Board Mining: Understanding the Use of Board-Based Collaborative Work Management Tools*

Alfonso Bravo[1][0000-0002-6376-0811], Cristina Cabanillas[1,2][0000-0001-9182-8847],
Joaquín Peña[2][0000-0001-9216-9695], and Manuel Resinas[1,2][0000-0003-1575-406X]

[1] SCORE Lab, Universidad de Sevilla, Spain
[2] I3US Institute, Universidad de Sevilla, Spain
{abllanos,cristinacabanillas,joaquinp,resinas}@us.es

Abstract. Board-Based Collaborative Work Management Tools (BBTs) like Trello and Microsoft Planner are widespread and massively used. Their use includes the management of projects, static information, or processes, which is achieved by assigning and moving cards through lists representing specific states, steps, or other classification criteria. BBTs are a flexible solution as boards, lists and cards can be changed by the user to adapt to new situations, e.g., changes in the processes or projects. However, understanding how a board is being used is challenging because what can be seen at a glance is a static snapshot of its current state. BBTs usually produce logs that capture all the activity that has taken place within the boards. In this paper, we leverage that data for mining BBT logs to understand how boards are used and evolve over time. The contribution is three-fold: (i) we characterize boards according to their components and the behavior detected based on their use during a specific time period; (ii) we detect structural changes in the boards, which may imply board redesigns, and visualize the evolution of the boards' lists; and (iii) we define a set of metrics to assess relevant features of BBT boards, which enables the classification of the boards led by BBT design patterns. To validate the approach, we have conducted an empirical analysis with more than 60 real event logs and a use case.

Keywords: board-based tools · board mining · collaborative work management tools · design patterns

1 Introduction

Board-Based Collaborative Work Management Tools (BBTs henceforth) like Trello, Planner, and Asana are widespread and massively used in informal as well as formal contexts. Organizations adopt these tools to manage their daily

---

* Financiado por los proyectos PID2021-126227NB-C21/ AEI/10.13039/501100011033/ FEDER, UE; TED2021-131023B-C22/ AEI/10.13039/501100011033/ Unión Europea NextGenerationEU/PRTR y US-1381595 (Junta de Andalucía/FEDER, UE

work and processes aiming to increase productivity in busy environments [5]. BBTs are flexible and adaptable to manage different projects and processes and to support changes in them. They were conceived to manage collaborative work but their actual use expands to other scenarios like sharing knowledge, managing shared processes or representing shared schedules [1,3,4,7,8,12]. This is possible because of the different meanings that lists and cards can be given in each board as well as the different ways in which they can be used.

Looking at a board at a given time, we get a static snapshot of its state at that moment. It is not possible to fully understand how the board is being used without additional information or historical use data. Further, boards may evolve to adapt to new needs or requirements. The changes performed may impact not only the structure of the boards but also how they are used. For instance, a board that has been used to classify papers according to where they are going to be published (e.g., workshop, conference, journal) may be converted into a board to classify them according to the publication lifecycle (e.g., writing, reviewing, published). The former is a static board whereas the latter is dynamic as cards move through the lists. This information is also overlooked when disregarding the boards' history and might be helpful when performing audits or to detect why the use of boards has stopped or why their quality has decreased over time.

BBTs usually produce (event) logs that capture all the activity that has taken place within the boards during their design and use. Such log data has been used to analyze how people interact with the boards, usually to identify collaboration patterns or assess participants' performance within educational settings [10,13].

In this paper, we investigate the use and evolution of BBT boards by mining their event logs. The goal is threefold: (i) characterize boards by their components and the behavior detected based on their use during a time period; (ii) detect structural changes in a board, which may imply board redesigns, and visualize the evolution of the lists; and (iii) define a set of metrics to assess relevant features of BBT boards (such as the percentage of card movements), which enables the classification of the boards based on BBT design patterns [9].

To evaluate the feasibility and applicability of the approach, we have implemented it and used it to analyze over 60 real event logs of public Trello boards. This results in a dataset of boards classified according to the defined metrics. Furthermore, we have applied our board mining approach to a use case to show its use to characterize boards and depict their evolution according to their use.

The paper is structured as follows. Section 2 outlines relevant BBT concepts. Section 3 presents our board mining approach to understand the use of BBT boards. Section 4 addresses its evaluation. Section 5 outlines the state of the art on the analysis of BBT boards. Section 6 describes conclusions and next steps.

