Can A Machine Learn Through Customer Sentiment? A Cost-Aware Approach To Predict Support Ticket Escalations

Colin Werner, Ze Shi Li, Daniela Damian

1 Introduction

In an ideal world, a customer support ticket would never be escalated; however, the reality is that support tickets are continually escalated to management at great costs to software organizations [1] [2]. An escalation usually occurs when a customer is not satisfied with how a support ticket is proceeding and is associated with an overly negative sentiment. Our previous analysis of support tickets in IBM software projects showed that the customer sentiment in the escalated tickets was significantly more negative than non-escalated support tickets [3].

Given the connection between the state of a customer's happiness, or lack thereof, and whether a support ticket is escalated, utilizing emotional awareness with respect to support ticket escalations could enhance a software organization's support performance to help ensure customers are satisfied. We describe an approach that 1.) analyzes and monitors the emotions in conversations between a customer and a support analyst and 2.) uses machine learning algorithms to predict if a support ticket is a candidate for escalation. We developed and evaluated our approach on a data set containing 356 software support tickets and 10,172 comments across all support tickets from one of our industrial collaborating software partners with an established support organization, Alpha[1]. We obtained promising results in predicting whether a support ticket will escalate by utilizing the sentiment of a particular support ticket's customer-analyst conversation. By implementing our approach in a practical setting, a support analyst, or the organization, may adapt their approach to handle a support ticket that has has been identified as a candidate for escalation. Overall, our approach enables an organization to improve its escalation awareness by tracking the customer sentiment across its support tickets. The organization can streamline and enhance its customer support process, therefore increasing its customer satisfaction.

Recent work has seen a growing interest in studying the effect of emotion and mood in software engineering, in particular with software development, globally distributed teams, and requirements engineering. For example, a difference in

[1] Name has been changed for confidentiality.

developer emotions could be the catalyst between strong and poor development performance and software quality. Similarly, higher awareness of emotional states in distributed development teams can lead to more effective collaboration between developers [4]. Our approach leverages the awareness of customer emotion in predicting a potential support ticket escalation and thus offering an organization automated support in processing and prioritizing a large number of support tickets. In addition, while predicting an escalation is an admirable goal, it only makes sense if it provides economic viability for the organization. Our approach provides an additional step to help an organization's decision making process and assessment of trade-offs in the practical application of sentiment analysis and machine learning to predict escalations. This practical step utilizes the average cost of an escalation to an organization (A1) relative to the additional effort to investigate a support ticket flagged as a candidate for escalation (A2) to provide a cost-analysis on which to base decisions. Through our approach at Alpha with hypothetical values of $10,000 and $5,000 for A1 and A2, respectively, Alpha could achieve an average return on investment (ROI) of $308,896.

Tab. 1: Escalation Prediction and Sentiment Analysis in SE

If a support ticket is not resolved in an acceptable manner, it may be escalated by the customer or the support analyst. An escalated support ticket may present itself as an extremely expensive operation for an organization. Although limited, research exists into models to predict support ticket of high risk of escalation. Our own previous research employed feature engineering to develop prediction models for customer support ticket escalations at IBM [1]. Other approaches ([2], [3], [4]) mine historical defect report data in an organization and predict the escalation risk of current defects. They use a cost-sensitive machine learning (ML) framework that calculates the return on investment in a particular ML algorithm. Our approach that uses sentiment analysis to make predictions about support tickets that are candidates for escalation also introduces a similar calculation of the relative cost to the organization in pursuing ML escalation prediction; furthermore, we provide guidelines on how to identify the financial break-even point in terms of the cost to the organization.

Table 1 (continued)
In addition, recently sentiment analysis has seen increased attention in various areas of software engineering research. Most studies involving sentiment analysis analyze the sentiment and emotions of various online artifacts, such as StackOverflow, Twitter and GitHub. One study by Graziotin et al.[5] analyzed the consequences of negative emotions and unhappiness, as opposed to positive emotions and happiness in software development activities in GitHub. They discovered multiple adverse repercussions related to unhappiness, including mental well-being, the software development process, and resulting output. Moreover, the vast majority of sentiment analysis research has focused on the polarity of the sentiment, i.e. positive versus negative sentiment. Others have focused on a particular emotion, as Gachechiladze et al. [6] implemented a machine learning classifier that detects anger direction by mining comments in Apache issue reports. Finally, Blaz et al. [7] developed a tool to help differentiate between the objective report of a problem and the polarity of the sentiment in IT support tickets across five organizations, and applied it to the analysis of internal IT support tickets (not commercial software products with external customers). Our previous investigations into sentiment analysis on customer comments in support tickets in a software organization[8] show promising results. We found a statistically significant difference in the customer sentiment between the escalated and non-escalated support tickets, which further demonstrates the usefulness in applying sentiment analysis tools to quantify sentiment and emotions in the software engineering domain, and in particular the potential that the customer sentiment data has in predicting ticket escalations.

