Blackbox, Five Years On: An Evaluation of a Large-scale Programming Data Collection Project

Neil C. C. Brown
King's College London
London, UK
neil.c.c.brown@kcl.ac.uk

Amjad Altadmri
King's College London
London, UK
amjad.altadmri@kcl.ac.uk

Sue Sentance
King's College London
London, UK
sue.sentance@kcl.ac.uk

Michael Kölling
King's College London
London, UK
michael.kolling@kcl.ac.uk

ABSTRACT

The Blackbox project has been collecting programming activity data from users of BlueJ (a novice-targeted Java development environment) for nearly five years. The resulting dataset of more than two terabytes of data has been made available to interested researchers from the outset. In this paper, we assess the impact of the Blackbox project: we perform a mapping study to assess eighteen publications which have made use of the Blackbox data, and we report on the advantages and difficulties experienced by researchers working with this data, collected via a survey. We find that Blackbox has enabled pieces of research which otherwise would not have been possible, but there remain technical challenges in the analysis. Some of these – but not all – relate to the scale of the data. We provide suggestions for the future use of Blackbox, and reflections on the role of such data collection projects in programming research.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; • **General and reference** → *Empirical studies*; Evaluation;

KEYWORDS

Blackbox; Shared Data; Mapping Study

ACM Reference Format:
Neil C. C. Brown, Amjad Altadmri, Sue Sentance, and Michael Kölling. 2018. Blackbox, Five Years On: An Evaluation of a Large-scale Programming Data Collection Project. In *ICER '18: 2018 International Computing Education Research Conference, August 13–15, 2018, Espoo, Finland.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3230977.3230991

1 INTRODUCTION

Researching programming education can be difficult. A researcher may have several interesting research questions, but finding or generating sufficient data can be a challenge. Many studies are published based on limited data (quantitative or qualitative) gathered from a single cohort, often with only a few dozen participants. Statistical analyses of such a dataset, as well as generalised conclusions, are necessarily of limited reliability and value – both because of the small sample size, and because the data is from a single-institution, often a single teacher (who may be the researcher themselves).

It seems advantageous to have a central large shared data pool which researchers could access. This idea led to the creation of the Blackbox project. Blackbox [9, 34] is a data collection project, designed and implemented by the creators of the BlueJ IDE, an educational Java development environment. Blackbox collects activity data from BlueJ users, including source code, edit sequences, testing and execution interactions, and compilation results.

Participation in the Blackbox data collection is voluntary, via an explicit opt-in choice of each user. Approximately 40% of BlueJ users choose to participate. Since BlueJ has several million users (who are typically novice programmers) per year, Blackbox has a large repository of novice programmer interaction data, which can form the basis of various research studies. Blackbox currently contains records of over 30 million programming sessions, including 300 million compilation events.

Blackbox was intended, from its conception, as a shared data repository that would be made available to various research groups for the investigation of many different research questions. It was hoped that the project could enable, maybe even stimulate, research that would otherwise be difficult or impossible to conduct.

This goal implicitly poses opportunities and difficulties. Making the data available to other interested researchers can increase the value of the data collection, thus better justifying asking users to participate and the effort in collecting and storing it. However, collecting observational data without a specific experiment carries multiple risks. The activity data is devoid of demographic and contextual data: we do not know who is programming, any details about them, or what their aim is. Additionally, the trade-offs of granularity and complexity of data collection and storage formats had to be designed based only on tentative predictions of future research questions and researchers' needs. It was possible that the data collected would not be useful to anyone.

Thus at the outset of the project there were several open questions about the project's success. Would it be possible to collect a meaningful amount of data? Would a single dataset in a fixed data format be useful to multiple different and diverse researchers and studies? Would other researchers be interested in working with an observational dataset when they had no opportunity to influence the details of the collected data or apply an intervention? Could a data access format be provided that makes it sufficiently easy, at the same time as sufficiently flexible, to be accessed with available technical expertise in research groups? Blackbox has now collected data for almost five years, and the repository has been available for researcher access for the whole duration. This is sufficient time to now conduct an investigation into the answers to these questions.

ICER '18, August 13–15, 2018, Espoo, Finland

ACM ISBN 978-1-4503-5628-2/18/08…$15.00
https://doi.org/10.1145/3230977.3230991

In this paper, we present a retrospective of the Blackbox project to date. Our contributions include:

- Quantitative results of the data collection. This includes the counts and size of the collected data, as well as some logistic details of operating a large-scale collection project, including server load and storage (Section 2).
- A mapping study of the literature published using Blackbox data. This helps us examine the use of the data by multiple research groups and provides a lens through which to assess the project's impact (Section 3).
- A survey of the researchers who used Blackbox or similar datasets, to examine the successes and difficulties of using such data for programming education research (Section 4).

The results presented here illustrate some of the possible scope and limitations of using Blackbox, and will be of use to other researchers contemplating working with this data in future. This paper also provides a useful examination of the more general role of large-scale datasets within computing education.

1.1 Related Work

The most closely related pieces of work are the studies based on the Blackbox data, which are covered in the mapping study in section 3. There exist several other similar datasets of novice programming data, such as code.org's Hour of Code dataset [5] and CloudCoder [32]. Ihantola et al. [16] and Hundhausen et al. [15] present useful overviews of these kinds of dataset. We are, however, not aware of a detailed retrospective evaluation of the usefulness and impact of a single dataset as presented in this paper.

2 BLACKBOX DETAILS

The first time a new user starts BlueJ, a novice Java IDE often used in programming instruction at school or introductory university level, they are asked whether they want to opt-in to the Blackbox data collection. Out of approximately 2.2 million unique users per year, roughly 40% do so. If a user has opted in, the Blackbox infrastructure collects activity details, including:

- The full source code of the user's project. For anonymisation purposes, the header comment of each class is removed.
- Edit actions at the source line level. Each time the cursor leaves a line, or the user compiles, the edit step is recorded.
- Compilation events, including success or failure, and any associated error information (source position and message).
- Use of BlueJ's testing tools, interactive method invocations, and various other IDE features.