2 Background on Board-Based Tools

BBTs are built of three main elements: boards, lists and cards. A board contains a set of lists that can be created, updated or closed. Each list contains a set of cards that can also be created, updated, removed or closed. Cards can also be moved from one list to another. These elements represent different concepts of the real world depending on the context of our problem (e.g., lists usually represent steps of a project, phases of a process, or topics, while cards frequently represent tasks, activities, information or resources). In the board analyzed in the evaluation (cf. Section 4), cards are tasks from a software development project, and lists represent the status in which cards are in a certain moment. Many BBTs also provide other elements such as people, labels, due dates, or checklists. Considering these other elements is out of the scope of this paper.

Boards have an implicit design that can be captured by the board design metamodel introduced in [9]. This allows representing both the structure of the board and its dynamic use. The former is composed of the board, its set of lists, and possibly a set of default cards. However, a board design is not only its static structure. Boards have an implicit or explicit set of rules collected in natural language about their semantics and how they should be used. The board design metamodel identifies three elements that characterize how the board must be used: the *type of cards*, the *semantic precedence* and the *card flow*. The first one determines the content of the board, distinguishing between cards that represent tasks and those that represent information or resources. The *semantic precedence* specifies that there is some high-level connection between two lists in the board besides their visual representation. For example, we can have an order relationship between the phases of a project (*requirements* before *execution*) or the states of a task lifecycle (*doing* before *done*). Having an order relationship between two lists does not mean that cards can be moved between them (e.g., lists represent phases of a project and cards represent specific tasks for each phase). Therefore, we need a third element, *card flow*, to specify the set of lists between which cards can be moved. The meaning of a card flowing between lists depends on the specific board. For instance, a board in which lists represent people and cards are tasks, moving a card from one list to another means assigning a task to another person. Instead, a board in which lists represent states and cards are resources, moving a card means changing the state of the resource. Attending to how these three elements are used, we can define eight board design patterns (cf. Table 1) that determine different ways of using a board [9].

Two main phases stand out of the life of a board [9]. The first one is *board design* and it includes all the activities related to the creation of the lists of the board and the initial set of cards. This also involves all the decisions about the static structure, the dynamic use and the meaning of lists and cards.

Once the board is designed, the *board use* phase begins. This involves using the board for the predetermined purpose usually by adding, closing, updating, or moving cards between lists. However, the structure of a board also *evolves* with its use. New lists are created, closed, renamed, or updated. Sometimes these changes are part of the regular use of the board. For instance, a new list is created for each week or each release (e.g., a to-do list for a new release). In other occasions, the *structural update* is the result of a major *redesign* of the board that may completely change the way in which the board represents the problem. Finally, during its use, we can also find *status updates*. They occur when the current snapshot of the board is obsolete with respect to that moment in the real world. This involves all actions on the elements of the board that synchronize the information represented in it with reality (e.g., creating new cards, updating, archiving, deleting or moving existing cards). Unlike structural changes, these updates do not change the board design or structure. In this paper, we do not consider status updates and focus only on the other aspects of board use.

3 Board Mining

Board Mining refers to the extraction of information related to a board based on its event log. To this end, we start with the definition of what a board-based event log is. Then, we discuss about structural updates of a board and how to detect them. After that, we describe how to discover a board-based design model based on the metamodel of [9]. Next, we detail how a board-based event log can be transformed into a process event log. Finally, we report a set of metrics to characterize boards and their relationship with the design patterns [9].

Due to space limitations, the formalization of the definitions and operations presented in this section have been derived to our repository[3].

A **board-based event log** is a set of unique events related to a board that are recorded in a particular order following the temporal sequence in which those events occurred in the board. Conceptually, it is similar to a business process event log but instead of using the concepts of *case* and *activity*, it uses the concepts of *list* and *card*. An event is uniquely identified by its *identifier* and is characterized by the *type of action* it represents and the *lists it affects* (if there are multiple lists involved, e.g., when moving a card from one list to another), among other attributes. An action can be performed upon a card, a list, or the board itself, thus determining the type of action.

