Science of Computer Programming 246 (2025) 103320

Contents lists available at ScienceDirect

Science of Computer Programming

journal homepage: www.elsevier.com/locate/scico

"Your AI is impressive, but my code does not have any bugs" managing false positives in industrial contexts

Szymon Stradowski [a,b,*], Lech Madeyski [b]

[a] Nokia, Szybowcowa 2, Wrocław, 54-206, Dolnoslaskie, Poland
[b] Wrocław University of Science and Technology, Wyb. Wyspianskiego 27, Wrocław, 50-370, Dolnoslaskie, Poland

ARTICLE INFO

Keywords:
Software defect prediction
Machine learning
Real-world
Industry

ABSTRACT

Context: "Your AI is impressive, but my code does not contain any bugs"— such a statement from a software developer is the antithesis of a quality mindset and open communication. What makes it worse is that it is oftentimes true.
Objective: This paper analyses false positives' impact and related challenges in machine learning software defect prediction and describes the mitigation possibilities.
Methods: We propose a broad-picture perspective on dealing with false positive predictions based on what we learned from our industrial implementation study in Nokia 5G.
Results: Accordingly, we draw a new direction in transitioning defect prediction into a well-established industry practice, as well as highlight potential emerging topics in predictive software engineering.
Conclusion: Increasing human buy-in and the business impact of predictions significantly improves the chances of future software defect prediction industry adoptions to succeed.

1. Introduction

Machine learning software defect prediction (ML SDP) has been an attractive field of research for many years. Despite meaningful efforts and an increasing year-on-year number of publications, the industry's adoption still needs to grow [1]. Specifically, we have yet to explore the consequences after the adoption project is complete and define solutions ensuring sustainability for the new process to continue to bring value to the company over time [2]. Hence, we argue that research on new technologies must be treated from a broader business perspective.

This paper is based on our industry adoption project in Nokia 5G, which included exploring the challenges [3], reviewing in vivo literature [1], developing an ML SDP solution with Explainable Artificial Intelligence (XAI) [4], evaluating the cost [5], and drawing lessons learned [6].

We illustrate the "holistic ML SDP workflow," and, based on this consideration, we address one of the most prohibiting aspects of commercial adoptions — dealing with false positives. Also, we provide insight into the practical importance of discussed matters based on our learnings from the commercial machine learning software defect prediction adoption project within the Nokia 5G quality assurance process [3,4].

* Corresponding author at: Wrocław University of Science and Technology, Wyb. Wyspianskiego 27, Wrocław, 50-370, Dolnoslaskie, Poland.
E-mail address: Szymon.stradowski@pwr.edu.pl (S. Stradowski).

https://doi.org/10.1016/j.scico.2025.103320
Received 11 January 2025; Received in revised form 24 March 2025; Accepted 20 April 2025

Available online 7 May 2025

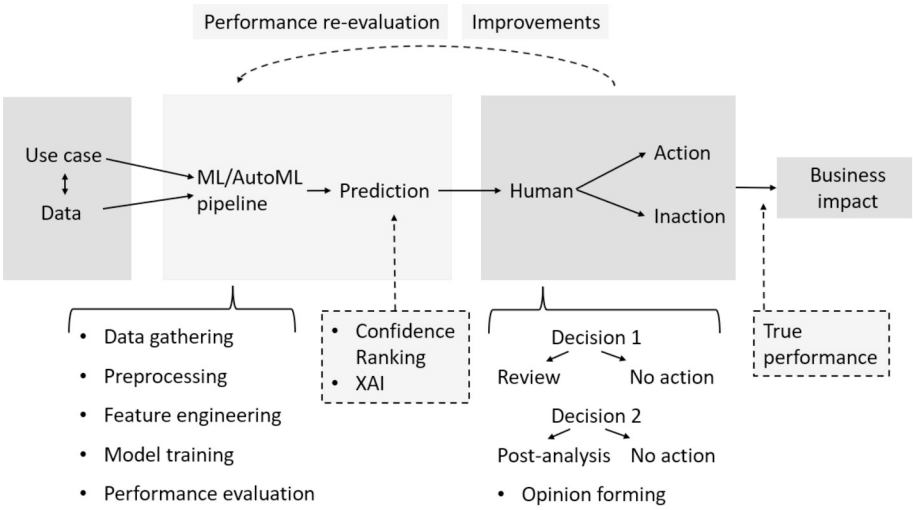

Fig. 1. Holistic ML SDP process workflow.