Data collection began in June 2013. As of April 2018, the Blackbox database consumes 2.31 terabytes of disk space. It contains records of 32.1 million sessions from 2.58 million users, and 306 million compilation events. The server has received 2.36 billion separate activity items. (All items given to 3 s.f.)

2.1 Practical details

The data is collected into a MySQL database on a single machine. Typical processor load on the collection server (a machine purchased in 2013, see Brown et al. [9] for specifications) is estimated at less than one core, and up to 2–4 cores during peak times.

The database is live-mirrored to a second machine which is used for analysis purposes. The decision to separate the recording machine from the analysis machine has worked well. The analysis machine is used sparsely, but when it is used, is often used to maximum capacity. Separating data collection and analysis across two machines provides a guaranteed prevention of the analysis interfering with the data collection.

Data collection has been almost continuous since 2013, with only a small number of interruptions. Only three recording outages lasted longer than a few minutes:

(1) The initial data collection was not multi-threaded, and some data was lost due to server requests timing out during busy periods before this was fixed.
(2) The domain registration was accidentally allowed to lapse, and for a day or two some data was not recorded.
(3) The only major change to the schema to date was performed in August 2017, involving recording down-time of 6 days. Unlike the others, this outage was pre-planned and Blackbox researchers were notified in advance.

2.2 Original data estimates

The original Blackbox proposal [34] included an upper limit estimate of the expected data volume, based on user numbers at the time and the maximum 100% opt-in rate, which was used to decide the required specification of the server hardware. It stated:

> "At current usage levels… this will lead to a maximum of around 27,000 users per day, performing on average 3 sessions per day, and generating about 100 events per session over an average period of 90 days. This would mean overall a maximum case of 8 million events per day, or just under 100 events per second, and a total of 3 terabytes of data per year."

We can now compare these initial estimates (adjusted to the actual opt-in rate of 40%) to actual observed data volumes. Blackbox presently sees 5,000 to 20,000 users per day, performing an average of 2-2.5 sessions per day, with around 80 events per session. The average so far in 2018 is 2 million events per day, and the total data size is now 2.3 terabytes. This means that most of the estimates were quite accurate, even though they were made before any data was recorded. The main discrepancy is that the overall database size is smaller than estimated, probably due to the compression afforded by storing only diffs for edits.

2.3 Project administration

To gain access to the data, a lead-researcher has to be identified who is a permanent member of staff of an established research institution. This lead-researcher can then request access for other researchers, who may be students. All access requests require the lead-researcher to provide a short description of the research aims, and to sign an ethics declaration and code of conduct, which includes assurances to maintain the confidentiality of the data. Researchers are given a copy of the 60 page Blackbox handbook.

The Blackbox analysis machine has accounts for around 130 users. At a rough estimate of an average of three users per research group, this means a little over 40 research groups have likely signed up for access.

As Blackbox administrators, we offer the following observation: our contact with users interested in the dataset has typically consisted of talking to academics about the initial data access, followed by correspondence with students about intricacies of the actual data access and analysis. In several cases, the queries we received from students indicated a weak understanding of SQL database queries and incomplete programming knowledge.

3 BLACKBOX USAGE: A MAPPING STUDY

Blackbox has been running for almost five years. Although research publication has an inherent lead-time, this seems long enough to investigate the studies which have resulted from the Blackbox data so far. Therefore we conducted a small mapping study. A mapping study, such as the one performed by Kaijanaho [19], examines the research that has been carried out in an area, with a focus on which topics have been investigated. Unlike a systematic review, a mapping study does not try to synthesise the results of the research. In planning the study, we followed the guidelines by Petersen et al. [27] for conducting mapping studies in software engineering.

3.1 Design

3.1.1 Scoping. In this study, we were interested in any papers that had directly analysed Blackbox data (primary studies), or had used any results of primary studies in a detailed further analysis (secondary studies). This narrow focus meant that the studies (which shared the same data source) could be summarised coherently – we deliberately chose not to consider studies performed on similar datasets to Blackbox, which would have been a much wider analysis.

3.1.2 Search Strategy. We conducted our search with the assumption that all published Blackbox studies will cite at least one of the original Blackbox papers. Thus we conducted a snowball search (as per Wohlin [36]), beginning with two early publications on Blackbox [9, 34], and one of the better-known results papers by the administrators of the Blackbox dataset [3]. We first looked for all citing papers of these initial three in both the ACM digital library and Google Scholar (de-duplicating by hand). Then, any papers found to be Blackbox primary or secondary studies had their citations examined, repeating until we found no more Blackbox primary or secondary studies in the references. As a sanity check, before conducting the snowball search we wrote down any Blackbox papers we knew *a priori*, and we made sure that they were in the returned results.

3.1.3 Classification. We did not decide classifications ahead of time. Instead, each relevant paper was examined by two of the authors, who independently constructed categorisations for the full set of papers. These keywords were then compared and merged into a single set of categorisations through discussion, and papers were re-categorised using this single categorisation. A narrative summary was then produced.

3.1.4 Pre-registration. The protocol described above was pre-registered on 21st February 2018 at https://osf.io/y2amu/ before conducting the search.

3.2 Search Outcome

The search was carried out on the 21st to 23rd of February 2018. The pre-registered protocol was followed with minor adjustments for two unanticipated situations. Google Translate was used to examine non-English papers (a situation not anticipated in the original protocol). We also discovered one paper via Google Scholar which at the time was an accepted pre-print; we included this as a search result. In total, 304 citation links were assessed, which resulted in 135 unique papers being examined. Including 2 of our original 3 seed papers[1], 16 publications were found that that were primary Blackbox studies [2–4, 7–9, 11, 18, 20, 22, 24–26, 28–30]. No secondary studies were found.