As described in Section 2, the structure of a board evolves with its use. We define the structural evolution of a board based on two operations: identify the timeline of every list in the board (i.e., the moment at which the list is created and the moment at which it is closed), and identify the moments in which a significant structural update has been performed. Given the start and end of each list, the **list evolution** is the set of events that occur on each list in that interval of time. Note that for lists that have not been closed, the timestamp of the last event of the board is taken as reference for the last event of the list. Given a board-based event log and a time period, the **structural change events** is the set of events that meet one of two conditions. Firstly, an event can be a structural change event if it is a list creation, update, or deletion event. Alternatively, an event can be a structural change event if it occurs within a certain time period, which is defined as the maximum time difference between any two events that are considered part of the same structural change event. A **structural change interval** is a period where significant changes in the structure of the board have occurred. These changes are detected by specifying

---

[3] `https://github.com/isa-group/board-mining`

a minimum number of list events that must be present in the interval. The potential structural change intervals are defined as a set of time intervals where there exist two events, which mark the start and end of the interval, respectively, and where no events immediately before the first one or immediately after the last one are also considered structural change events within that time period. All events in that period must be structural change events.

Board usage involves creating, updating, moving, deleting and closing cards. Although BBTs give freedom to users to apply any of these actions on any card that is on any list, in practice, users set implicit rules about how the board should be used. For instance, in a board with the following lists: *Inbox*, *Planned*, *Doing*, and *Done*, implicit rules on this board might be that new cards should be created in *Inbox*, and they should be moved to *Planned*, *Doing*, and *Done* in that order (being updated along them if needed). Finally, cards in *Done* should be closed. The goal of discovering a board design model involves using the information from the BBT log to uncover these rules. To this end, we use the metamodel presented in [9] with two key differences. First, we do not distinguish between lists that contain information cards and lists that contain task cards because performing this automatically involve semantic analyses that are out of the scope of this paper. Second, we include additional elements reflecting the interaction of the user with the cards other than card flow. A **board design** comprises the set of lists of the board, the card flow, the semantic precedence, and the set of lists in which cards are created, closed and updated. A threshold can be defined such that the board design must satisfy the condition that the values of the aforementioned parameters have a frequency higher than the given threshold. Using a BBT log it is possible to **discover a board design**: the lists are the set of lists that appear in the events of the log, the card flow is the set of lists connected to each other by means of card moves, and the set of lists created, closed and updated are composed of those lists where cards are created, updated or closed with a frequency higher than a threshold.

Note that we are representing a board design (i.e., its structure and the implicit rules that determine how it should be used), not an actual board use. That is why cards are not part of a board design. Furthermore, this definition does not provide information on how the structure of the board can change. Also, semantic precedence has not been included in the definition of process discovery because it requires an additional consideration. A semantic precedence between two lists involves a high-level relationship between them. However, the way this high-level relationship manifests can change depending on the semantics of the board [9]. For instance, in a board where lists represent a lifecycle and cards move through these lists, the semantic precedence manifests by card movement. On the contrary, in another board, lists may represent the months of the year and cards may not move through these lists. In this case, the semantic precedence between lists (i.e., the temporal precedence) manifests by periods of card creations or updates that occur in a different list every month. In this paper, we focus on the semantic precedence that manifests by card movement. Specifically, we consider that there is a semantic precedence between two lists if the number of cards that move from one to another is higher than a threshold.

A consequence of the discussion about semantic precedence is that depending on the board and the semantics associated to its lists and cards, it is possible to identify a process event log derived from the board-based event log. A **process event log** is a set of process events that are temporally ordered based on their timestamps, with the same conditions as in a board-based event log. We focus on boards that follow the *information lifecycle* or *kanban* patterns identified in [9]. In them, cards may represent tasks or resources; they are moved around lists, which represent activities or stages; and there is a semantic precedence between the lists based on card movement. The **mapping to a process event log** is thus rather straightforward since cards can be seen as cases, and lists can be seen as activities. Two considerations about this mapping. First, if we had considered additional data attributes in the board-based event log, their mapping to a process event log would be a straightforward 1:1 mapping. Second, we are assuming that the process events correspond to completed activities. For this reason, each activity will be added in the process log when the card is moved out of the respective list or it is closed. We also add an initial activity called *created* to represent the creation event.

