From Explaining How Random Forest Classifier Predicts Learning of Software Engineering Teamwork to Guidance for Educators

Dragutin Petkovic, Sabiha H Barlaskar, Jizhou Yang
Department of Computer Science, San Francisco State University, U.S.A.
petkovic@sfsu.edu; sbarlaskar; jyang15@mail.sfsu.edu

Rainer Todtenhoefer
Department of Applied Computer Science, University of Applied Science, Fulda, Germany
rainer.todtenhoefer@informatik.hs-fulda.de

Abstract— **This *Research Full Paper* builds on our previous research in Software Engineering (SE) Teamwork Assessment and Prediction project (SETAP) where we used Random Forest (RF) classifier to predict with over 70% accuracy the student learning effectiveness in software engineering teamwork based on 115 objective and quantitative Team Activity Measures (TAM). These TAM measures have been obtained from monitoring and measuring activities of 74 student teams during the creation of their final class project in a joint software engineering classes which ran concurrently at three universities (San Francisco State University, Fulda University and Florida Atlantic University) over the period of four years and, together with team outcomes, have been collected in publicly available SETAP database. In this paper, we provide much deeper analysis of how and why RF made its decisions, namely we address the *explainability* of our RF classification. We also provide in-depth analysis of differences in grading and behavior of local and global student teams (composed of students from multiple schools). We then use these insights to provide concrete and practical guidance to educators teaching SE teamwork in local and global classroom setting.**

Keywords — Assessment, Machine Learning, software engineering teamwork education, Random Forest, explainability, global software engineering

1. INTRODUCTION AND BACKGROUND

In this paper, we present final results and teaching recommendatioins from our research in Software Engineering (SE) Teamwork Assessment and Prediction project (SETAP) [1,2,3] where we used Random Forest (RF) classifier [4] to predict with over 70% accuracy the student learning effectiveness in software engineering teamwork based on over 100 objective and quantitative Team Activity Measures (TAM). These TAM measures have been obtained from monitoring and measuring activities of 74 student teams during the creation of their final class project in a joint software engineering classes which ran concurrently at San Francisco State University (SFSU), Fulda University, Germany, and Florida Atlantic University (FAU) over the period of four years (from 2012 to 2015). These TAM measures, together with team outcomes (e.g. instructors' team grades) have been collected in now publicly available SETAP Machine Learning (ML) training database [5]. We used RF classifier trained on this training database to predict student learning of SE teamwork, as reported in [1, 2]. In this paper, we provide much deeper analysis of how and why RF made its decisions, namely we address the *explainability* of our RF classification. We also analyze our data in terms of differences in grading and behavior for *global* student teams (those composed of students from multiple universities) and *local* student teams. We then use these insights to provide concrete and practical guidance to educators teaching SE teamwork in both local and global classroom settings.

SETAP data are collected from a joint software engineering class taught concurrently at above three universities where student teams at all three schools were "embedded and observed" in as realistic project and teamwork development environment as possible, thus providing a rich learning environment for students and more realistic data for researchers. This joint class involved about 140 students each year, working in 25-30 teams of 5-7 students each. *Local* student teams were composed of students from the same university, and *global student teams* were composed of volunteer students from multiple—usually two— universities. Each student team developed the same software application each semester. The semester was divided into five formally managed milestones, M1 through M5, which were synchronized across all three schools (M1 – high level specs; M2- detailed specs; M3 – first prototype development; M4 – beta launch, QA and usability review; M5 – final delivery). TAM Data is collected separately in each of these time intervals T1-5 corresponding to M1-M5. Details about class teaching, team selection and data collection in our joint SE class have been reported in [2, 3]. Data collection and analysis was done at SFSU.

Our focus was on predicting team and not individual student performance, therefore the SETAP training database is organized by student teams, and consists of Team Activity Measures (TAM) data (or features) for each student team paired with separate evaluations of software engineering teamwork learning outcomes, one for *SE Process* and one for SE *Product* component of the team grade [2]. These outcomes, for the purposes of this research, are categorized in two ML *classes* for SE Process and SE Product separately: "A" ML class represents team performance at or above expectations, and "F" ML class represents team performance below expectation or needing attention. We then try to predict these two classes A and F using observed TAM data, and we are aiming specifically for discovering teams going to fail, e.g.