All papers that were known to us before beginning the search were found in the search, except McCall and Kölling [23]. This paper is indexed by Google Scholar and did cite Brown et al. [9] but it seemed that a technical issue with IEEE's citation extraction meant that this citation was not automatically processed (and the paper did not cite any other Blackbox papers, so was not found otherwise by our snowball search). We manually added this paper to the results.

In hindsight, our search protocol was over-elaborate: every primary Blackbox study would have been found by simply looking through the Google Scholar citations for Brown et al. [9] (except for that paper itself).

We made one further unplanned alteration to the search. The original search took place before the papers from SIGCSE 2018 were indexed on the ACM digital library and Google Scholar. Conscious that there may be relevant papers published at SIGCSE 2018, we re-checked the direct citations of the Brown et al. [9] paper on the ACM digital library and Google Scholar on 5th March 2018, which yielded one further publication by Becker et al. [6], for a final total of 18 publications (16 original, 1 manual correction, 1 extra) [2–4, 6–9, 11, 18, 20, 22–26, 28–30].

3.3 Analysis Outcome

Two of the authors acted as coders. They independently created a tagging scheme for the full set of 18 publications, and then jointly agreed a single tagging scheme. They then independently tagged all 18 publications using the new scheme. This resulted in 89% agreement (of all tag-paper pairings), 5% where one or both researchers felt more clarification was needed on tag definitions, and 6% disagreement. A final discussion achieved 100% agreement.

Most papers described work which had been carried out, while some papers (in particular, Kurtiker and Wagh [22], Mirza et al. [24, 25]) described work which was planned. We tagged all work by topic regardless of whether it had or had not been carried out (in the spirit of a mapping study – which looks at topics, rather than results). In our results section, however, we note the planned versus carried-out distinction.

3.4 Results

Of the 18 papers examined, three were published in 2014, four in 2015, three in 2016, five in 2017, and three in 2018 (up to early

[1] Utting et al. [34] was a seed paper for the search but featured no Blackbox analysis, therefore is not eligible for consideration as a result of the search.

March). Seven of the papers had one or more of the Blackbox administrators as authors[2], and the remaining papers could be separated into nine disjoint author clusters (thus ten in all).

3.4.1 Topics. The most popular topic was examining errors in code, with 13 of the 18 papers [2–4, 6–9, 18, 22, 23, 26, 28, 29] investigating some aspect of programming errors. Five of these papers relied solely on using the content of the Java compiler's error messages (which are known to change between versions, making the analysis fragile), while eight [2–4, 7, 8, 23, 26, 29] made use of a custom error classification system that was partially or totally independent of the compiler. Six papers split the errors into higher level categories (e.g. syntax vs semantic), ten investigated error frequencies, six analysed time-to-fix. Two papers looked at the content of error messages, and two investigated suggesting possible fixes for errors. Four papers examined code style issues and two were concerned with plagiarism detection.

In total, thirteen papers performed a manual or automatic analysis of Java source code, while the others were based on non-source data, such as numbers of compile errors, numbers of edits, etc. Only Santos et al. [29] made any use of machine learning techniques to analyse the data.

Not all data collected in Blackbox saw much use. Only de Souza et al. [11] made use of the JUnit test-related data subset. Kurtiker and Wagh [22] planned to make use of local participant tagging to add demographic data, but to date only Ahadi et al. [2] have actually done so. Jadud and Dorn [18] used the location to analyse country differences, and two other papers [6, 26] have made use of country locations to constrain analysis sets. Several other parts of the data (e.g. dynamic invocations, exceptions) have not yet been used in any published work.

3.4.2 Research Methodologies. Ten papers applied a pre-existing theory to analysing the dataset. Three of them used the data to construct a model which could be used in future work (and a further paper planned to do so). Many of the papers performed exploratory data analysis without a particular theory, and/or reported results without using them to construct an explicit model.

Two papers described replications of previous work[3]: Jadud and Dorn [18] used the dataset to replicate earlier work on Jadud's error quotient, and Ahadi et al. [2] replicated earlier Blackbox work by Brown and Altadmri [8] in a local context.

4 BLACKBOX USAGE: RESEARCHER SURVEY

The published papers on Blackbox reveal some interesting information about topics of interest, but they cannot capture two aspects in particular. One is the detailed experience of researchers in using the data (was it easy or hard, what were the challenges, etc.) and the other is the possible experience of researchers who were perhaps interested, but did not use the data for a completed published study. To try to study these two cases, we conducted an online survey of programming researchers.

[2]To clarify: authorship is not a condition of dataset use, so this means the administrators were actively acting as researchers in this work.

[3]Several papers presented tables of compiler error frequencies and compared them to previous such results in other work, but we did not class this alone as a replication.

4.1 Design

4.1.1 Motivation. We had three informal primary hypotheses which we wanted to investigate:

- Sharing the Blackbox data is useful to researchers, due to the large data size and the convenience of not needing to collect your own data.
- Complex SQL databases are too difficult for many computing education researchers and their students to work with effectively.
- Many researchers have interesting questions, but frequently computing education researchers lack the analysis techniques to be able to map high-level research goals to actual analysis strategies of source code.

We were also generally interested in opinions and experiences surrounding the use of Blackbox (or similar datasets) even if they did not relate directly to these three themes.

4.1.2 Survey Design. The online survey had two main branches. A key question early in the survey was:

> What is your relationship to Blackbox?
> (1) I have signed up for access and have used the data for research.
> (2) I have signed up for access, but have not really used it.
> (3) I have heard of it before now, but I have not signed up for access.
> (4) I have not heard of it before now.

If the respondent answered with one of the top two options (a "Blackbox user"), they were asked a set of questions about their actual or planned use of Blackbox, in order to capture the experience of users who used or planned to use the data. If one of the bottom two answers were chosen (a "Blackbox non-user"), they were asked questions about their use of, and opinions on, similar datasets, in order to capture the experience with other datasets for potential comparison. These open-ended text questions were primarily designed as prompts to help explore one of our three themes. The study followed appropriate ethical procedures and was approved by the King's College London ethics committee.