2. False positive challenges

One of the most disregarded aspects of research in predictive software engineering is the fact that usually, in the end, there is a human actor — a software engineer who needs to take action and decide what to do. The buy-in of this individual is critical for the proposed changes to be acted upon. Conversely, each false positive causes frustration, decreases trust and damages the future potential applicability of the solution. There is a clear expectation from practitioners that false predictions need to be kept to a minimum [7]. Studies analysing individual developer statements highlight general enthusiasm and openness towards AI; however, things get complicated when personal consequences are concerned [7]. When individual work is challenged by AI, initial openness may change to pushback, defensiveness, and distrust.

False positives have been a subject of research for many years, and consequently, many techniques have been developed to increase the predictive performance. In this context, papers presenting the entire confusion matrix and/or using performance measures built upon all of the quadrants of the confusion matrix (e.g., Matthew's Correlation Coefficient (MCC)) as the primary performance metric are important. However, the literature on solutions to tackle misleading predictions that go beyond an increase in predictive performance is scarce. We define two new perspectives to approach the false positive challenges:

- First, we have addressed the iterative nature of the process and expanded the definition after the implementation phase by drawing the entire ML SDP workflow (Section 2.1).
- Second, the confidence of the particular prediction/classification result with a probability ranking (Section 2.2) is a viable tool to help reduce false positives, complementary to increasing the MCC of the created solutions.

2.1. Holistic ML SDP workflow

Integrating AI into larger systems is a difficult challenge on its own [8]. In our case, a human decision point needs to act upon the result to implement the actual change in the existing code, and the prediction itself has minimal value without it. There are many papers describing the ML SDP workflow (e.g., [8]); however, the human aspect needs to be more frequently addressed. Hence, we identify the need to expand to the bigger picture when embedded into a wider quality assurance process in Fig. 1. We consider how the solution fits its business purpose and define new perspectives beyond the main subject of ML SDP research. Below, we offer a generic explanation of the proposed concepts:

- Use case: the starting point of any ML adoption project is the definition of what we want to achieve. Practitioners must clarify the expectations for the solution and define how the predictions will impact the quality assurance process on a daily basis.
- Data: the defined use case needs proper data to support it. Preferably, the required data should be readily available; otherwise, they need to be created (calculated) and maintained, increasing the cost and complexity.
- ML/AutoML pipeline: the standard subject of the majority of ML SDP studies. It consists of several steps such as: data gathering, preprocessing, modelling, and performance evaluation. The pipeline can include more steps and sophisticated techniques to optimise the results [8]. Nevertheless, this is a well-explored process with a wide range of established solutions, algorithms, and frameworks that can be used.
- Prediction (including XAI results and confidence ranking on posterior probabilities: the output of the ML/AutoML pipeline is the prediction - a set of future data models that lead practitioners to achieve better outcomes.