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

class F. TAM data comprise 115 measures, such as: a) measurements of student weekly activity via on-line time cards; b) instructor observations; and c) analysis of software (SW) development tools logs, plus some other basic data (team size, global/local type etc.). TAMs are grouped in above mentioned time intervals Ti to reflect different dynamics of student work (e.g. low level of coding in early stages) [2]. To protect students' privacy, SETAP data contains no individually identifiable student information. In order to allow other researchers to perform their own analysis on our difficult to collect SETAP database we made it publicly available UC Irvine Machine Learning Repository [5].

1.1 Our goals

Our work reported in [2] was just the beginning. While we obtained very good RF prediction results (e.g. prediction of grades A or F based on TAMs), many important questions remained which we address in this paper. Specifically (and separately for SE Process and SE Product prediction) we drive our analysis by user (educator) needs in the form of questions they would pose in order to understand and adopt our findings and improve their teaching of SE teamwork:

- Can we provide more in-depth analysis of how RF made its decisions and more insights into specific TAMs being most powerful predictors in order to increase the confidence of potential adopters?
- Is it possible to develop tradeoffs between using small subset of TAMs vs. achievable prediction accuracy in order to enable adopters to apply our findings more effectively by measuring or monitoring fewer TAMs?
- Are there differences among global and local teams in terms of overall grades as well as their behavior reflected in TAM measures and the grading (e.g. do global teams do worse than local)?
- Do teams rated F in SE Process also tend to produce poorer product deliveries e.g. get poorer grades in SE Product?
- And finally, our ultimate goal: Can all this analysis result in concrete and easy to implement recommendations for educators of SE teamwork (local and global) that will enable them in early discovery of student teams bound to fail?

Many of above questions relate to very current topic of Explainable ML [see for e.g. 6] where the goal is to better understand why and how ML reached its decisions in order to increase user confidence and its adoption as well as enable new discovery. In this work we leverage our recent research in this area, namely development of Random Forest Explainability (RFEX) method [7] successfully applied in biomedical application, as well as some additional analysis for local vs. global teams. This paper aims then to answer above questions and from this analysis provide a set of practical guidelines for educators of SE teamwork.

1.2 Related work

ML and data mining have been increasingly used in educational research [8] for a variety of applications ranging from trying to predict retention, grades as well as assessment of student performance [8- 12]. In addition, ML is also starting to be used in software SE to analyze quality of the code or SE process, for example as in [13, 14]. General idea is to collect the data/measurements with known "ground truth" e.g. outcomes, and then apply ML and data mining algorithms to try to predict desired outcomes based on these measurements.

For our ML we use RF ML method [4] due to its power, popularity, availability of tools (we use toolkit R [16]), our experience with it, as well as because it provides certain level of explainability of how it works (e.g. with feature ranking), which we enhance and leverage further in our work described below. RF is an ensemble classifier, consisting of a set of decision trees (CART) classifiers forming a "forest" of classifiers voting for a particular class. To train a RF, two parameters, the number of decision trees (*ntree*) in the forest and the number of randomly selected variables (TAMs in our case) used to evaluate at each decision tree node (*mtry*) must be optimized during the training phase using the training database. RF also allows adjustment of the voting threshold or *cutoff* (fraction of trees in the forest needed to vote for a given class), which we have also exploited in this study to adjust RF sensitivity. For accuracy estimate as well as for optimizing or training RF, (namely finding optimal *ntree, mtry, cutoff)*, we perform grid search over those parameters and choose those which produce best F1 score defined as 2*(recall*precision)/(recall + precision).), where recall refers to class of our interest, namely F. F1 score provides better accuracy estimate than traditional OOB (out of bag error) [4, 15]. One of the RF algorithm's strengths, and a reason we chose it, is its ability to calculate various feature/variable importance measures which can form the basis for enhancing its explainability [4]. We use RF's *MDA (mean decrease in accuracy)* as our main feature importance (ranking) measure. MDA measures the average increase of the error rate (i.e. decrease of accuracy) against random permutation of feature values across all cases, and can be committed separately for each class (e.g. A and F).

