourced from MDP or PROMISE) is appeared toward the finish of its reference in the rundown of included reviews.

IV. REFERENCES

[1] N. Fenton and M. Neil, “A critique of software defect prediction models,” Software Engineering, IEEE Transactions on, vol. 25, no. 5, pp. 675–689, 1999.

[2] C. Catal and B. Diri, “A systematic review of software fault prediction studies,” Expert Systems with Applications, vol. 36, no. 4, pp. 7346–7354, 2009.

[3] B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering (version 2.3),” Keele University, UK, Tech. Rep. EBSE Technical Report EBSE-2007-01, 2007.

[4] M. Jorgensen and M. Shepperd, “A systematic review of software development cost estimation studies,” Software Engineering, IEEE Transactions on, vol. 33, no. 1, pp. 33–53, 2007.

[5] B. Kitchenham, “What’s up with software metrics?-a preliminary mapping study,” Journal of Systems and Software, vol. 83, no. 1, pp. 37–51, 2010.

[6] Q. Song, Z. Jia, M. Shepperd, S. Ying, and J. Liu, “A general software defect-proneness prediction framework,” IEEE Transactions on Software Engineering, vol. 37, pp. 356–370, 2011.

[7] K. Petersen and C. Wohlin, “Context in industrial software engineering research,” in Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement. IEEE Computer Society, 2009, pp. 401–404.

[8] P. G. Armour, “Beware of counting loc,” Commun. ACM, vol. 47, pp. 21–24, March 2004.

[9] D. Bowes and T. Hall, “SLuRp: A web enabled database for effective management of systematic literature reviews,” University of Hertfordshire, Tech. Rep. 510, 2011.

[10] T. Menzies, A. Dekhtyar, J. Distefano, and J. Greenwald, “Problems with precision: A response to "comments on ’data mining static code attributes to learn defect predictors’",” Software Engineering, IEEE Transactions on, vol. 33, no. 9, pp. 637 –640, sept. 2007.

[11] H. He and E. Garcia, “Learning from imbalanced data,” IEEE Transactions on Knowledge and Data Engineering, pp. 1263–1284, 2008.

[12] D. Gray, D. Bowes, N. Davey, Y. Sun, and B. Christianson, “Further thoughts on precision,” in Evaluation and Assessment in Software Engineering (EASE), 2011.

[13] D. S. Cruzes and T. Dybå, “Research synthesis in software engineering: A tertiary study,” Inf. Softw. Technol., vol. 53, pp. 440–455, May 2011.

[14] R. Rosenthal and M. DiMatteo, “Meta-analysis: Recent developments in quantitative methods for literature reviews,” Annual review of psychology, vol. 52, no. 1, pp. 59–82, 2001.

[15] B. Turhan, T. Menzies, A. Bener, and J. Di Stefano, “On the relative value of cross-company and within-company data for defect prediction,” Empirical Software Engineering, vol. 14, no. 5, pp. 540–578, 2009.

[16] N. Japkowicz and S. Stephen, “The class imbalance problem: A systematic study,” Intell. Data Anal, vol. 6, no. 5, pp. 429–449, 2002. [17] W. Liu, S. Chawla, D. A. Cieslak, and N. V. Chawla, “A robust decision tree algorithm for imbalanced data sets,” in SDM. SIAM, 2010, pp. 766–777.

[18] C. Hsu, C. Chang, and C. Lin, “A practical guide to support vector classification,” Department of Computer Science and Information Engineering, National Taiwai University, 2003.

[19] N. Cristianini and J. Shawe-Taylor, An introduction to support Vector Machines: and other kernel-based learning methods. Cambridge university press, 2006.

[20] T. Hall, D. Bowes, G. Liebchen, and P. Wernick, “Evaluating three approaches to extracting fault data from software change repositories,” in International Conference on Product Focused Software Development and Process Improvement (PROFES). Springer, 2010, pp. 107–115.

[21] D. Gray, D. Bowes, N. Davey, Y. Sun, and B. Christianson, “The misuse of the nasa metrics data program data sets for automated software defect

prediction," in Evaluation and Assessment in Software Engineering (EASE), 2011.

