Robust Correctness Testing for Digital Forensic Tools

Lei Pan and Lynn M. Batten

School of EIT, Deakin University, Burwood, Victoria 3125, Australia
{l.pan,lmbatten}@deakin.edu.au

Abstract. In previous work, the authors presented a theoretical lower bound on the required number of testing runs for performance testing of digital forensic tools. We also demonstrated a practical method of testing showing how to tolerate both measurement and random errors in order to achieve results close to this bound. In this paper, we extend the previous work to the situation of correctness testing.

The contribution of this methodology enables the tester to achieve correctness testing results of high quality from a manageable number of observations and in a dynamic but controllable way. This is of particular interest to forensic testers who do not have access to sophisticated equipment and who can allocate only a small amount of time to testing.

Keywords: digital forensics, correctness testing, data carving tools.

1 Introduction

Working from a legal perspective, digital forensics is one of the most potent deterrents to digital crime. Within the last ten years, more than a dozen definitions of digital forensics have been proposed; however, the one common element in all these definitions is the preparation of evidence for presentation in a court of law. In courtrooms, witnesses present the facts of a case as perceived by them. But a witness sometimes serves as an "expert" in giving personal opinions about what has been found or observed during a digital investigation. Such opinions are formed on the basis of professional experience and deductive reasoning.

A digital forensic expert must be familiar with many forensic tools, but no expert can know or use all of the forensic tools available. Questions regarding digital forensic software tools used in an investigation are often asked in the courtroom. Such questions may be phrased as: "have you personally used tool A?"; "did you use tool B because it is faster than tool A?"; "among tools A, B and C, which tool performs better in assisting this case?"; and so on. The judge, as well as lawyers on opposing sides, may be very interested in the answers to these questions in order to find possible flaws or errors in the reasoning.

Where can the forensic expert obtain information about the effectiveness of the tools he chooses to use? Current testing work is led by a few official organizations [2,3,4] often government supported, with many results unavailable to the

M. Sorell (Ed.): e-Forensics 2009, LNICST 8, pp. 54–64, 2009.

general public, or only published for tools which have become commonly used. Mohay [9] has argued that the increasing time gap between available testing results and the release of testing results of new tools is a major reason why newly developed tools are rarely accepted into general digital forensic practice.

The focus of this paper is on developing a simple way of helping an expert witness in digital forensics to implement some standard software tests in order to be ready with answers for questions such as those above. Because the quality of software tools depend on many parameters, testing paradigms vary on the basis of the tester's intention; thus a test may be aimed at performance, correctness, reliability or security.

In previous work [10] the authors focused on testing the performance of tools. In this paper, we focus on correctness testing in the context of the forensic tool testing discussed in [1,15]. A correctness test evaluates how much the functionality of a tool differs from the user's expectations. The definition of correctness and the results of any test for correctness must both be measurable. In this paper, we define 'correctness' as both *accuracy* and *precision* of the tool.

We phrase our problem as: how can an expert witness without any specialized equipment quickly and correctly gain knowledge of the capacity for correctness of a given set of digital forensic tools? We propose an effective and efficient correctness testing framework. Our framework regulates what digital forensic tools should be compared in one experiment and it determines a testing plan prior to the experiment so that the tester can balance the test effort against the accuracy of the test results Moreover, the framework interprets the test results and includes necessary conditions for their reproducibility. The key contributions of this work are two-fold: (a) the development of a correctness testing methodology for digital forensic tools achieving effectiveness and efficiency simultaneously, and (b) theoretical and practical contributions for forensic expert witnesses from the findings of a case study.

Our methodology on executing testing plans is described in Section 2. Section 3 provides an example to illustrate our methodology. The paper is summarized in Section 4.

2 Our Approach

In previous work [10,11], the authors described the background of forensic tool testing as a scientific method and introduced orthogonal arrays [7] as an approach to organizing testing runs. We shall use the same terminology as in that paper. In this section, we demonstrate how the use of orthogonal arrays gives us a flexible methodology for testing correctness of forensic tools. The approach is dynamic and improves the quality of testing results during the execution. Our methodology reduces the impact of inaccurate observations with large values by using Taguchi's logarithmic function [14]; it detects outliers by engaging paired Student t-tests [16]; and it determines the number of necessary observations by applying recent statistical findings.