Table 1 (continued)

References

[1] L. Montgomery and D. Damian. "What do Support Analysts Know About Their Customers? On the Study and Prediction of Support Ticket Escalations in Large Software Organizations". In: *2017 IEEE 25th International Requirements Engineering Conference (RE)*. 2017, pp. 362–371. DOI: 10.1109/RE.2017.61.

[2] Charles X. Ling et al. "Predicting Software Escalations with Maximum ROI". In: *Fifth IEEE International Conference on Data Mining (ICDM'05)*. Fifth IEEE International Conference on Data Mining (ICDM'05). Houston, TX, USA: IEEE, 2005, pp. 717–720. ISBN: 978-0-7695-2278-4. DOI: 10.1109/ICDM.2005.120. URL: http://ieeexplore.ieee.org/document/1565765/ (visited on 12/11/2018).

[3] Tilmann Bruckhaus et al. "Software escalation prediction with data mining". In: (2004).

[4] Victor S. Sheng et al. "Cost-sensitive Learning for Defect Escalation". In: *Know.-Based Syst.* 66.1 (Aug. 2014), pp. 146–155. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2014.04.033. URL: https://doi.org/10.1016/j.knosys.2014.04.033.

[5] Daniel Graziotin et al. "Consequences of Unhappiness While Developing Software". In: *Proceedings of the 2Nd International Workshop on Emotion Awareness in Software Engineering*. SEmotion '17. Buenos Aires, Argentina: IEEE Press, 2017, pp. 42–47. ISBN: 978-1-5386-2793-8. DOI: 10.1109/SEmotion.2017...5. URL: https://doi.org/10.1109/SEmotion.2017...5.

[6] D. Gachechiladze et al. "Anger and Its Direction in Collaborative Software Development". In: *2017 IEEE/ACM 39th International Conference on Software Engineering: New Ideas and Emerging Technologies Results Track (ICSE-NIER)*. 2017, pp. 11–14. DOI: 10.1109/ICSE-NIER.2017.18.

[7] C. C. A. Blaz and K. Becker. "Sentiment Analysis in Tickets for IT Support". In: *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)*. 2016, pp. 235–246. DOI: 10.1109/MSR.2016.032.

[8] C. Werner et al. "How Angry are Your Customers? Sentiment Analysis of Support Tickets that Escalate". In: *2018 1st International Workshop on Affective Computing for Requirements Engineering (AffectRE)*. 2018. DOI: 10.1109/AffectRE.2018.00006.

2 Approach

Our approach consists of 1.) performing sentiment analysis on conversations between customer and support analyst (referred to as customer conversations henceforth), 2.) evaluating machine learning algorithms trained on sentiment-related features to predict support ticket escalations, and then 3.) selecting a machine learning algorithm based on cost calculations of benefit to the organization. We describe each of these steps below, together with results we obtained from the application and evaluation of our approach at our industrial partner Alpha.

Alpha maintains a database of its software support tickets, from which, in confidentiality, a sample was made available to us for sentiment analysis. The data contained 356 support tickets and their associated customer conversations, amounting to a total of 10,172 comments in these conversations. The duration of a ticket ranged from a day to about 16 months, and had between 2 and 315 comments. The support ticket information contained an identification number, priority, severity, and a thread of comments – incoming (customer) or outgoing (support team) – in the customer conversation over the lifetime of the support ticket; information was also provided as to whether the support ticket was escalated, and the date of its escalation.