Recently, a consensus is emerging that we need more explainable ML in order to understand why and how ML systems make their decisions [e.g. recent workshop on explainable ML in biomedical area [6]). Better ML explainability promises not only to increase the confidence of adopters and users in applying such systems but also to offer more insights and new knowledge by analyzing details of ML decision making. This is also specifically mentioned in related work in educational context [10] as a necessary condition for such ML approaches to be useful. RF classifiers lend themselves to explainability being tree-based. Current methods for RF explainability fall into two basic categories. *Feature ranking* uses RF-provided variable importance measures like e.g. RF-provided Gini, MDA (mean decrease in accuracy) or others, to present them in *tables or horizontal bar charts* sorted by chosen variable importance measure [4, 7]. Highly ranked features are assumed to play important role RF predictions, which in turn may offer more insights into the observed classification process. The second basic approach is *rule extraction from trained RF*. This method consists of: a) performing standard RF training; b) defining rules by analyzing trained RF trees (resulting in a very large set of rules, order of 100 K); and c) reducing the number and complexity of extracted rules by optimization e.g. minimizing some metrics (accuracy, coverage, rule complexity etc.) to reduce to 10s – 100s of rules, each with 1-10 or so conditions [7]. Common problem with this approach is large number of

complex rules hard to interpret by humans and lack of tradeoffs between accuracy and number of rules used.

To improve explainability of RF classifiers we developed Random Forest Explainability method (RFEX) which extracts explainable model and easy to understand one page RFEX summary from trained RF, and has been successfully applied to biomedical data [7]. RFEX is designed with non-ML expert users in mind. It provides deeper and simpler to understand insights into RF decisions such as detailed analysis of factors (features) and tradeoffs in using subsets of features vs. accuracy, as well as other explainability measures (e.g. which variables co-occur together to make predictions). In [7], we also evaluated our RFEX approach with 13 expert and non-expert users to demonstrate that it increases their confidence and understanding of how RF classified biomedical data. In this paper, we apply our RFEX method to SETAP data and use it to develop deeper insights and recommendations for educators.

2. EXPERIMENTAL RESULTS OF APPLYING RFEX METHOD AND DATA ANALYSIS TO SETAP DATA

The RFEX method [7] consists of a set of steps applied to RF classifier already optimized/trained on SETAP data. In this section, we present each of these RFEX steps. Discussion of these results will be presented in subsequent section.

RFEX Step 1: Establish Base RF Accuracy using all TAM features

This is a basic step performed by all RF practitioners to determine base accuracy of RF. It consists of determining optimal values of *ntree, mtry* and *cutoff* parameters (using grid search) to achieve best RF accuracy e.g. highest F1 score computed as 2*(recall*precision)/(recall + precision). This step has already been presented in [2] using all 115 TAMs, with accuracy reported as F1 score of 0.63 for SE Process in interval T2, and 0.70 for SE Product in interval T3. We have verified this accuracy independently with three students on two different RF tools.

RFEX Step 2: Rank features/variables according to their predictive power

The next step in RFEX method is to rank features/variables by their predictive power. In our case we use MDA (Mean Decrease in Accuracy) measure provided with standard RF implementations [15,16]. This measure involves random permutation of features and measurement of overall decrease in RF accuracy due to this permutation, with those features producing largest decrease in accuracy deemed most important [4]. We then provide top MDA ranked features where we recommend using only about top 5-10% of variables (TopN) for SE Process and SE Product predictive analysis, shown in Tables 1 and 2. Notably, MDA rankings are very stable wrt. to different random runs of RF, which we tested and confirmed.

Rank	TAM features	MDA
1	lateIssueCount	18.6
2	issueCount	6.5
3	helpHoursStandardDeviation	6.1
4	standardDeviationMeetingHoursAverageByWeek	4.6
5	standardDeviationHelpHoursTotalByWeek	4.4
6	codingDeliverablesHoursAverage	4.2
7	averageNonCodingDeliverablesHoursAverageByWeek	3.8
8	averageCodingDeliverablesHoursAverageByWeek	2.97
9	averageResponsesByWeek	2.91
10	averageCodingDeliverablesHoursAverageByStudent	2.5