In an array, each parameter corresponds to a column and each test performed corresponds to a row. The entries in the array are the values taken on by the

parameters. These entries are placed according to constraints set by the testing requirements. In general, we wish only that the constraints are those of fairness (each value of every parameter of the system should be tested an equal number of times) and blindness (no conclusions are drawn until the entire test is ended). These conditions are the primary focus of special types of arrays known as 'orthogonal arrays' or OAs. In what follows, N will refer to the number of rows and k to the number of columns of the array. The entries in the array can be taken from any set S of size s. In the general case, there is an additional parameter t known as the *strength*, but it is sufficient for our purposes to always use $t = 2$. We can now give the definition due to [7]: An $OA(N, k, s, 2)$ is an $N \times k$ array with entries from S such that

F every $N \times 2$ sub-array contains each pair of elements of S the same number of times as a row.

In other words, any row of the array restricted to any two columns appears the same number of times in the restriction. An OA allows us to test the performance of a tool against itself, or to test it against other tools on the same parameter set.

We lay out two key strategies to tackle the testing error issue. First, we set a simple criterion to identify the potentially contaminated observations, referred to as *suspicious samples*. The negative effect caused by suspicious samples will be mitigated by introducing additional observations with fewer contaminated ones. We increase the robustness of testing results by using the logarithmic function in the Taguchi method [14], and predict the total number of testing observations by using Zhou and Zhu's result in [17]. Second, we alter the standard execution order of a testing array from row-wise to column-wise. This technique enables additional observations to be introduced on demand and so a tester can improve the quality of testing results without interrupting the experiment.

2.1 Reducing the Impact of Errors

In any experiment, errors are inevitable. Generally, errors are of two types — measurement error and random error. A measurement error is any bias caused by the observation methods or instrument used. A random error is caused by the variation of the experimental environment. The first type of error can be dealt with by good calibration, while the second is more difficult to handle. In either case, data outliers will occur once the error has accumulated to a certain level. An outlier is "an observation that lies an abnormal distance from other values" [8]. The effect of outliers is to distort the results of the test, and so their impact must be reduced. There are several ways to effective reduction of outlier impact, but we choose that of Taguchi et al. in [13,14].

Taguchi proposed the use of the logarithmic function $f(x) = -10 \log(x^2)$ [13], which has the effect of reducing the impact of error for real input values of $x > 1$, but increasing the impact for $0 < x < 1$. Thus, use of the logarithmic scale makes it increasingly important to obtain accurate results with small values. When less precise measuring devices are used to obtain observation results, a measuring error may introduce an outlier which strongly degrades the quality of the results.

In order to identify suspicious data, we therefore propose the following algorithm which is applied to the logarithmic version of the data.

Algorithm input: a set of test output observations in real numbers and a known (or estimated) margin of error caused by measuring errors.
Algorithm output: a set of suspicious data in real numbers.
Begin Algorithm

Step 1. Subtract the margin of error from each value in the set of observations to obtain a second set of data.
Step 2. Apply the paired Student t-test on the two data sets.
Step 3. If the two data sets are not significantly different according to the t-test, then output an empty set and terminate the algorithm; otherwise, proceed to *Step 4*.
Step 4. Output a set containing the smallest observation from the first set, repeated as many times as it occurs. Then go back to *Step 1* using an input the difference between the first set and the output set.

This algorithm will eventually terminate and output a set of real numbers or an empty set. If the output set is nonempty, we define the elements in the set as *suspicious samples*. If suspicious samples are detected, then we will need to improve the quality of testing results by ensuring enough instances of observations.

The next subsection will discuss how to obtain an adequate number of observations.

2.2 Determining the Number of Tests to Run

To reduce the impact of outliers, merely applying Taguchi's logarithmic function is insufficient because observations with small values have to be accurate. To improve the quality of these observations, more test runs will reduce the impact caused by any random error, and an appropriate statistical analysis reduces the impact caused by any measurement error.

Zhou and Zhu in [17] determined the relationship between the number of suspicious samples and the number of observations. Their Theorem 1 is restated in the following theorem in order to fit our situation.

Theorem 1. *Suppose that an experimental design has N rows. Then for any integer $R \geq 1$, in order for $N \times R$ observations to withstand the impact caused by the measure error and the random error, the number of suspicious samples should not exceed*

$$N_c = \min\left\{ \left\lfloor \frac{N-1}{2} \right\rfloor + \left\lfloor \frac{N+1}{2} \right\rfloor \cdot \left\lfloor \frac{R-1}{2} \right\rfloor,\ N \cdot \left\lfloor \frac{R-1}{2} \right\rfloor \right\},$$

where $\lfloor x \rfloor$ denotes the largest integer less than or equal to x.

Specifically, there are $\lfloor \frac{N-1}{2} \rfloor + \lfloor \frac{N+1}{2} \rfloor \cdot \lfloor \frac{R-1}{2} \rfloor$ outliers caused by the measurement error and $N \cdot \lfloor \frac{R-1}{2} \rfloor$ outliers caused by the random error.