2.1 Sentiment Analysis on Customer Conversations

The first step in our approach is to analyze textual conversation data to evaluate the sentiment and emotion in the customer conversations. Sentiment is defined as the overall view, attitude, or feeling that is expressed and is typically measured in terms of positive, negative, or neutral sentiment [5]. To obtain a richer and deeper understanding, we also examined emotion, which is measured in terms of the five basic emotions (sadness, joy, fear, disgust, and anger) [6]. We utilized IBM Watson's Natural Language Understanding (NLU) tool for sentiment analysis, which has the capability to analyze both sentiment and emotion. For each support ticket, we computed the sentiment and emotion for each comment in its associated customer conversation. Thus, for each comment we obtained a value between -1 (negative) to 1 (positive) for sentiment and and a value between 0 (emotion not exhibited) and 1 (emotion exhibited) for each of the five emotions (sadness, joy, fear, disgust, and anger) [3]. In order to monitor the change in emotion as conversations progress in a support ticket, we also calculate a tendency value. Tendency captures how much the value of an individual metric (sentiment or emotion) increases or decreases over the lifespan of a support ticket as more comments accumulate.

2.2 Machine Learning to Predict Escalations

Once the sentiment analysis is performed on each support ticket, we evaluated 27 different machine learning algorithms to predict if a particular support ticket would be escalated (see Table 4 for full results). In our study at Alpha, the machine learning algorithms were trained on fourteen attributes (priority, severity, the average of anger, disgust, fear, joy, sadness, and sentiment, and the tendency for anger, disgust, fear, joy, sadness, and sentiment) to classify each support ticket.

The Alpha data set had 356 support tickets, 242 (68%) were escalated and 114 (32%) were not escalated. To overcome the data imbalance (68% escalated) we tested a variety of under and over sampling methods to create balanced training data [7] [2]. The results of evaluating the 27 machine learning algorithms are averaged across these methods and shown in Table 4. They range between 51% and 84% for accuracy, 65% and 88% for precision and 52% to 100% for recall.

To determine *how* early in the lifespan of a support ticket we could predict an escalation, we trained the algorithms on partial data, representing the 25%, 50%, and 75% of the comments for each support ticket. The results (also shown in Table 4) are promising, with the Bagging Random algorithm[3] having achieved 73% accuracy, 80% precision, and 83% recall with only 25% of the comments across all support tickets. Not surprisingly, better results are achieved as the conversations in the training data set are longer.

2.3 Choose Most Cost Effective Machine Learning Algorithm

An organization should choose the machine learning algorithm that provides the largest reduction in costs, as such we adopt a cost-calculation, applied by Ling et al., which reflects return on investment (ROI) [10]. Based on the predictions of a machine learning algorithm, results are categorized as true positives (TP), i.e. the escalated support tickets that were correctly predicted as escalations, and false positives (FP), i.e. the non-escalated support tickets that were incorrectly predicted as escalations. As such, we use

$$ROI = TP * (A1 - A2) - FP * A2 \tag{1}$$

Recall, A1 is the average cost of an escalation to an organization and A2 is the additional investigative cost of a support ticket flagged as a candidate for escalation. Essentially ROI represents the savings to an organization by using sentiment-based machine learning to predict support ticket escalations, thus representing the difference between reacting to incoming escalations (i.e. status quo) and being proactive in resolving a candidate for escalation prior to the actual escalations. Using this definition of ROI an organization can choose the machine learning algorithm that maximizes ROI. Furthermore, ROI *may* increase quite drastically, if the ratio of A1:A2 is substantially large, or decrease if the number of FP is substantially large.

The ROI for each algorithm run on the Alpha data set is shown in Table 4. The first observation is that the algorithm with the highest ROI (Bagging Random) increases from the 25% to 50%, 50% to 75%, and 75% to 100% data sets; this is not surprising as a machine learning algorithm performs better with more comments to train on.

Further, since ROI shows the difference between the value gained versus the investment cost of escalation prediction, $ROI = 0$ when an organization

[2] Random Under Sampler, Near Miss, One Sided Selection, Repeated Edited Nearest Neighbours, Random Over Sampler, SMOTE, ADASYN, SMOTEENN (SMOTE and ENN), and SMOTETomek (SMOTE and Tomek)

[3] A bagging algorithm in combination with Random Forest [8] [9]

reaches the *break-even point* for their investment in escalation prediction. If $ROI < 0$, an organization would incur more investment costs than value gained from escalation prediction. In contrast, if $ROI > 0$, an organization would experience cost savings as the value gained from escalation prediction surpasses the cost of using sentiment analysis for escalation prediction. Therefore, from Table 4 it can be seen that Alpha would achieve a relatively significant ROI regardless of the which algorithm is chosen.

Similarly, various A1:A2 ratios will result in significant ROI changes as shown by Figure 1 in Table 2. The figure depicts how ROI changes for four selected algorithms as A1:A2 change and the break-even point (i.e. $ROI = 0$) for each chosen algorithm. At Alpha, the Bagging Random machine learning algorithm would be chosen in the implementation of our approach as it would result in the highest ROI.