Table 1: MDA ranking of TAM features for SE Process in time interval T2

Rank	TAM features	MDA
1	uniqueCommitMessageCount	9.9
2	averageUniqueCommitMessageCountByWeek	9.5
3	commitMessageLengthTotal	4.73
4	standardDeviationInPersonMeetingHours AverageByWeek	4.72
5	standardDeviationCodingDeliverables HoursTotalByWeek	4.5
6	standardDeviationCodingDeliverablesHours AverageByWeek	4.2
7	standardDeviationUniqueCommitMessage PercentByWeek	4.05
8	meetingHoursTotal	3.6
9	standardDeviationNonCodingDeliverablesHours AverageByStudent	2.95
10	averageCodingDeliverablesHoursTotalByStudent	2.93

Table 2: MDA ranking of TAM features for SE Product in time interval T3

RFEX Step 3: Provide tradeoffs between features used and accuracy

This is a critical step and very much required for any explainable ML model. Its goal is to reduce complexity (e.g. using feature reduction) and to provide tradeoffs where user can observe (and chose) which subsets of features produce what accuracy, and then make a choice of using less features for a given loss of accuracy. In our experience in [7] (and confirmed in this work) using only a few (e.g. 5%) of top ranked features achieves accuracy close to base RF accuracy when all features are used. For accuracy we use F1 score computed for class F. This step is performed as follows:

Let TopN be the number of features ranked highly by MDA in Step 2 (TopN = 20 in our case)

For K = 2 to K=TopN
Train RF using only K top ranked features (vary ntree, mtry and cutoff to get max F1)
Record maximal F1(K) for set of K top ranked features

One can then observe relationship and tradeoffs between K and F1(K) and determine which K is sufficient for desired accuracy F1(K). In Fig. 1 and 2 we present the relationship between using top ranked K features and RF accuracy F1(K) for SE Process and SE Product separately. Note that only a few (about 10 from 115) top ranked TAM features produce accuracy very close to one obtained by using all TAMs – a critically important feature reduction with practically no loss of accuracy.

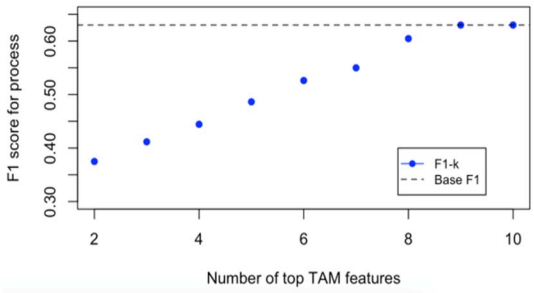

Fig 1: Tradeoff between accuracy F1(K) using all TAM features vs. using only K top ranked features for *SE Process*. Base F1 score (0.63) using all TAM features is at dotted line

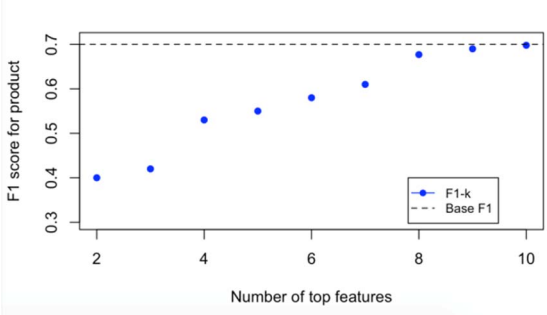

Fig 2: Tradeoff between accuracy F1(K) using all TAM features vs. using K top ranked features for *SE Product*. Base F1 score (0.7) using all TAM features is at dotted line

RFEX Step 4: Determine class-specific feature properties

In this step we present a simple statistics (others like RANGE could also be used) where we show average and standard deviation (AV/SD) of feature values for class F vs. A in chosen time interval (T2 for SE Process, T3 for SE Product). This gives the user a sense of what to expect from teams graded F vs. A in terms of actual class-specific TAM values. Good separation of these values increases user confidence of usefulness of this particular TAM/feature plus informs users if "*more of it*" or "*less of it*" is desired. This information is presented in Tables 3 and 4 for SE Process and SE Product prediction separately.

RFEX Step 5: Present RFEX results (e.g. RFEX Explainable Model) in simple and easy to understand way

Finally, RFEX Explainable Model summary of how RF classifies SETAP data is presented in easy to understand one page format in Tables 3 and 4 for SE Process and SE Product respectively.