4.1.3 Pre-registration/Materials. The study was pre-registered on 14th February 2018, prior to the beginning of data collection, at https://osf.io/z48v7/ which includes the full survey in the Files section.

4.1.4 Outcome. Data collection was carried out from 14th February to 2nd March 2018 inclusive. The survey was advertised on the Blackroom (a Blackbox users' forum), the mailing list for a relevant Dagstuhl seminar, the csed-research mailing list, on Twitter by several of the authors, and by directly emailing all listed authors on the original 17 papers we found (see section 3.2) who did not work for King's College London (the authors' own institution). We received 21 complete responses to the survey: 13 responses were from Blackbox users, and 8 were from non-users.

4.1.5 Threats to validity. One threat to the survey's validity is that we may not have a representative sample of the Blackbox users and non-users. Researchers may have been more likely to return the survey if they used the data, and we may not have captured non-users who could have used it but did not. However, it is difficult to see how this threat to validity can be avoided.

Another threat to validity is that the survey was advertised and analysed by the authors, several of whom act as Blackbox administrators. Although the survey was anonymous, the answers about the respondent's own research potentially allowed identification. Participants were specifically assured that their responses would be treated professionally and that any decision to respond (or not) or response content would not influence any future treatment by the Blackbox administrators. However, it is possible that some participants may have altered their content of their responses to be less negative about the project.

4.1.6 Analysis. The responses were analysed by two of the authors performing an iterative process of open coding for thematic analysis: first, they made a pass through the whole dataset and formed an individual set of tags for the responses. Then they agreed on a canonical set of tags, and made another pass to tag the data using this set of tags. Inter-rater reliability was assessed using Cohen's Kappa. This was found to be low (less than 0.75) so an additional pass was made after discussing and clarifying the definition of the tags. This second pass resulted in a median Kappa of 0.735, and another pass was not conducted – the union of the two researchers' tagged items was used for each tag, and since these results were primarily used as a basis for a higher level manual summary, we considered this level of agreement to be sufficient.

4.2 Results

4.2.1 Demographics. Participants were asked about their research area (free text response): 12 mentioned computing education, 7 mentioned software engineering. They were asked for their role: 11 were permanent academic staff, 8 were postgraduate students, none were postdocs. They were asked for their relation to Blackbox: 9 were users who had signed-up and used the data, 4 said they had signed up but not really used the data, 6 had heard of the project but not signed up, and 2 had not heard of it before.

We asked each respondent for the role of the person responsible for directly analysing the data. Of the 13 responses from those who had signed up, 3 mentioned academic staff, 4 mentioned undergraduate students, 7 mentioned postgraduate students and 1 mentioned research assistants. The numbers do not add up as some responses mentioned multiple groups, and there could be double-counting if both a PhD student and their supervisor filled in the form (which we cannot know as the survey was anonymous). However, this does indicate that the majority of analysis was performed by students.

4.2.2 Ranking Exercises. The results of all ranking exercises given here use the mean rank, specified to one decimal place.

Blackbox users were asked to perform two ranking exercises. The first involved ranking Blackbox's features in importance to the respondent's work (rank 1 being most important, 7 least important):

Feature	Mean rank
Large size of data set	2.8
Access to source code	2.8
Ability to see edits over time for each user	2.9
Compiler error data	3.8
Other IDE usage data	4.9
Ability to geographically partition users	5.0
Ability to tag users for local experiments	5.8

The next question asked for rankings of Blackbox's features in terms of how problematic they were (rank 1 being most problematic, rank 7 least problematic):

Feature	Mean rank
Lack of information on what tasks users are accomplishing (lack of specific assignments)	2.5
The need to write your own software to pull information from the database	3.3
The need to write your own analysis of Java source code	3.7
Short-lived duration of many users in the dataset	4.1
Lack of demographic information on individual users	4.2
Data is Java-only; no other programming languages	4.8
Data is BlueJ-only; no other Java IDEs/editors	5.5

Blackbox non-users were asked to perform one ranking exercise: to rank Blackbox's features by what would have made the data more suitable (rank 1 being most important, rank 8 least important):

Feature	Mean rank
Adding more programming languages besides Java	3.0
Information on the user's current assignment/task	3.1
Adding demographic information	3.9
Adding more editors/IDEs besides BlueJ	4.0
More long-term tracking of users	4.4
Better support for tools to analyse source code data	5.7
More advertising (I was unaware of Blackbox)	5.9
Better support for tools to access/filter the data	6.0

4.2.3 Text Responses. One theme of interest was how important the Blackbox data was to each respondent's research. Five respondents said that Blackbox was essential to their research: "The research was only made possible by having Blackbox available", "I was only going to go after this question because of the existence of Blackbox", "Without Blackbox much of this work would not be possible. I would have had to alter everything." Five other respondents mentioned that the scale of the data was a major advantage: "[The] worldwide availability of such facility Blackbox data collection project made our research more fascinating and interesting", "Some things we concluded from mining the Blackbox data we would hesitate to conclude using [our own data]."

The Blackbox dataset is an SQL database with many different tables. Eight respondents mentioned having difficulty understanding the structure of the database: "There's such rich data that understanding the table structure can take a while", "We spent a long time trying to figure out to get what we wanted", "it was difficult to link so many tables". Five respondents explicitly stated that more examples or tutorials would be useful.

Four respondents expressed a desire to be able to export the data to a single CSV file. Four respondents also stated that they would like to be able to download the data locally to their own machine for analysis.

Three respondents independently stated that all they wanted was a temporal succession of source code snapshots with compiler errors, without the rest of the data that Blackbox provides, suggesting that this is a particularly common use-case. One respondent had a particular use case (omitted here to guarantee respondent anonymity) which Blackbox satisfied, stating "if anything like that was overlooked it would be problematic".