3 Practical Implications for Deployment

3.1 Steps to Deploy

An organization maintaining textual records of conversations between customers and support analysts can deploy our approach using these steps:

1. Within status-quo practice, calculate the cost of an escalation to the organization (A1) and the additional cost required to investigate a support ticket flagged as a candidate for escalation (A2). If $A1 > A2$ an organization may proceed to the next step. However, if $A1 < A2$ an organization may not realize any financial cost saving measures through our approach.
2. Extract and label support ticket data containing conversational text and whether it was escalated.
3. Analyze the conversational text with a sentiment analysis tool.
4. Calculate predictions using sentiment analysis results with machine learning algorithms.
5. Choose algorithm based on highest ROI.

3.2 Enhanced Support Process, Increased Customer Satisfaction, and Decrease in Escalations

Once an organization implements our sentiment-based escalation prediction approach, they have the ability to create a multitude of significant applications. These applications will help streamline and enhance the support process, increase customer satisfaction, and decrease the number of escalations. These competitive advantages can be achieved through these example applications:

- monitor, track, and visualize the trajectory and tendency of the sentiment and emotion across all support tickets in an organizational dashboard;

this can provide additional knowledge and statistical metrics to support staff and the organization through real-time analysis of the tickets. In particular, specific increases in emotions such as anger or disgust can be highlighted and be indicative of an escalation.

- at a lower level, provide sentiment and emotion metrics to a support analyst currently working on a particular support ticket; which provides the support analyst current and historical assessments of the customer's sentiment and emotion.

- improve escalation awareness and evaluation by analyzing escalation candidates early-on based on the sentiment of the customer; in our analysis at Alpha, analysis of partial data sets achieved accuracy as high as 73% with as little as 25% of amount of comments; with early-warning an organization has more time to respond to and prevent an escalation.

- develop customer relationship management strategies that leverage analysis of customer sentiment based on a procedure and knowledge management baseline to engage escalation candidates; with the ability to predict an escalation an organization is able to implement, measure, and assess responses and best practices to resolve an escalation candidate. For example, individual emotions can be associated with escalations and strategies can be developed to prevent those particular emotions.

Tab. 2: Choosing a Machine Learning Algorithm in Our Case Study at Alpha

To demonstrate how to compare machine learning algorithms using our approach, we chose the algorithm with highest ROI (Bagging Random), highest recall (Weighted SVM), highest precision (Weighted Random Forest), and lowest recall (Naive Bayes) from the 100% data set. Using hypothetical values of $A1 = \$10,000$ and $A2 = \$5,000$ we calculated ROI for each algorithm to compare. Since Ling et al. estimated in their previous work that the cost of fixing an escalation is 7 times more expensive than prematurely fixing the escalation [10], therefore, our A1:A2 ratio of 2 is relatively moderate (2:1).

Tab. 3: Results of Sample Algorithms With A1= $10,000 and A2 = $5,000

Algorithm	ROI	Acc.	Pre.	Rec.
Bagging Random	**$451,000.00**	**84%**	87%	90%
Weighted SVM	$307,500.00	66%	66%	**100%**
Weighted Random Forest	$418,053.57	81%	**88%**	83%
Naive Bayes	$334,000.00	71%	87%	66%

Table 3 demonstrates the usefulness of ROI, relative to recall or precision for four sample algorithms, as Bagging Random algorithm has the maximum ROI, yet having neither the highest recall nor the highest precision; however, it does have the highest accuracy.