TAM feature name	MDA value	F1 score Using top K TAM features	AV/SD For all Teams rated F	AV/SD For all teams rated A
lateIssueCount	18.6	N/A	0.68/0.7	0.06/0.2
issueCount	6.5	0.375	2/1.7	0.8/0.8
helpHoursStandard Deviation	6.1	0.42	0.95/0.62	1.12/0.6
standardDeviation Meeting HoursAverageByWeek	4.6	0.44	1.11/1.3	0.95/0.5
standardDeviation HelpHoursTotalByWeek	4.4	0.48	1.7/0.97	2.3/1.12
codingDeliverables HoursAverage	4.2	0.52	1.18/0.9	1.9/0.9
averageNonCoding DeliverablesHoursAverageByWeek	3.8	0.55	2.2/0.95	1.88/0.6
averageCodingDeliverablesHoursAverage ByWeek	2.9	0.60	1.3/1.06	1.5/0.9
averageResponses ByWeek	2.91	0.63	4.96/2.0	5.6/1.67
averageCoding DeliverablesHoursAverageByStudent	2.5	0.63	1.23/1.0	1.5/0.96

Table 3: RFEX explainable model for SE Process: Top 10 ranked TAM features for SE process with their MDA values, best F1 score and the average/standard deviation for teams rated F vs. teams rated A, all for interval T2.(IssueCount is number of times team asked for help or instructors requested a response; lateIssueCount is number of times the team was late in some official response to instructors' request)

TAM feature name	MDA value	F1 score Using top K TAM features	AV/SD For all teams rated F	AV/SD For all teams rated A
uniqueCommit MessageCount	9.9	N/A	72.3/39.2	114.4/95.
averageUniqueCommit MessageCountByWeek	9.5	0.4	18.5/9.3	28.4/17
commitMessageLength Total	4.73	0.42	5698.4/ 3590.3	7329.7/ 5741.4
standardDeviationInPerson MeetingHoursAverage ByWeek	4.72	0.53	1.02/0.56	0.73/0.6
standardDeviationCoding DeliverablesHoursTotal ByWeek	4.5	0.55	10.9/7.23	7.22/6.6
standardDeviationCoding DeliverablesHoursAverage ByWeek	4.2	0.58	2.3/1.33	1.5/1.25
standardDeviationUnique CommitMessagePercent ByWeek	4.05	0.61	0.14/0.08	0.10/ 0.07
meetingHoursTotal	3.6	0.68	44.2/23.8	39/27.42
standardDeviation NonCoding DeliverablesHoursAverage ByStudent	2.95	0.69	0.7/0.4	0.85/0.6
averageCodingDeliverables HoursTotalByStudent	2.93	0.698	13.5/9.0	10.9/7.9

Table 4: RFEX explainable model for SE Product: Top 10 ranked features for SE product with their MDA values, best F1 score and the average/standard deviation for teams rated F and teams rated A, all for interval T3. (Variable names are self describing)

RFEX tables 3 and 4 offer easy analysis for potential adopters of this RF prediction. First, one can easily see what

TAM variables are most important in decision making since they are ranked. Then, one can easily determine which top K variables to use for desired accuracy, e.g. using only top 8 TAMs one can achieve F1 score very close to the one using all 115. Finally, by observing simple TAM variable statistics for F vs. A class one gets an idea of their respective range. More on this will be presented in the discussion section.

2.1 Differences between global and local teams

In this subsection we show data on differences in grading and teamwork behavior between local (all students in one school) vs. global teams (students from different schools worked together on same project) which will be analyzed in discussion section. We use this data to answer some of the questions outlined in section 1.1.

SETAP data contains TAMs and related grades of SE Process and SE Product for total of 74 teams - 59 local and 15 global. Grades for all 74 local and global teams were as below:

SE Process grades	SE product grades
49 A; 25 F	42 A; 32 F

Table 5: Grade distribution for all 74 teams

The Table 6 below shows differences in grades for SE Process and SE Product for local vs. global teams.