The most common desire for additional data in Blackbox was for information about the task the student was working on and/or their progress towards a correct answer, with eight respondents making mention of this: "[I would like] information on the task that the student is trying to achieve, including information on the correctness of the solution".

Respondents reported a wide variety of topics. We choose not to report specific individual items here (as, especially combined with the mapping study, this could identify who responded), but common higher-level themes were analysis of students' behaviour either with respect to success (four respondents) or to details like emotional state (two respondents). Two respondents mentioned plagiarism detection. Two others used the data as a code repository for testing error detection or correction tools. Programming errors were a very common item, mentioned by nine respondents.

Three respondents mentioned doing some analysis by hand. Where they mentioned programming languages that they used or would like to use for analysis, the main languages mentioned were Python (four respondents) and R (three respondents).

5 DISCUSSION

For further discussion, we now synthesise the results of the mapping study and the user survey.

5.1 Impact

Our mapping study found eighteen publications that used the Blackbox data. This figure is perhaps unrepresentative of the impact of Blackbox, as several of the publications featured Blackbox administrators as authors. However, excluding the Blackbox administrators, a total of nine disjoint sets of researchers used the data, which we believe is sufficient to show that the dataset has had a reasonably wide impact: we are not aware of another computing education dataset with such wide usage.

5.2 Topics and outputs

Our survey and mapping study confirmed that the study of programming errors remains the most popular topic to investigate with Blackbox. An initial flurry of papers looked at compiler message frequency. This is a somewhat shallow analysis technique, both in the sense that it requires little processing and also that compiler error messages can change with different Java releases and between different compilers, rendering the results quite fragile.

Overall, however, topics that have been investigated are not restricted to programming errors, and it seems that the dataset is versatile. Several of the publications and survey responses made mention of using the database solely as a repository of program code without mention of education.

5.3 Data capture

One of the central challenges of the Blackbox project was that the data capture is designed independently of any individual experiment or analysis. The intention of the original design was to capture data that would support a wide variety of experiments. Therefore it is to be expected that several researchers in our survey found the data to have more information than they individually needed. However, it is clear from the survey that there is one especially common use case: the need to obtain a series of temporally ordered source code snapshots plus compile errors. This data is present in the Blackbox dataset, but perhaps more could be done to make this view of the data easier to access for new researchers. For any individual researcher, more is not better – but the set of researchers overall would be smaller if not all data was recorded. As one respondent said, "One advantage... is the detail of the data. If anything like [the unique feature they needed] was overlooked it would be problematic."

There remain several features of the Blackbox data which have seen little exploration. Blackbox includes some run-time behaviour such as exceptions and invocation results, which have not yet been used in any published work. The testing framework aspects have only been used by one publication [11] while the version control recording has also not been used. The general pattern is that IDE-agnostic, code-centric features such as source code and compilation errors have been the focus of much analysis, while the more IDE-related and workflow features (code invocations, testing, version control) have received little attention.

One of the major drawbacks of the Blackbox dataset is that the recording of programming activity is not explicitly linked to any information on what the programmer is trying to achieve, or any measure of their progress. This is inherent in the design: we record users of the BlueJ tool, and they could be using the tool for any purpose. This drawback was confirmed as an issue by existing Blackbox users in our survey, and was ranked the second most important drawback by non-users. A further issue is the lack of demographic information in Blackbox, although this was mentioned less by existing users, and ranked lower by non-users. Blackbox includes a mechanism that allows researchers to establish unique ID numbers, which could then be used to collect and link contextual and demographic data about individual users. However, only one paper [2] has used this mechanism.

Another additional mechanism is that BlueJ extensions can be implemented to record extra data (to a separate server controlled by the individual researcher) that can be tallied against Blackbox users. This mechanism has not yet been used by any researchers. It seems that in general, the convenience of the large-scale pre-existing dataset outweighs any additional gain from implementing extra tools or gathering additional information from a local study.

The highest-ranked drawback for Blackbox by non-users was that the data is Java-only, suggesting that the single language aspect (again, inherent in using BlueJ as the data source) may prevent several researchers making use of the data. One researcher responding to the survey mentioned that they ended up using Java in their work because they wanted the scale of Blackbox: "We actually had to switch languages that we supported to use Blackbox data. We used [other programming languages] to begin with and Blackbox was all Java."

5.4 Scale

The scale of Blackbox is reflected in our results as both an advantage and disadvantage. Survey respondents mentioned scale as an important aspect that helped to better generalise their results (and was ranked as the joint most important feature in a ranking exercise), and several papers used the scale of the data as a way to "sell"

their paper. However, it was also mentioned by survey respondents as causing difficulties with the analysis, leading to long-running or more complex analysis. Several uses of the country-identification feature of Blackbox in published papers felt to us like a way to simply subset the data to a more manageable size.

In some senses, for unfiltered analysis, Blackbox is now as large as it needs to be. One of the authors of this paper recently ran some analysis on the data, which took a long time to run on the full data-set: approximately 120 [wall clock] hours usage of all 12 cores on the analysis machine. For parallelisation, it was randomly sliced into a hundred sub-tasks. Each sub-task produced results that were almost exactly equal to each other (due to the law of large numbers), leading to the obvious question: if one percent of the data produces a statistically reliable sample of the full dataset, is there any need to analyse the full dataset? It should be noted that this point quickly disappears if the data is filtered: for example, only one or two percent of Blackbox users make use of BlueJ's version control support, and separately only a few percent use unit tests. Therefore, if one wanted to analyse the intersection of two such features, the dataset of interest can be much smaller than the full dataset.