- Human: the prediction has to be acted upon by a human operator to introduce changes to the code base, which can be the most consequential risk phase for the process. The necessary actions can be described as follows:
  - Decision 1: the first decision point is to spend the effort to take into account the results of the prediction. A favourable decision leads to a more detailed review and an opposition to inaction.
  - Review: human actor needs to review the results and build a judgement on their validity. Here, a standalone prediction has much slimmer chances than one equipped with an explanation and confidence.
  - Decision 2: after the results review and initial judgement, the human actor needs to decide if a deeper post-analysis of the code base is required to find and correct the defect.
  - Post-analysis: this is the additional quality assurance effort needed to analyze all evidence to locate and fix the defect in the code.
  - Opinion forming: possibly the most important moment from the long-term perspective comes after finalisation of the human actors activities. It is related to building confidence (or lack thereof) in the ML SDP process outcomes. Negative experiences based on wasted time and effort that do not lead to meaningful results (usually related to past false positives) can impact the future practitioner's decisions in next iterations.
- Action or Inaction: a direct software developer activity to correct the found defect, integrate a new software version and have it tested. This result is the primary business-impacting outcome of ML SDP.
- Iteration (including improvements and performance re-evaluation in time): importantly, ML SDP is an iterative process in nature. After all singular predictions have been decided upon, a new cycle of forecasts based on new data should occur. After each cycle, a performance re-evaluation should happen that monitors changes over several cycles and identifies any traces of degradation. Hence, from a research perspective, longitudinal studies on performance should be conducted instead of one-time evaluations that lose the alignment between the model and incoming data. Second, a forgetting mechanism for outdated data can be activated to keep the models current. Third, as with any other operation in the industry, continuous improvement should be applied, constantly seeking opportunities to improve the process.
- Business impact: resulting from the defined use case and the real predictive performance of fixed defects, including the cost of the operation, reflects how much the process influenced the quality of the product. The actual, iterative performance measured by fixed defects impacts the business outcomes of a solution, not the ML SDP learner's performance measured by MCC. Hence, to achieve positive business outcomes and evident advantages [5,9], the developed solutions must be extended to support practitioners in making favourable decisions.

2.2. Confidence ranking

The majority of predictive methods do not produce a measure of the reliability of the results [10]. We want to address this issue within the ML SDP domain by using a vector of probabilities, also called "posterior probabilities" of an observation belonging to each class, as it provides information about the confidence of the predictions and is available out of the box for classifiers in the mlr3 framework.[1] As eliminating false positive predictions entirely while improving the overall model performance is the challenge in itself, the feasible solution is to limit the number of predictions for which action is triggered. False positive tolerance of the practitioners needs to be balanced with the forbearance of the system. In our cost evaluation study, we applied a straightforward approach and limited the top predictions to act upon at ten for each new predictive cycle [5]. Nevertheless, the feedback loop based on the desired classification confidence level and, additionally, the confidence assigned to the XAI-based explanation will be the direction of our future research.

Algorithm 1 Confidence evaluation for ML SDP.

Input: training data: $(x_1, y_1), ..., (x_m, y_m)$
Input: new data $(x'_1, y'_1), ..., (x'_n, y'_n)$
1) Learners train on 'training data' & make predictions R for $new\,data$
2) Performance is evaluated with MCC metric
3) Local explainability provided with XAI techniques
4) Posterior probabilities C evaluated for predictions R, where $c_n = C(r_n)$
5) Break-even point P calculated for $new\,data$
Output: prediction set R
Output: set of posterior probabilities C for R; where $c_n > P$
Output: prediction set R' sorted by C and cut off at P

The true performance reflects real changes in the code that have been made due to the model's predictions. Studies should show how the solution can maintain good performance with iterative small new data increments and how the confidence of singular predictions translates to real issues being fixed. ML SDP is a supporting mechanism for humans to make better decisions and should be measured accordingly. Finally, from the cost and benefit perspective, it is imperative to define a context-dependent threshold for the confidence level above which the action is mandatory and below which the risk is absorbed.

[1] Many classification models such as logistic regression, naive Bayes, neural networks, models based on generative models, and models derived from mixture densities compute posterior class probabilities directly. Decision trees can be easily adapted to output a class probability by returning the proportion of positive examples from the leaves of the tree.

3. Discussion

While adapting the new ML SDP solution to real conditions, we encountered consequential pushback on the individual results during our observations and interviews. We offer the following motivation to mitigate the negative assumptions and increase the interest of stakeholders in the ongoing deployment.

3.1. Motivating example

First of all, we have defined a stakeholder management framework based on our XAI results described in a dedicated paper [11]. We analysed six distinct groups of stakeholders: technical staff (believers and agnostics, using our solution), technical staff (sceptics, we want to be using our solution), sponsors and decision-makers (with strong influence on the future of the project), management staff (believers and supporters), management staff (sceptics and agnostics), underlying process owners, as well as technical and management staff outside of the project (to be kept informed and interested).