Based on the described operations, we define a set of metrics that can be used to characterize the use of a board based on its card flow. Given the lists and cards of a process event log, **moving cards** are defined as those that change lists at least once in the log. **Closed cards** are those that are closed in said interval. **Relative moving cards** refers to the percentage of cards that move in relation to the total number of cards present in the log. **Relative closed cards** is analogous but with the closed cards. **Relative lists with card movement** relates to the number of lists that contain or have contained moving cards in a period. Finally, **relative moves per moving card** relates to the number of movements made by each moving card.

4 Evaluation

We have validated our approach with an empirical analysis and a use case. Information required for replication, datasets and algorithms are available online[3].

We have performed an empirical analysis with real Trello logs from 616 public logs obtained by scrapping them from `www.trello/b/*`. We first filtered to keep the logs of the boards that (i) represent their entire life (i.e., the log starts with a board creation event); and (ii) have over 2,000 events and over 12 weeks of use, to ensure an extensive/real use. This results in 63 logs to be analyzed.

Structural changes (cf. Definition in Sec.3) were detected using a time period of 24 hours and a minimum number of 4 list events. All the events in between structural changes were extracted from the log as a sub-log obtaining 311 sublogs, each of them corresponding to a stable board structure.

Pattern	**Card type/Sem. prec./Card flow**	**Rel. # moving cards**	**Rel. # lists with card moves**	**Rel. # moves per moving card**	**Rel. # closed cards**	**% in [9]**	**% in logs**	**Diff.**
Info. lifecycle	Info/Yes/Yes	*0.52*	*0.83*	*0.21*	0.12	0.27	0.25	0.01
Ordered Info.	Info/Yes/No	*NA*	*NA*	*NA*	*NA*	0.13	0.02	0.11
Kanban	Tasks/Yes/Yes	*0.60*	*0.88*	*0.18*	0.21	0.22	0.27	0.05
Tasks Process	Tasks/Yes/No	NA	NA	NA	NA	0.06	0.02	0.04
Assigned Info.	Info/No/Yes	*0.29*	*0.78*	*0.12*	*0.18*	0.33	0.05	0.03
Categ. Info.	Info/No/No	*0.03*	*0.27*	*0.03*	*0.07*	0.24	0.35	0.10
Assigned Tasks	Tasks/No/Yes	*0.39*	*0.87*	*0.21*	*0.32*	0.01	0.04	0.02
Categ. Tasks	Tasks/No/No	NA	NA	NA	NA	0.05	0	0.05

Table 1. BBT design patterns and values obtained for the metrics. *NA*: not applicable.

4.1 Empirical analysis

We analyzed the latest snapshot of the board and manually identified the design pattern followed. From the 63 last-interval sub-logs, 55 follow a specific pattern and 8 present a hybrid design using several patterns, which cannot be identified by our approach. We calculated the average values for the board use metrics (Def. 3), as shown in Table 1. We did not find enough logs for patterns *Categorized Tasks* (0 logs), *Task Process* (1) and *Info. Process* (1), as expected since this matches the distribution of patterns found in [9].

To characterize and understand the real use of the boards, Table 1 shows the value of card metrics as in the logs (average values for each log labelled with each pattern). For most patterns, the real values (in italics) are far from the theory:

- **Relative # moving cards:** The average for *Info. Lifecycle* is 0.52, and for *Kanban* is 0.60. Half of the cards not moving may indicate they do not change or are not updated, since their purpose is to monitor the state change of cards. For rest of patterns, empirical values match their theoretical values.
- **Relative # lists with card movement:** Patterns that involve list movements (*Kanban*, *Info. Lifecycle*, and *Assigned info./task*) have values over 0.8, being the expected theoretical value 1. This may represent users' lack of discipline, design errors, or documentation lists without card movement. For *Categorized Info.*, 0.27 lists with card movement may indicate initial classification until getting the final static classification.
- **Relative # moves per moving card** For *Info. Lifecycle* and *Kanban*, this metric should tend to 1, since every card should undergo the complete lifecycle. For *Assigned Info/Task*, this number can vary but should be greater than 0. However, in both cases this metric is around 0.21, far from the theoretical value. This again can be caused by users' lack of discipline. For *Categorized* and *Process Task/Info*, it should be 0, which matches the practice.
- **Relative # closed cards:** A card is closed when we finish the task or we stop handling the information it represents. Although the number of closed cards is higher in task patterns (0.21 and 0.32) than in information patterns (on average 0.11), in reality people tend to leave cards open, possibly due to low finished tasks or to using lists to represent this final state (e.g., done) with the same meaning as card closure but leaving cards open. For *Categorized Info*, the value is 0.03, as expected to represent information statically.