5.5 Privacy

One faster way to analyse the full dataset would be to use big data tools, such as a map-reduce framework. These tools tend to introduce a tension between the faster analysis available in the cloud, versus controlling access to the data [13, 33]. It is impossible to completely anonymise source code without almost destroying it. There are examples in the Blackbox dataset of users using people's names for classes, variables, in string literals and so on. Unless every token in the data is replaced with a generic example (which would remove useful context, not to mention the technical challenges of also anonymising compiler and run-time errors and so on), it is inevitable that some of the data is not truly anonymous. For this reason, access to the data is restricted, and requests both for an anonymous subset (which we believe to be impossible without human examination) and to take copies of the data for analysis elsewhere have been refused: all primary analysis should take place on the analysis machine. Thus at present, the data access policy prevents the use of big data analysis in an external cloud or use in public data challenges such as the annual Mining Software Repositories data challenge.

5.6 Tools and techniques

A number of researchers developed software tools for themselves to aid in their analysis of the data. Theoretically, some of these tools might be shared between researchers, and the Blackbox team provides a forum to facilitate this sharing of tools and information. No-one in the survey, however, made mention of re-using tools besides those provided by the Blackbox administrators. In the published papers, some researchers re-used their own tools from previous studies, but apart from one replication [2] where tools were re-used, there was no tool sharing between researchers. The forum for the Blackbox community (the "Blackroom") receives very low traffic. This may be because there are few long-term users of the Blackbox dataset: apart from the Blackbox administrators, only two research groups have issued multiple publications using the Blackbox data. Given that each researcher seems to conduct only an isolated piece of research with the data, there is little opportunity to build cohesive sets of tools that are shared and maintained.

With machine learning being a popular area of interest with many researchers at the moment, and tools like Google's TensorFlow [1] and techniques such as deep learning gaining in popularity, it was interesting that none of the survey respondents mentioned machine learning, and only one paper [29] actually used machine learning on the data. This may be because machine learning is difficult to directly apply to program code (which has a complex and exact intra-relational structure), or because the researchers who are using the data do not have a machine learning background with the required knowledge and skills.

Anecdotal observations of the Blackbox administrators were backed up by the survey, suggesting that most of the people doing analysis on Blackbox data are undergraduate and postgraduate students. The difficulties in analysing Blackbox thus appear at "both ends": the dataset itself is a complex relational database of large size, and those who are trying to analyse it sometimes fall short of being fully competent and practised programmers.

5.7 Methodologies

Many of the Blackbox papers lacked a formal theory as the basis for their investigation, and most did not construct any kind of model from the data that could be used for future work. Many of the papers were largely exploratory in nature, analysing and reporting on the dataset without an explicit connection to wider theory. This may simply be symptomatic of being "early days" in analysing these kinds of dataset: given a new data source it is perhaps to be expected that the early work is exploratory. It may mirror a wider pattern in computing education literature, where papers have been criticised for being largely experience-centric rather than theory-driven (see Valentine [35] for the original critique, and Guzdial [14] for a rebuttal). Alternatively, it may be a sign of "looking for your keys under the street lamp": the easily available large-scale dataset may encourage researchers to build their work around the data, rather than approach the data with an existing theory.

Blackbox functions as a global dataset, but can also be used as a data collection platform for local studies by adding an identifying tag for a local population. So far, only one study [2] has made use of this possibility, and respondents ranked it as the least important feature. It is interesting that respondents generally disliked the lack of contextual and demographic information about users in the dataset, but did not seem to contemplate performing local data collection using Blackbox, which would allow them to collect such data for participants. It seems that the scale and prior availability of the data is the most important feature, and worth trading-off against the lack of context.

In our mapping study, we found two replication papers [2, 18]. Two out of eighteen papers is not a high proportion, but given that Kaijanaho [19] previously found only three replications in an analysis of forty *years* of research, it can be viewed as a promising amount.

6 CONCLUSIONS

The Blackbox dataset provided the basis of eighteen publications by ten research groups in its first five years. Some researchers who used it stated that their research could not have been carried out without Blackbox, several more said it would have been difficult to find other data, and some stated that they could not otherwise have had access to this scale of data. Thus we believe the project can be viewed as a success and a positive asset in computing education and software engineering research. It is also a vindication of the original decision to make the Blackbox open to other researchers: had its use been constrained only to the Blackbox creators, some of this research would not have been possible. Two of the Blackbox papers have been replications, which is encouraging in light of the so-called replication crisis in Psychology and other fields [10].

Researchers have found that analysis with Blackbox can be challenging. Some of these challenges are inherent: the scale of the data means that manual analysis is of limited utility, and that analysis software and database queries may need optimisation to be practical. The way that the data is collected from BlueJ users means that contextual data (such as demographic data and especially data about the user's current task) is not available, which hampers some analyses. Some of the challenges relate to trade-offs in the data: Blackbox records a wide variety of data to support multiple different use cases, but this in turn makes each individual researcher's use of the data more complicated. Finally, other challenges may be solvable by improving the analysis tools or tutorials available, especially to support common use cases such as analysis of sequences of code snapshots punctuated by compiler error data.

We believe that large data collection projects and datasets have a useful place in programming research. Blackbox demonstrates that it is possible to create a dataset decoupled from a specific purpose, and for it to provide the basis of differing studies by multiple research groups. Careful sharing of this data has enabled more research to proceed and achieve more generalisable results than otherwise would have been possible. There is, however, a careful trade-off to be made between, on the one hand, richer detail and higher granularity in the data, which may notionally enable more types of research, versus, on the other hand, a simpler data schema which makes the data easier to work with. More data may be better, but more detail not necessarily so. Additionally, there remain concerns about the "gravitational pull" of such datasets. Researchers view the easy availability of large-scale data as a positive, but there may be an opportunity cost of not investigating important questions which can only be answered with different, perhaps smaller datasets.

An interesting question is whether large-scale data only produces a quantitative shift in research results. Computing education does not really have a central model of the programming process (the plan-composition model is perhaps the most comprehensive attempt [12, 31]) or many reliable metrics (Jadud's error quotient [17] being one of the few). Thus, the way that analyses conducted on Blackbox tend to focus on concrete observables such as compiler errors, rather than the complete programming process, is completely in line with previous research. There are no signs that having larger scale data will by itself produce a qualitative shift in the types of analysis performed on student code.