[22] N. Pizzi, A. Summers, and W. Pedrycz, "Software quality prediction using median-adjusted class labels," in Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on, vol. 3, 2002, pp. 2405 –2409.

[23] J. Davis and M. Goadrich, "The relationship between Precision- Recall and ROC curves," in Proceedings of the 23rd international conference on Machine learning, 2006, pp. 233–240.

[24] A. Lamkanfi, S. Demeyer, E. Giger, and B. Goethals, "Predicting the severity of a reported bug," in Mining Software Repositories (MSR), 2010 7th IEEE Working Conference on, 2010, pp. 1–10.

[25] I. Myrtveit, E. Stensrud, and M. Shepperd, "Reliability and validity in comparative studies of software prediction models," IEEE Transactions on Software Engineering, pp. 380–391, 2005.

[26] G. Liebchen and M. Shepperd, "Data sets and data quality in software engineering," in Proceedings of the 4th international workshop on Predictor models in software engineering. ACM, 2008, pp. 39–44.

[27] H. Zhang and X. Zhang, "Comments on "data mining static code attributes to learn defect predictors"," Software Engineering, IEEE Transactions on, vol. 33, no. 9, pp. 635 –637, sept. 2007.

[28] G. Batista, R. Prati, and M. Monard, "A study of the behavior of several methods for balancing machine learning training data," ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 20–29, 2004.

[29] T. Khoshgoftaar and N. Seliya, "Comparative assessment of software quality classification techniques: An empirical case study," Empirical Software Engineering, vol. 9, no. 3, pp. 229–257,2004.

[30] S. Lessmann, B. Baesens, C. Mues, and S. Pietsch, "Benchmarking classification models for software defect prediction: A proposed framework and novel findings," Software Engineering, IEEE Trans on, vol. 34, no. 4, pp.485–496, july-aug. 2008.

[31] A. Mahaweerawat, P. Sophatsathit, and C. Lursinsap, "Adaptive self-organizing map clustering for software fault prediction," in Fourth international joint conference on computer science and soft-IEEE TRANSACTIONS ON SOFTWARE ENGINEERING This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. 24 ware engineering, Thailand, 2007, pp. 35–41.

[32] T. Mende, R. Koschke, and M. Leszak, "Evaluating defect prediction models for a large evolving software system," in Software Maintenance and Reengineering, ' 13th European Conference on, march 2009, pp. 247 –250.

[33] T. Menzies, J. Greenwald, and A. Frank, "Data mining static code attributes to learn defect predictors," Software Engineering, IEEE Transactions on, vol. 33, no. 1, pp. 2 –13, jan. 2007.

[34] T. Menzies, B. Turhan, A. Bener, G. Gay, B. Cukic, and Y. Jiang, "Implications of ceiling effects in defect predictors," in Proceedings of the 4th international workshop on Predictor models in software engineering, ser. PROMISE '08. New York, NY, USA: ACM, 2008, pp. 47–54.

[35] O. Mizuno, S. Ikami, S. Nakaichi, and T. Kikuno, "Spam filter based approach for finding fault-prone software modules," in Mining Software Repositories. ICSE Workshops '07. Fourth Int'll Workshop on, may 2007, p. 4.

[36] S. Shivaji, E. J. Whitehead, R. Akella, and K. Sunghun, "Reducing features to improve bug prediction," in Automated Software Engineering, 2009. ASE '09. 24th IEEE/ACM International Conference on., 2009, pp. 600–604.

[37] T. Zimmermann, N. Nagappan, H. Gall, E. Giger, and B. Murphy, "Cross-project defect prediction: a large scale experiment on data vs. domain vs. process," in Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering, ser. ESEC/FSE '09. New York, 2009, pp. 91–100.

[38]Tracy Hall, Sarah Beecham, David Bowes, David Gray and Steve CounsellA Systematic Literature Review on Fault Prediction Performance in Software Engineering: IEEE transaction on software engineering