Consequently, we acknowledged the customer for the ML SDP output from the very beginning. It is essential to keep in mind that the recipient of the prediction is usually a different person than the one who has designed the solution. Hence, we need to distinguish the role of the user, whom we call the technical staff, from that of the persons responsible for the design and implementation called AI integrators. As in our implementation, we used the time-based expanding and sliding window approaches to predict test failures induced by software defects [4], the AI integrator pre-analyses the results for every iteration of predictions with new data and proposes only specific predictions to the technical staff to mitigate false positives and maximize the probability of meaningful action (Fig. 1). As some of our models achieved high precision[2] (above 0.9), all predictions can be used; however, for models with lower levels of precision, we decided to use a fixed number of the most promising ones with the highest confidence ranking to limit the adverse effects of false positives.

At Nokia, we focused specifically on the first two groups of mentioned stakeholders to provide them with different sets of predictions. Believers and agnostics can tolerate more positive predictions but with lower overall confidence. The sceptics we want to gain inroads with are offered only a limited set of predicted defects but with a very high confidence threshold. In such a case, the precision metric that we calculate while building the models and the probability of particular predictions (Algorithm 1) allows accurate estimations of which predictions to act upon and which to neglect. Thanks to this approach, we have been able to decrease the number of sceptics throughout the project meaningfully.

Detecting true positive instances with confidence when they are exceedingly rare within the analysed data set remains a challenge [12]. So, the question we put forward in our internal discussions is “Why fail to try to have it all when you can succeed by having just enough?” Although acting on only a limited set of positive predictions satisfying the confidence thresholds can cause not all predicted defects to be acted upon, it still has a positive business impact for the company by streamlining the quality assurance process with an additional defect prediction mechanism. Finally, in our dedicated cost-benefit analysis, the return on investment of the lightweight scenario using ten predictions with the highest confidence in every prediction cycle was equal to 3.73 [5].

3.2. Generalizability

As our study is based on proprietary industrial data sets and processes, the generalizability of any specific results is limited. Trained models, performance measurements, and XAI outputs will differ depending on the data sets used. In addition, we have only verified our solution within a single project, and cross-project verification still needs to be done. However, as a general approach, the proposed concepts are applicable to any company or large-scale software system. This comes from the fact that the required posterior probabilities are available for machine learning algorithms in machine learning frameworks (such as mlr3[3]). This creates an excellent opportunity to reuse the idea and involve human interaction [12] in other SDP solutions, increasing the generalizability of our proposal. The concept based on using posterior probabilities can be used even beyond SDP.

Second, our approach has now been verified on a real-world data set showing good predictive performance (achieving MCC $>$ 0.3, see Madeyski and Stradowski [4]). Similar efforts can easily include other learners, better hyperparameter tuning, and additional performance-enhancing techniques that will make it suitable for different contexts and provide even better results. Moreover, further elements can be added to our process flow (Fig. 1) depending on the industry context and specific requirements of the underlying quality assurance process.

Last, the proposed concepts promote robust solutions that support practitioners in acting upon the models rather than solely optimizing the model’s predictive performance. This proposition is valid across many different fields within the software engineering domain [2,9].

4. Conclusions

In this paper, we have explored the challenges of integrating ML SDP into a larger quality assurance process of Nokia 5G, explicitly from the practitioner’s perspective, and provide a motivating example. Accordingly, we have defined a big-picture workflow,

[2] Precision shows how often a machine learning model correctly predicts the positive class.

[3] https://mlr3book.mlr-org.com/.

expanding the existing ML research area and proposed to utilise the confidence ranking for singular predictions to enable practitioners to act only on a pre-selected subset, decreasing the risk of having to deal with false positives. Consequently, we determine the actual performance of the SDP by measuring the number of corrected defects rather than the number of predicted ones to find an acceptable risk threshold and increase the overall profitability of ML SDP. The proposed approach supports robust AI/ML solutions that encourage practitioners to act on the models rather than only optimising the models themselves.