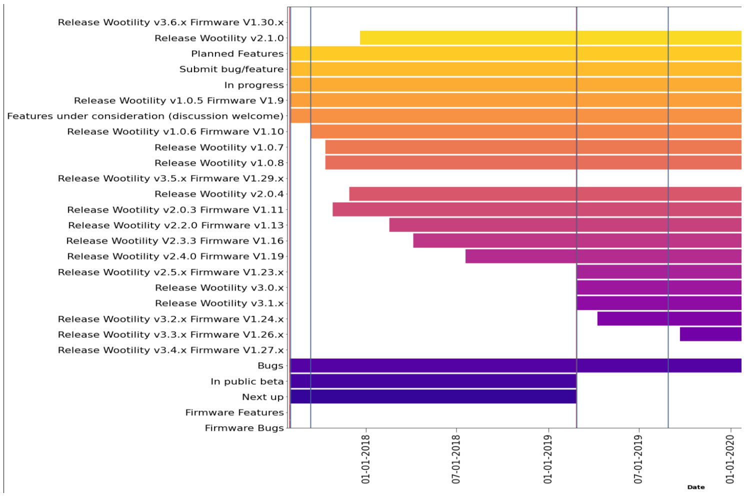

Fig. 1. Evolution of the lists of Board A over time (x-axis). Each row represents a list. Vertical lines represent structural updates (in red its start and in blue its end).

4.2 Use case

The board we have selected[4] represents a software development process where users collect the features on the cards classifying them by their level of completion and indicating their status on the lists. It has been used for more than four years, with more than 1,000 actions over the elements of the board.

Fig. 1 depicts list evolution and structural changes showing how users are continuously creating new lists: one per each new release. Many of them are not depicted with a vertical line in the figure because the threshold established requires at least four list events within a structural change interval to signal it. This is a structural change that represents a regular use of the board, and not a board redesign, since it does not imply a change in the way the board is used.

However, the figure also depicts a structural change around February 2019 that involved a board redesign. In particular, lists *In public beta* and *Next up* were removed meaning that they simplified their workflow (or at least the parts of their workflow managed by the BBT). Checking the percentage of cards closed before and after that date, it changes from around 2% to up to a 20%, meaning that they also decided to act differently with the cards in that regard.

After understanding the evolution of this board, we now turn our attention to its design. Particularly, we focus on the board design from its creation until the redesign that happened in February 2019. Fig. 2 shows the board design discovered represented using the notation of [9]. The thresholds used for semantic precedence is 0.025, and for card creation and update is 0.10. We do not depict card closure because less than 2% of the cards of the board are closed. The diagram shows that cards flow between all the lists of the board except for

---

[4] `https://trello.com/b/NpKEdAgB/wooting-roadmap`

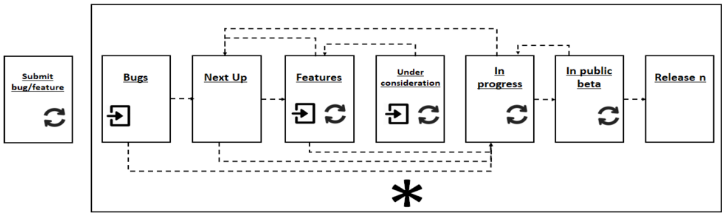

Fig. 2. Design of Board A using the notation of [9]. Each box with a name is a list. The outer box represents the card flow between lists. Arrows between lists are the semantic precedence. The symbol in some lists means that cards are created in them.

Submit bug/feature, whose cards are not moved from or to any other list. This is because users submit bugs or features not as new cards, but as comments in one predefined card of this list, having a high level of card updates in it. If we turn our attention now to the other lists, we can see how the creation of cards takes place in lists *Bugs*, *Features* or *Under consideration*, as appropriate. Then, some of them pass to *Next up* when they are scheduled to be included in the next release, while others directly go to *In progress* list, when the corresponding feature/bug is being developed/fixed. After this phase, most of the cards go to *In public beta*, but some other cards have to go back to *Next-up* (those cards which do not have been finished for the corresponding release). Finally, cards move to their corresponding *Release* list, once they are developed. This behaviour fits with a *Kanban* or *Info. Lifecycle* pattern.