For better or for worse, analysis of large-scale data like Blackbox must be done using a program. This adds the side benefits of specificity and rigour: a program is an unambiguous and reproducible way of conducting analysis. However, it also excludes the possibility of rich, nuanced qualitative analyses. The work of McCall and Kölling [23] indicated that human categorisation of errors, which does not scale, was a more promising route than simpler automatic classification – although the work did use a subset of the Blackbox data as part of its source data. This would seem to be a good model for the use of large-scale data in computing education research: not as a panacea, but as part of the classic observe-hypothesise-experiment cycle, where large-scale datasets can aid in parts of the observe and experiment phases, sometimes in tandem with additional small-scale local observations or analyses.

6.1 Future work

There are some aspects of the Blackbox dataset which have yet to be explored. This may simply be data that is of no use to anyone, but we believe that there are interesting studies which could still be conducted. For example, information on which exceptions students encounter may be interesting to explore, as would be the interplay between code execution and code editing, or test frequency and correctness. Several studies we found consider student behaviour after receiving a compiler error, but not after receiving a runtime output. We also believe that there may be interesting uses of the data involving the local data collection mechanism, which allows the addition of demographic data. Our survey, however, suggests that most researchers to date do not consider this to be a feature of particular interest.

The Blackbox project has collected nearly five years of data. BlueJ has so far maintained its popularity, and the Blackbox server is still running; there is no obvious reason why the project cannot continue for another five years. Recently, in August 2017, the first major changes were made to the database schema since the project's inception. These were necessary to accommodate the changes introduced in the release of BlueJ 4: support for recording Stride [21] data was added, as well as changes reflecting BlueJ's move to continuous background error checking. Blackbox remains available for interested researchers who wish to access the data.

ACKNOWLEDGMENTS

We are grateful to Philip Stevens and particularly to Ian Utting for their work on starting and administering the Blackbox project. We thank Amelia McNamara for making us aware of the Centre for Open Science's pre-registration facility.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-scale Machine Learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI'16)*. USENIX Association, Berkeley, CA, USA, 265–283.

[2] Alireza Ahadi, Raymond Lister, Shahil Lal, and Arto Hellas. 2018. Learning Programming, Syntax Errors and Institution-specific Factors. In *Proceedings of the 20th Australasian Computing Education Conference (ACE '18)*. ACM, New York, NY, USA, 90–96. https://doi.org/10.1145/3160489.3160490

[3] Amjad Altadmri and Neil C. C. Brown. 2015. 37 Million Compilations: Investigating Novice Programming Mistakes in Large-Scale Student Data. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15)*. ACM, New York, NY, USA, 522–527. https://doi.org/10.1145/2676723.2677258

[4] Amjad Altadmri, Michael Kölling, and Neil C. C. Brown. 2016. The Cost of Syntax and How to Avoid It: Text versus Frame-Based Editing. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 1. 748–753. https://doi.org/10.1109/COMPSAC.2016.204

[5] Ashok Basawapatna and Alexander Repenning. 2017. Employing Retention of Flow to Improve Online Tutorials. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)*. ACM, New York, NY, USA, 63–68. https://doi.org/10.1145/3017680.3017799

[6] Brett A. Becker, Cormac Murray, Tianyi Tao, Changheng Song, Robert McCartney, and Kate Sanders. 2018. Fix the First, Ignore the Rest: Dealing with Multiple Compiler Error Messages. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. ACM, New York, NY, USA, 634–639. https://doi.org/10.1145/3159450.3159453

[7] Neil C. C. Brown and Amjad Altadmri. 2014. Investigating Novice Programming Mistakes: Educator Beliefs vs. Student Data. In *Proceedings of the Tenth Annual Conference on International Computing Education Research (ICER '14)*. ACM, New York, NY, USA, 43–50. https://doi.org/10.1145/2632320.2632343

[8] Neil C. C. Brown and Amjad Altadmri. 2017. Novice Java Programming Mistakes: Large-Scale Data vs. Educator Beliefs. *Trans. Comput. Educ.* 17, 2, Article 7 (May 2017), 21 pages. https://doi.org/10.1145/2994154

[9] Neil C. C. Brown, Michael Kölling, Davin McCall, and Ian Utting. 2014. Blackbox: A Large Scale Repository of Novice Programmers' Activity. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE '14)*. ACM, New York, NY, USA, 223–228. https://doi.org/10.1145/2538862.2538924

[10] Open Science Collaboration. 2015. Estimating the reproducibility of psychological science. *Science* 349, 6251 (2015). https://doi.org/10.1126/science.aac4716 arXiv:http://science.sciencemag.org/content/349/6251/aac4716.full.pdf

[11] Draylson Micael de Souza, Michael Kölling, and Ellen Francine Barbosa. 2017. Most common fixes students use to improve the correctness of their programs. In *2017 IEEE Frontiers in Education Conference (FIE)*. 1–9. https://doi.org/10.1109/FIE.2017.8190524

[12] Kathi Fisler, Shriram Krishnamurthi, and Janet Siegmund. 2016. Modernizing Plan-Composition Studies. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)*. ACM, New York, NY, USA, 211–216. https://doi.org/10.1145/2839509.2844556

[13] Andreas Grillenberger and Ralf Romeike. 2014. Big Data – Challenges for Computer Science Education. In *Informatics in Schools. Teaching and Learning Perspectives*, Yasemin Gülbahar and Erinç Karataş (Eds.). Springer International Publishing, Cham, 29–40. https://doi.org/10.1007/978-3-319-09958-3_4

[14] Mark Guzdial. 2013. Exploring Hypotheses About Media Computation. In *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research (ICER '13)*. ACM, New York, NY, USA, 19–26. https://doi.org/10.1145/2493394.2493397