CRediT authorship contribution statement

Szymon Stradowski: Writing – original draft, Conceptualization. **Lech Madeyski:** Writing – review & editing, Supervision, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Szymon Stradowski reports financial support and administrative support were provided by Nokia Corporation. Szymon Stradowski reports financial support was provided by Ministry of Education and Science of the Republic of Poland. Szymon Stradowski reports a relationship with Nokia Corporation that includes: employment, equity or stocks, and non-financial support. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research was conducted in partnership with Nokia (with Szymon Stradowski as an employee) and was financed by the Polish Ministry of Education and Science 'Implementation Doctorate' program (ID: DWD/5/0178/2021).

References

[1] S. Stradowski, L. Madeyski, Industrial applications of software defect prediction using machine learning: a business-driven systematic literature review, Inf. Softw. Technol. 159 (2023) 107192, https://doi.org/10.1016/j.infsof.2023.107192.
[2] H. Noman, N. Mahoto, S. Bhatti, A. Rajab, A. Shaikh, Towards sustainable software systems: a software sustainability analysis framework, Inf. Softw. Technol. 169 (2024) 107411, https://doi.org/10.1016/j.infsof.2024.107411.
[3] S. Stradowski, L. Madeyski, Exploring the challenges in software testing of the 5g system at Nokia: a survey, Inf. Softw. Technol. 153 (2023) 107067, https://doi.org/10.1016/j.infsof.2022.107067.
[4] L. Madeyski, S. Stradowski, Predicting test failures induced by software defects: a lightweight alternative to software defect prediction and its industrial application, J. Syst. Softw. 223 (2025) 112360, https://doi.org/10.1016/j.jss.2025.112360.
[5] S. Stradowski, L. Madeyski, Costs and benefits of machine learning software defect prediction: industrial case study, in: 32nd ACM International Conference on the Foundations of Software Engineering, 2024, pp. 92–103, https://doi.org/10.1145/3663529.3663831.
[6] S. Stradowski, L. Madeyski, Bridging the gap between academia and industry in machine learning software defect prediction: thirteen considerations, in: 38th International Conference on Automated Software Engineering, 2023, pp. 1098–1110, https://doi.org/10.1109/ASE56229.2023.00026.
[7] Z. Wan, X. Xia, A.E. Hassan, D. Lo, J. Yin, X. Yang, Perceptions, expectations, and challenges in defect prediction, IEEE Trans. Softw. Eng. 46 (2020) 1241–1266, https://doi.org/10.1109/TSE.2018.2877678.
[8] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, T. Zimmermann, Software engineering for machine learning: a case study, in: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, 2019, pp. 291–300, https://doi.org/10.1109/ICSE-SEIP.2019.00042.
[9] B. Johnson, C. Bird, D. Ford, N. Forsgren, T. Zimmermann, Make your tools sparkle with trust: the picse framework for trust in software tools, in: 45th International Conference on Software Engineering: Software Engineering in Practice, IEEE Press, 2023, pp. 409–419, https://doi.org/10.1109/ICSE-SEIP58684.2023.00043.
[10] I. Nouretdinov, S.G. Costafreda, A. Gammerman, A. Chervonenkis, V. Vovk, V. Vapnik, C.H. Fu, Machine learning classification with confidence: application of transductive conformal predictors to mri-based diagnostic and prognostic markers in depression, NeuroImage 56 (2011) 809–813, https://doi.org/10.1016/j.neuroimage.2010.05.023.
[11] S. Stradowski, L. Madeyski, Interpretability/explainability applied to machine learning software defect prediction: an industrial perspective, IEEE Softw. 42 (2025), https://doi.org/10.1109/MS.2024.3505544.
[12] X. Liu, Y. Zhou, Y. Tang, J. Qian, Y. Zhou, Human-in-the-loop online just-in-time software defect prediction: what have we achieved and what do we still miss?, Sci. Comput. Program. 244 (2025) 103296, https://doi.org/10.1016/j.scico.2025.103296.