5 Related Work

BBT logs have been used to analyze aspects related to people involvement and participation in the context of controlled experiments with students. Mora et al. [6] used gamification to investigate personalized learning experiences' impact on student motivation and engagement. They divided the students in groups, created Trello boards and used the logs to perform descriptive analyses on the actual involvement of students in the groups based on their activity. An approach to predict low performance in teams of students is described by Shamshurin and Saltz [13]. They analyzed 80 Kanban projects corresponding to teams' Trello boards on the basis of predefined Kanban metrics [11] as well as on several ad-hoc metrics based on counts. Afterwards, they used machine learning models to predict risks of low quality results. In [2], Buffardi presents an exploratory analysis of 10 software engineering teams of students based on evidence in teams' Git logs and user stories (Kanban boards). The goal was to derive individual assessments from collaborative work results. Here, BBT log data was not used and the analysis was done manually by checking the number of cards in a specific list at the end of the experiment. Pisoni et al. [10] identified patterns of collaborative behavior from Trello data obtained from 16 teams. Their approach starts with a preprocessing to detect sequences of actions that could be separately analysed, followed by the identification of Trello actions, their coding in 5 categories, a data analysis and a final clustering to identify similar group activity patterns.

To the best of our knowledge, our work is the first attempt to analyze how BBTs are used in a domain-independent way focusing on understanding how boards evolve over time rather than on analyzing them from a people standpoint.

6 Conclusions and Future Work

In this paper, we have investigated BBTs usage to address real problems by analyzing their logs to discover the evolution of lists, structural changes, and board design. We have also defined a set of board use metrics to analyze how they are actually used and identify their design pattern. The results of our empirical analysis of 62 real event logs show a lack of discipline using boards since real values of the metrics lie on average under 50%, showing a deficient use of the boards. In addition, mechanisms to ensure, monitor, and recommend the right use of boards are missing. From the results of our use case we can conclude that the application of board mining allows us to get insights into the real use of BBT boards and their evolution over time. As next steps, we plan to develop tool support to enable the use of our approach and extend the analyses to include other sources of BBT logs.

References

1. Ault, A., Krogmeier, J., Buckmaster, D.: Mobile, Cloud-Based Farm Management: A Case Study with Trello on My Farm. In: ASABE (2013)
2. Buffardi, K.: Assessing Individual Contributions to Software Engineering Projects with Git Logs and User Stories. In: ACM SIGCSE. pp. 650–656 (2020)
3. Fic, P.: Moved to published: Using trello in content management. Dianoia (2019)
4. Gould, E.M.: Workflow Management Tools for Electronic Resources Management. Serials Review **44**(1), 71–74 (2018)
5. Khoury, A., Bucknor, A., King, I., Kerstein, R., Nduka, C.: Use of Trello as a Project Management Tool for Collaborative Surgical Research and Audit. British Journal of Surgery (2022)
6. Mora, A., Tondello, G.F., Nacke, L.E., Arnedo-Moreno, J.: Effect of personalized gameful design on student engagement. In: IEEE EDUCON. pp. 1925–1933 (2018)
7. Naik, N., Jenkins, P.: A Web Based Method for Managing PRINCE2 Projects Using Trello. In: IEEE ISSE. pp. 1–3 (2019)
8. Ostergaard, K.: Applying Kanban principles to electronic resource acquisitions with Trello. Journal of Electronic Resources Librarianship **28**(1), 48–52 (2016)
9. Peña, J., Bravo, A., del Río-Ortega, A., Resinas, M., Ruiz-Cortés, A.: Design Patterns for Board-Based Collaborative Work Management Tools. In: CAiSE. pp. 177–192 (2021)
10. Pisoni, G., Gijlers, H., Chen, H., Nguyen, T.H.: Collaboration Patterns in Students' Teams Working on Business Cases. In: L2D Workshops. vol. 2876, pp. 14–27 (2021)
11. Power, K.: Metrics for Understanding Flow. In: Agile Conference (2014)
12. Ray, N.: Prioritize, Plan, and Maintain Motivation with Trello. AEM (2016)
13. Shamshurin, I., Saltz, J.: A predictive model to identify Kanban teams at risk. Model Assisted Statistics and Applications **14**, 321–335 (2019)