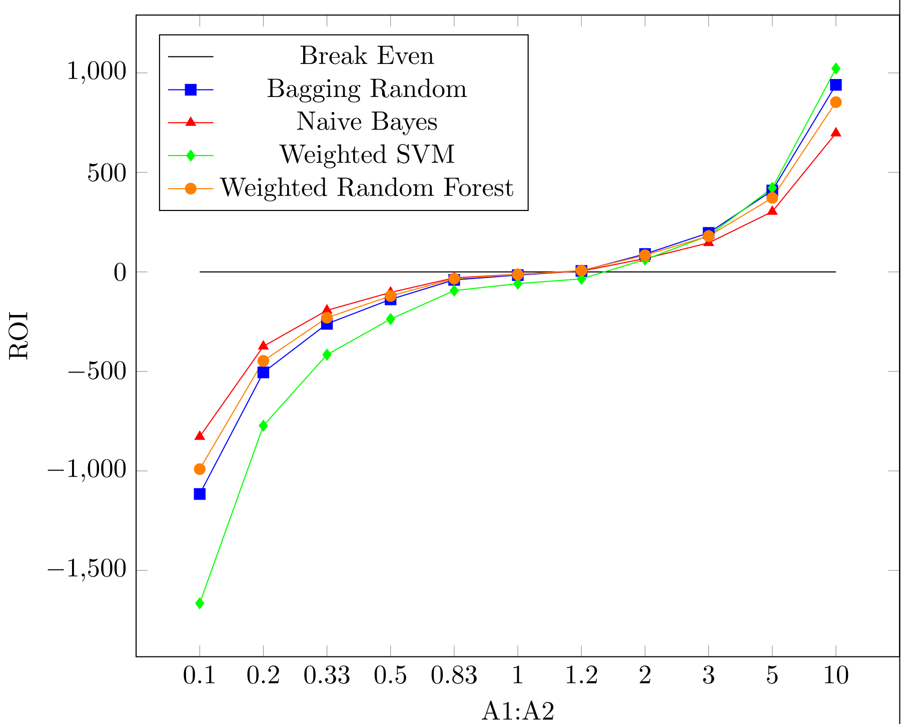

Fig. 1: Changes in ROI relative to $A1 : A2$ for Sample Algorithms

Figure 1 shows that when $A1 : A2$ is relatively small the ROI is negative; however, the ROI increases relative to the $A1 : A2$.

Tab. 4: Summary of Results ($A1 = 2; A2 = 1$)

	100				75			
Algorithm	Acc.	Pre.	Rec.	ROI	Acc.	Pre.	Rec.	ROI
Dummy	**51%**	66%	**54%**	**33**	**51%**	66%	**54%**	**33**
KNN1	77%	81%	85%	76	72%	78%	80%	67
KNN5	77%	82%	85%	77	73%	81%	78%	69
Weighted KNN1	77%	81%	85%	76	72%	78%	80%	67
Weighted KNN5	78%	82%	85%	78	73%	81%	77%	69
Log. Reg.	68%	76%	77%	62	67%	74%	75%	59
Weighted Log. Reg.	66%	66%	100%	62	66%	66%	100%	62
Perceptron	56%	69%	57%	40	56%	66%	61%	41
Weighted Perceptron	64%	68%	88%	57	64%	67%	92%	59
Naive Bayes	71%	87%	66%	67	67%	82%	62%	60
Dec. Tree	75%	81%	82%	75	72%	79%	78%	69
Weighted Dec. Tree	79%	84%	83%	80	74%	82%	79%	73
Rand. Forest	81%	86%	85%	84	78%	85%	81%	78
Weighted Rand. Forest	81%	**88%**	83%	84	78%	**86%**	79%	77
SVM	65%	69%	78%	57	65%	67%	77%	57
Weighted SVM	66%	**66%**	**100%**	62	66%	66%	**100%**	62
Linear SVC	67%	75%	77%	61	66%	74%	76%	58
Bagging KNeighbors	77%	82%	85%	78	72%	80%	77%	68
Bagging Log. Reg.	68%	75%	77%	61	66%	73%	75%	58
Bagging Perceptron	62%	76%	64%	51	61%	74%	64%	50
Bagging Naive Bayes	71%	87%	66%	67	66%	82%	62%	61
Bagging Dec. Tree	80%	87%	83%	83	78%	85%	80%	78
Bagging Rand.	**84%**	87%	90%	**90**	**82%**	85%	88%	**86**
Bagging SVM	64%	67%	75%	54	63%	**65%**	74%	52
AdaBoost	81%	83%	90%	87	74%	80%	80%	74
Gradient Boosting	83%	85%	90%	89	82%	84%	89%	85
Soft Voting	82%	86%	87%	87	78%	83%	83%	79
AVERAGE	72%	78%	81%	70	70%	77%	78%	65