	SE Process Grades	SE Product Grades
Local teams	**43 As**; 16 Fs (**A's were 73%** of all 59 local teams)	**34 As**; 25 Fs; (**A's where 58%** of 59 local teams)
Global teams	**6 As**; 9 Fs (**A's were 40%** of all 15 global teams)	**8As**; 7 Fs (**A's were 53%** of all 15 global teams)

Table 6: Grade differences between local and global teams (SE Process and SE Product)

The table 7 below shows how well getting grade A or F grade in SE Process "predicts" getting grade A or F in SE Product

	SE Process grade was A	SE Product grade was F
If SE Process Grade was A	30	19
If SE Process Grade was F	12	13

Table 7: The table shows how many students got A (or F) for SE Product if they had A (or F) in SE Process grade, for all 74 teams

Finally, the tables 8 and 9 below show AV/SD values for top 5 ranked TAM features, separately for local vs. global teams, for SE Process in interval T2 and SE Product in interval T3:

TAM Feature name	Local teams AV/SD	Global teams AV/SD
lateIssueCount	0.2/0.4	0.6/0.7
issueCount	1/1.3	1.7/1
helpHoursStandardDeviation	1/0.4	1.3/1.1
codingDeliverablesHoursAverage	1.4/0.9	1.3/0.9
standardDeviationHelpHoursTotalByWeek	2.2/1.1	1.6/1

Table 8: Ranges of top predicting TAMs for local vs. global teams for SE Process data for time interval T2

TAM Feature name	Local teams AV/SD	Global teams AV/SD
uniqueCommitMessageCount	95.7/82.3	97.9/66.2
averageUniqueCommitMessageCount ByWeek	24/15.6	24.9/15.3
commitMessageLengthTotal	6670.2/5112.4	6443.6/4495.3
standardDeviationInPersonMeeting HoursAverageByWeek	0.83/0.54	0.99/0.84
standardDeviationCodingDeliverables HoursTotalByWeek	9.6/7.35	5.9/5.2

Table 9: Ranges of top predicting TAMs for local vs. global teams for SE Product data for time interval T3

3. DISCUSSION AND RECOMMENDATIONS FOR EDUCATORS

In this section we provide discussion of the results in order to provide concrete and simple recommendations for educators in SE engineering teamwork based on the analysis presented in this paper. In addition to recent attention to teamwork teaching and assessment in SE education, increased attention has been paid to teaching and assessment of student teams working in different locations (e.g. global student teams) since this is one of the common ways SE teams are being organized in industry [17,18]. Our recommendations (and data we collected in SETAP database) notably apply both to SE education of local and global student teams.

First, for any ML system to gain trust of potential adopters, it has to be reproducible. We have verified this by reproducing basic accuracy results from our previous work [2] by three students using different RF toolkits. We have also published our SETAP data in UC Irvine ML Repository [5] so it can be used by others. Overall, RF prediction of SE Process and SE Product teams that need attention or are likely to fail (graded F) is good and is best performed in early stages of the class (periods T2 and T3) which is significant and helpful for early intervention. Note that we applied very intense coaching of problematic teams after mid-course of the class (after T3 period) which in turn prevented many of teams ending up with grade F. Have we not done this, besides not doing our job as educators, we would have had many more teams rated F and would likely be able to show even better RF prediction for Class F.

Our confidence in RF classification has significantly increased by observing Tables 1 and 2 with top MDA ranked TAMs for SE Process and SE Product since top TAM features with large MDA rank values do make intuitive sense. (We repeated these rankings several times and got very similar rank order, further increasing our confidence). For SE Process TAMs with largest MDA were *lateIssueCount* (number of times the team was late in some official response to instructors' request or delivery) and *issueCount* (number of times team asked for help or instructors requested a response) and they certainly make sense and correlate with our intuition. Note that these variables were also part of the grading rubric (e.g. determined instructors' grading of A or F), hence there may be a possibility of some bias, but also note that the grading rubrics had many other parameters in turn minimizing this bias. For SE Product the highest ranked variables with largest MDA were *uniqueCommitMesageCount* and *AverageUniqueCommitMessageCountbyWeek*, related to commit comments for code submission, a surprising finding which in hindsight makes a lot of sense and correlates with observations and experience of SE managers (many teams graded F used "update" or empty as a standard commit message comment, e.g. not a unique nor a useful one – a very poor SE practice). In this case there was no bias wrt. instructors' grading (e.g. assignment of A and F class labels) since these variables were not directly part of the grading rubric.