Theorem 1 indicates that increasing the number of observations increases the overall credibility of the results. This theorem also indicates that a large product of the N value (the row size of the testing array) and the R value (the replication number) is crucial for tolerating experimental errors. This theorem implies that the R value must be equal or larger than 2 to tolerate outliers caused by the measurement error, and must be equal or larger than 3 to tolerate outliers caused by the random error.

We use the results of this theorem to determine the necessary number of observations for an experiment so that the impact of outliers will be negligible. The next subsection presents our new adaptive procedure for executing an experiment.

2.3 Our Adaptive Procedure

Consolidating the above ideas and results, we present an adaptive procedure for conducting tests. The procedure helps the tester to obtain high-quality testing results based on a given testing array. Our procedure ensures that the overall quality of the results is good enough for deriving reliable and trustworthy conclusions, because this procedure reduces the impact of outliers caused by experimental errors to a negligible level.

Procedure input: an $N \times k$ orthogonal array associated with the experiment.
Procedure output: $N \times R$ observations.
Begin Procedure

Step 1. Execute the array once so that each row will be tested. After this step, the tester should have conducted N observations associated to every row of the array.

Step 2. Transform the results by using Taguchi's logarithmic function and check for suspicious samples in accordance with Section 2.1. If there are no suspicious samples, go to *Step 6*; otherwise, letting m be the number of suspicious samples, proceed to the next step.

Step 3. Compare m with the value of N_c in Theorem 1, and increase R to the smallest value of R such that $m \leq N_c$.

Step 4. Let R' be the number of observations which have already been made. Retest using the array $R-(R'+1)$ times. The execution order is column-wise. (Input may be changed at this stage; see the example of Section 3.)

Step 5. Transform the new results using the logarithmic function and search for suspicious samples again. Newly identified suspicious samples should be counted before proceeding to the next step.

Step 6. Test each row of the array for the last time, transform the results and do a final check for suspicious samples. If the number of suspicious samples exceeds the number identified in Theorem 1, then go back to *Step 3*; otherwise, the experiment is completed.

End Procedure

Our methodology provides a cure for the error-handling problem suffered by the Taguchi method. It has the following advantages over these two methods:

– Our method fulfills the blindness requirement. That is, no conclusions are drawn before a test is completed.
– Our method does not waste testing effort. That is, every observation is kept, so that the tester knows the exact testing effort and the complete set of results.
– Our method is more automated than Taguchi model. Our method does not require the tester to determine the replication number before the start of a test, and does not pause the test before an adequate number of observations are obtained.

3 Correctness Testing of Data Carving Tools

To illustrate the procedure of obtaining good-quality results from a comparative experiment, we conducted a correctness test on data carving tools. We chose the number of false positives as the only metric on which to compare the correctness across the tool set. If a tester wishes to test additional items, these must be dealt with one at a time. In particular, we evaluated the correctness of these tools by the number of unsuccessfully recovered files. In this experiment, a data carving tool is a tool whose input data format is the raw image of a disk partition and whose output data format is that of the recovered file.

We chose five data carving tools. The choice of tools is represented as sequential integers. In this experiment, we included 5 scenarios:

0 *Foremost*
1 *FTK*
2 *Magicrescue*
3 *Scalpel*
4 *X-Way Forensics*

An OA whose total number of rows is divisible by 5 was required if each of these five scenarios was tested once; accordingly, if three scenarios were tested twice and two scenarios were tested once, then the number of rows should be divisible by 8.

We considered recovering 3 common types of files — MS word documents, jpeg files and MS xls files. By excluding (labeled as "0") or including (labeled as "1") these types of files in the testing input image, we had a total of 3 binary parameters in the experiment, and so needed an OA with at least 3 columns and 6 rows.

We iterated through the online OA library [12,10]. No 5×4 or 6×4 OA was available. We found several options with 16 rows satisfying the **F** condition, and we did not choose any OA with more rows to save the overall testing effort. The closest fit is the 16×9 array listed below. It has one column with 8 variables which we list first and use to allocate tools to be tested, and 8 columns with binary variables. The OA is indexed as

$$16 \qquad 4(*) \quad 2^8 \quad 8^1$$

in table 1 on the page `http://www.research.att.com/~njas/doc/cent4.html`.

0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 1 1
1 0 0 0 0 1 1 1 1
1 1 1 1 1 0 0 0 0
2 0 0 1 1 0 0 1 1
2 1 1 0 0 1 0 0 0
3 0 0 1 1 1 1 0 0
3 1 1 0 0 0 0 1 1
4 0 1 0 1 0 1 0 1
4 1 0 1 0 1 0 1 0
5 0 1 0 1 1 0 1 0
5 1 0 1 0 0 1 0 1
6 0 1 1 0 0 1 1 0
6 1 0 0 1 1 0 0 1
7 0 1 1 0 1 0 0 1
7 1 0 0 1 0 1 1 0

We arbitrarily related each of the columns (other than the first) to one of the 3 binary parameters and deleted the last 5 columns. In terms of redundant symbols ("5", "6" and "7") in the first column, we substituted them with the first three scenarios ("0", "1" and "2") respectively.