[15] C. D. Hundhausen, D. M. Olivares, and A. S. Carter. 2017. IDE-Based Learning Analytics for Computing Education: A Process Model, Critical Review, and Research Agenda. *ACM Trans. Comput. Educ.* 17, 3, Article 11 (Aug. 2017), 26 pages. https://doi.org/10.1145/3105759

[16] Petri Ihantola, Arto Vihavainen, Alireza Ahadi, Matthew Butler, Jürgen Börstler, Stephen H. Edwards, Essi Isohanni, Ari Korhonen, Andrew Petersen, Kelly Rivers, Miguel Ángel Rubio, Judy Sheard, Bronius Skupas, Jaime Spacco, Claudia Szabo, and Daniel Toll. 2015. Educational Data Mining and Learning Analytics in Programming: Literature Review and Case Studies. In *Proceedings of the 2015 ITiCSE on Working Group Reports (ITICSE-WGR '15)*. ACM, New York, NY, USA, 41–63. https://doi.org/10.1145/2858796.2858798

[17] Matthew C. Jadud. 2006. Methods and Tools for Exploring Novice Compilation Behaviour. In *Proceedings of the Second International Workshop on Computing Education Research (ICER '06)*. ACM, New York, NY, USA, 73–84. https://doi.org/10.1145/1151588.1151600

[18] Matthew C. Jadud and Brian Dorn. 2015. Aggregate Compilation Behavior: Findings and Implications from 27,698 Users. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research (ICER '15)*. ACM, New York, NY, USA, 131–139. https://doi.org/10.1145/2787622.2787718

[19] Antti-Juhani Kaijanaho. 2014. The extent of empirical evidence that could inform evidence-based design of programming languages: A systematic mapping study. *Jyväskylä licentiate theses in computing; 1795-9713; 18.* (2014).

[20] Hieke Keuning, Bastiaan Heeren, and Johan Jeuring. 2017. Code Quality Issues in Student Programs. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '17)*. ACM, New York, NY, USA, 110–115. https://doi.org/10.1145/3059009.3059061

[21] Michael Kölling, Neil Brown, and Amjad Altadmri. 2017. Frame-Based Editing. *Journal of Visual Languages and Sentient Systems* 3 (July 2017), 40–67. http://doi.org/10.18293/VLSS2017-012

[22] Prathmi Kurtiker and Ramrao Wagh. 2016. Understanding and analyzing students frustration level during programming. In *Proceedings of the 24th International Conference on Computers in Education*. Asia-Pacific Society for Computers in Education, 7–9.

[23] Davin McCall and Michael Kölling. 2014. Meaningful categorisation of novice programmer errors. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. 1–8. https://doi.org/10.1109/FIE.2014.7044420

[24] Olfat M. Mirza, Mike Joy, and Georgina Cosma. 2017. Style Analysis for Source Code Plagiarism Detection – An Analysis of a Dataset of Student Coursework. In *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)*. 296–297. https://doi.org/10.1109/ICALT.2017.117

[25] Olfat M. Mirza, Mike Joy, and Georgina Cosma. 2017. Suitability of BlackBox dataset for style analysis in detection of source code plagiarism. In *2017 Seventh International Conference on Innovative Computing Technology (INTECH)*. 90–94. https://doi.org/10.1109/INTECH.2017.8102424

[26] Cormac Murray. 2016. *A Comparative Study of Java Compiler Error Profiles Using the Blackbox Dataset.* Master's thesis. University College Dublin.

[27] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology* 64 (2015), 1 – 18. https://doi.org/10.1016/j.infsof.2015.03.007

[28] David Pritchard. 2015. Frequency Distribution of Error Messages. In *Proceedings of the 6th Workshop on Evaluation and Usability of Programming Languages and Tools (PLATEAU 2015)*. ACM, New York, NY, USA, 1–8. https://doi.org/10.1145/2846680.2846681

[29] Eddie Antonio Santos, Joshua Charles Campbell, Dhvani Patel, Abram Hindle, and José Nelson Amaral. 2018. Syntax and Sensibility: Using language models to detect and correct syntax errors. In *25th IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER 2018)*, Vol. 29. 1–5.

[30] Stewart D. Smith, Nicholas Zemljic, and Andrew Petersen. 2015. Modern Goto: Novice Programmer Usage of Non-standard Control Flow. In *Proceedings of the 15th Koli Calling Conference on Computing Education Research (Koli Calling '15)*. ACM, New York, NY, USA, 171–172. https://doi.org/10.1145/2828959.2828980

[31] Elliot Soloway. 1986. Learning to Program = Learning to Construct Mechanisms and Explanations. *Commun. ACM* 29, 9 (Sept. 1986), 850–858. https://doi.org/10.1145/6592.6594

[32] Jaime Spacco, Paul Denny, Brad Richards, David Babcock, David Hovemeyer, James Moscola, and Robert Duvall. 2015. Analyzing Student Work Patterns Using Programming Exercise Data. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15)*. ACM, New York, NY, USA, 18–23. https://doi.org/10.1145/2676723.2677297

[33] Bhavani Thuraisingham. 2015. Big Data Security and Privacy. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy (CODASPY '15)*. ACM, New York, NY, USA, 279–280. https://doi.org/10.1145/2699026.2699136

[34] Ian Utting, Neil Brown, Michael Kölling, Davin McCall, and Philip Stevens. 2012. Web-scale Data Gathering with BlueJ. In *Proceedings of the Ninth Annual International Conference on International Computing Education Research (ICER '12)*. ACM, New York, NY, USA, 1–4. https://doi.org/10.1145/2361276.2361278

[35] David W. Valentine. 2004. CS Educational Research: A Meta-analysis of SIGCSE Technical Symposium Proceedings. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '04)*. ACM, New York, NY, USA, 255–259. https://doi.org/10.1145/971300.971391

[36] Claes Wohlin. 2014. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE '14)*. ACM, New York, NY, USA, Article 38, 10 pages. https://doi.org/10.1145/2601248.2601268