Fig.1 and Fig. 2 offer very important insight, confirmed in our previous RFEX work [7], that only a few (5-10%) of top ranked features offer most of predicting power. This has significant practical implications in that it tells us that by observing only top few ranked TAMs one can achieve accuracy comparable with using all 115 TAMs, thus making application of this more economical and effective.

RFEX Explainable Model Summaries, in Tables 3 and 4 (for SE Process and SE Product respectively) effectively communicate all necessary information, namely they list top ranked TAMs, with MDA values and importantly show how many of those top K TAMs one needs in order to achieve desired F1 accuracy score (column 3) allowing easy tradeoffs between smaller set of features used and desired accuracy. Finally, they show TAM AV/SD values for F vs. A teams, so one can get a sense of whether those variables are *high* or *low* in their class-specific values. For example, for SE Process, Table 3 reveals that indeed teams graded F have much more issues and late issues (TAMs ranked 1 and 2) and that they spend much less on coding (TAM ranked 6) than teams graded A. For SE Product Table 4 reveals that teams graded F have less postings and less unique commit comment messages (many simply used "update" for commit comments) and that the length of their commit messages was low compared to teams graded A (TAMs ranked 1-3). These observations directly drive our recommendations for educators, see below.

In terms of differences between local and global teams we observed some important patterns. First, Table 6 shows that local teams get better SE Process grades than local teams (73% of local teams got in SE Process vs. 40% for global teams) which is to be expected given complexity of global team organization. However, they get approximately equal distribution of A and F grades for SE Product meaning that global teams achieve about the same quality of delivered SW, in part influenced by the fact that usually more skilled students volunteer for global teams.

We also wanted to know how well we could "predict" that teams who got F in SE Process will also get F in SE Product, an important hypothesis in SE engineering. From Table 7 we see that among 25 teams who got F in SE Process, 13 teams got F in SE Product or 52%, compared with 32 teams out of 74 or 43% among all teams (Table 5). So, using grade in SE Process to "predict" grade in SE Product offers some slight advantage, increasing the prediction from 43% to 52% or about 21% improvement. There is one explanation for the above relatively low improvement in this prediction. As we published before [2], in order to be able to measure "negative signals" e.g. behavior of teams bound to fail e.g. those who could get F in SE Product grade, until the end of Milestone 3 (T3 interval) e.g. by the middle of the class, we have not applied intense coaching. Then, in order to help those teams we applied intensive coaching and mentoring (reviews, checkpoints etc.) in the second half of the class and thus helped many teams move from grade F to a grade A in SE Product. This in turn reduced number of final grades F in SE Product and consequently reduced the prediction power of SE Process grade as measured above. In terms of learning outcomes this was a positive development because we helped many teams to learn and improve.

Finally, Table 8 shows that local teams had much less issues and late issues than global teams, a natural thing to except given complexity of global team collaboration. However, table 9 then shows that in terms of SE Product top ranked TAMs', global and local teams had very similar "SW development activity and behavior", which correlates well with the fact that their product grades were approximately equal in terms of percentages.

3.1 Recommendations for educators of SE teamwork

We finally derive recommendations for educators in SE teamwork in terms of concrete strategies to identify student teams that might fail and need early intervention, both in local and global classroom environment. We are focusing only on recommendations we derive from the analysis in this paper, namely from RFEX Explainable Models in Tables 3 and 4, and from Tables 8 and 9.

Early interventions and mentoring of student teams in order to identify teams that might fail proves to be critical, indicated by best RF predictions in intervals T2 (design phase) and T3 (early implementation). We recommend the following guidance and recommendations to be applied starting *early* in the SE class, namely in SW design phase and early in implementation/coding phases.