	50				25			
Algorithm	Acc.	Pre.	Rec.	ROI	Acc.	Pre.	Rec.	ROI
Dummy	**51%**	66%	53%	**32**	**52%**	**71%**	**54%**	**33**
KNN1	65%	75%	71%	55	69%	79%	77%	55
KNN5	68%	80%	70%	60	67%	79%	73%	54
Weighted KNN1	65%	75%	71%	55	69%	79%	77%	55
Weighted KNN5	68%	80%	70%	60	68%	79%	75%	55
Log. Reg.	66%	74%	75%	59	67%	77%	79%	56
Weighted Log. Reg.	66%	66%	100%	62	71%	72%	99%	60
Perceptron	58%	67%	64%	47	58%	77%	61%	42
Weighted Perceptron	64%	67%	91%	58	68%	73%	89%	57
Naive Bayes	60%	79%	**52%**	49	61%	80%	60%	51
Dec. Tree	68%	76%	74%	63	65%	77%	73%	52
Weighted Dec. Tree	69%	79%	74%	65	63%	76%	70%	50
Rand. Forest	72%	81%	74%	68	70%	**80%**	77%	59
Weighted Rand. Forest	71%	**83%**	71%	66	66%	80%	70%	53
SVM	64%	**65%**	75%	55	68%	76%	83%	57
Weighted SVM	66%	66%	**100%**	62	71%	71%	**99%**	60
Linear SVC	65%	73%	75%	58	65%	75%	76%	52
Bagging KNeighbors	68%	80%	69%	60	67%	79%	74%	55
Bagging Log. Reg.	66%	73%	75%	58	67%	77%	79%	55
Bagging Perceptron	59%	73%	60%	45	62%	77%	67%	49
Bagging Naive Bayes	60%	79%	53%	50	61%	80%	61%	51
Bagging Dec. Tree	72%	82%	74%	69	68%	79%	76%	57
Bagging Rand.	**75%**	81%	82%	**75**	**73%**	80%	83%	**64**
Bagging SVM	63%	66%	73%	52	67%	76%	80%	55
AdaBoost	67%	76%	73%	61	65%	77%	74%	54
Gradient Boosting	74%	79%	81%	72	72%	79%	82%	61
Soft Voting	71%	79%	75%	68	70%	79%	80%	60
AVERAGE	66%	75%	73%	59	66%	77%	76%	54

References

[1] Victor S. Sheng et al. "Cost-sensitive Learning for Defect Escalation". In: *Know.-Based Syst.* 66.1 (Aug. 2014), pp. 146–155. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2014.04.033. URL: https://doi.org/10.1016/j.knosys.2014.04.033.

[2] Tilmann Bruckhaus et al. "Software escalation prediction with data mining". In: (2004).

[3] C. Werner et al. "How Angry are Your Customers? Sentiment Analysis of Support Tickets that Escalate". In: *2018 1st International Workshop on Affective Computing for Requirements Engineering (AffectRE)*. 2018. DOI: 10.1109/AffectRE.2018.00006.

[4] N. Novielli, A. Begel, and W. Maalej. "Introduction to the special issue on affect awareness in software engineering". In: *The Journal of Systems & Software* 148 (Feb. 2019), pp. 180–182. DOI: 10.1016/j.jss.2018.11.016. URL: https://doi.org/10.1016/j.jss.2018.11.016.

[5] Bing Liu. "Sentiment analysis and subjectivity". In: *Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca.* 2010.

[6] P. Ekman, W.V. Friesen, and P. Ellsworth. *Emotion in the human face: guide-lines for research and an integration of findings.* Pergamon general psychology series. Pergamon Press, 1972. URL: https://books.google.ca/books?id=MeB9AAAAMAAJ.

[7] Jason Van Hulse, Taghi M. Khoshgoftaar, and Amri Napolitano. "Experimental Perspectives on Learning from Imbalanced Data". In: *Proceedings of the 24th International Conference on Machine Learning*. ICML '07. Corvalis, Oregon, USA: ACM, 2007, pp. 935–942. ISBN: 978-1-59593-793-3. DOI: 10.1145/1273496.1273614. URL: http://doi.acm.org.ezproxy.library.uvic.ca/10.1145/1273496.1273614.

[8] Leo Breiman. "Bagging predictors". In: *Machine learning* 24.2 (1996), pp. 123–140.

[9] L Breiman. "Random Forests Machine Learning. 45: 5–32". In: *View Article PubMed/NCBI Google Scholar* (2001).

[10] Charles X. Ling et al. "Predicting Software Escalations with Maximum ROI". In: *Fifth IEEE International Conference on Data Mining (ICDM'05)*. Fifth IEEE International Conference on Data Mining (ICDM'05). Houston, TX, USA: IEEE, 2005, pp. 717–720. ISBN: 978-0-7695-2278-4. DOI: 10.1109/ICDM.2005.120. URL: http://ieeexplore.ieee.org/document/1565765/ (visited on 12/11/2018).