- Formally schedule number of tasks with strict deadlines and deliverables and observe teams who fail to deliver on time early in the class – they are candidates to watch and apply early intervention
- Verify that teams start coding early after initial design, those who fail to start early likely need intervention
- Verify (and demand) that students use meaningful, unique and rich enough comments to code commits and submissions

- Monitor distribution of code submissions in the team and ensure they are similar among team members e.g. verify that the coding is not concentrated to only a very few student coders in the team
- Ensure all of the above observations are recorded, made part of assessment and grading rubrics and communicate this to students early
- Provide more coaching to global teams since even those teams producing good SW delivery seem to need more attention.
- It is critical that instructors of global student teams are coordinated in terms of project objectives and coaching messages and participate together and in a timely manner in resolving conflicts and miscommunication

4. REFERENCES

[1] Dragutin Petkovic: "Using Learning Analytics to Assess Capstone Project Teams", IEEE Computer, Issue No.01 - Jan. (2016 vol.49).

[2] D. Petkovic, M. Sosnick-Pérez, K. Okada, R. Todtenhoefer, S. Huang, N. Miglani, A. Vigil: "Using the Random Forest Classifier to Assess and Predict Student Learning of Software Engineering Teamwork" Frontoiers in Education FIE 2016, Erie, PA, 2016

[3] D. Petkovic, M. Sosnick-Pérez, S. Huang, R. Todtenhoefer, K. Okada, S. Arora, et. al,: "SETAP: Software Engineering Teamwork Assessment and Prediction Using Machine Learning", Proc. FIE2014, Madrid, Spain 2014

[4] L. Breiman, "Random Forests," Machine Learning 45(1). 2001. pp. 5–32.

[5] SETAP database at UC Irvine Machine learning Archive ([https://archive.ics.uci.edu/ml/datasets/Data+for+Software+Engineering+Teamwork+Assessment+in+Education+Setting]).

[6] D. Petkovic (Chair), L. Kobzik, C. Re: "Workshop on Machine learning and deep analytics for biocomputing: call for better explainability", Pacific Symposium on Biocomputing PSB 2018, Hawaii

[7] D. Petkovic, R. Altman, M. Wong, A. Vigil: "Improving the explainability of Random Forest classifier – user centered approach", Pacific Symposium on Biocomputing PSB 2018, Hawaii

[8] Alejandro Peña-Ayala: "Review: Educational data mining: A survey and a data mining-based analysis of recent works", Int. Journal of Expert Systems with Applications, Vol 14, Issue 4, March 2014

[9] Khoa Le, Caslon Chua, Rosalind Wang, "Mining Software Engineering Team Project Work Logs to Generate Formative Assessment", 2017 24th Asia-Pacific Software Engineering Conference Workshops

[10] D. Delen: "A comparative analysis of machine learning techniques for student retention management", Decision Support Systems, Volume 49, Issue 4, November 2010, Pages 498–506

[11] A. Elbadrawy, A. Polizou, Z. Ren, M. Sweeney, G. Karypis, H. Rangwala:" Predicting Student Performance Using Personalized Analytics", IEEE Computer, Volume: 49, Issue: 4, Apr. 2016

[12] L. P. Macfayden, S. Dawson:"Mining LMS data to develop an "early warning system" for educators: a proof of concept", Comput Educ 54: 588-599 (2010)

[13] J. Moeyersoms, E, Junque de Fortuny, K. Dejaeger, B. Baesens, D. Martens: " Comprehensive software fault and effort prediction: A data mining approach", Journal of Systems and Software, Vol 100, Feb 2015

[14] M. Kym, T. Zimmermann, R. DeLine,A. begel :" The Emetgiong Role of Data Scientist on Software Teams", 2016 IEEE/ACM 38th Intenational Conference on Sofware Engineering (ICSE), May 2016

[15] C. Chen, A. Liaw, L. Breiman, Leo, " Using Random Forest to Learn Imbalanced Data", Report ID: 666, UC Berkeley, July 2004

[16] R Core Team, R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. 2013. http://www.R-project.org

[17] D. Damian, A. Hadwin, B. Al-Ani:" Inbstructional design and assement strategies for teaching global software development: a frameowrk", ICSE '06 Proceedings of the 28th international conference on Software engineering , pp 685-690, Shanghai, China, May 20 - 28, 2006

[18] J. Grandin, E. D. Hirleman:" Educating Engineers as Global Citizens: A Call for Action / A Report of the Natinal Summit Meeting on the Globalization of Engineering Education", Online Journal for Global Engineering Education, Vol 4., April 2009