We compared the MD5 hash values of the recovered files and those of the original files. The possible MD5 collision indicated that we should employ 1 miscounted file per 1,000 files as margin of error. Moreover, we assume a base false positive rate is 0.002. Then we followed the procedure in Section 2.3 step by step:

Step 1. By following the array instructions using the DFTT testing image #11 [6], we obtained 16 observations (one for each row). In order from the first row to the last row, the false negatives were:

$$0.002, 1.002, 0.002, 0.002, 0.002, 1.002, 0.002, 1.002,$$

$$1.002, 0.002, 1.002, 0.002, 0.002, 0.002, 1.002, 0.002.$$

Step 2. These 16 numbers were transformed by using Taguchi's logarithmic function to obtain the sequence:

$$53.979, -0.017, 53.979, 53.979, 53.979, -0.017, 53.979, -0.017,$$

$$-0.017, 53.979, -0.017, 53.979, 53.979, 53.979, -0.017, 53.979.$$

Subtracting the margin of error 0.0005 from the results in *Step 1*, we obtain possibly contaminated observations as:

$$0.001, 1.001, 0.001, 0.001, 0.001, 1.001, 0.001, 1.001,$$

$$1.001, 0.001, 1.001, 0.001, 0.001, 0.001, 1.001, 0.001.$$

By applying Taguchi's function to the second set of observations adjusted for error, we then obtained a second transformed sequence:

$$60.000, -0.009, 60.000, 60.000, 60.000, -0.009, 60.000, -0.009,$$

$$-0.009, 60.000, -0.009, 60.000, 60.000, 60.000, -0.009, 60.000.$$

A paired t-test indicated a significant difference between these two sequences (t = -5.0106, df = 15 with p-value = 0.0001551). By following the outlier detection algorithm mentioned in Section 2.1, we removed the observations of the least value from the original observation set and identified ten potential outliers which are equal to 0.002 in *Step 1*.

So we proceed to *Step 3*.

Step 3. To tolerate 10 potential outliers, Theorem 1 requires 16×3 observations. And the number of actual outliers which can be handled by these observations is

$$\min\left\{\left\lfloor\frac{16-1}{2}\right\rfloor + \left\lfloor\frac{16+1}{2}\right\rfloor \cdot \left\lfloor\frac{3-1}{2}\right\rfloor,\ 16 \cdot \left\lfloor\frac{3-1}{2}\right\rfloor\right\} = \min\{7+8\cdot 1, 16\cdot 1\} = 15.$$

Step 4. By using the DFRWS-06 challenge image file [5], we ran the test array once more and obtained 16 new observations. From the first row to the last, the false positives were:

$$0.002, 13.002, 0.002, 8.002, 1.002, 10.002, 1.002, 5.002,$$

$$6.002, 3.002, 10.002, 3.002, 4.002, 4.002, 9.002, 2.002.$$

Step 5. We found no more suspicious samples in the available 32 observations other than the ten observations identified in *Step 2*.

Step 6. By running the test array for the last time and using the DFTT testing image #12 [6], we obtained another 16 observations (one for each row). From the first row to the last, the false positives were:

$$0.002, 0.002, 0.002, 3.002, 1.002, 2.002, 1.002, 1.002,$$

$$2.002, 2.002, 0.002, 0.002, 2.002, 1.002, 2.000, 1.002.$$

There are five suspicious samples in this set of observations: the five smallest values equal to 0.002. Therefore, we have in total 15 potential outliers in this experiment. Our 48 observations are obtained in 3 group tests, so our experimental results can tolerate these 15 outliers as calculated in *Step 3* according to Theorem 1. So we completed the experiment with this set of 48 observations shown in Table 1.

The average execution times in each testing run varied in a relatively wide range from zero to 5, as shown in Figure 1. Variations in false positives between different runs were evident — the three largest values were 4.669, 4.335 and

Table 1. The Experimental Results of the 16-Run Correctness Test for the Data Carving Tools

Row	Group 1	Group 2	Group 3
1	0.002	0.002	0.002
2	1.002	13.002	0.002
3	0.002	0.002	0.002
4	0.002	8.002	3.002
5	0.002	1.002	1.002
6	1.002	10.002	2.002
7	0.002	1.002	1.002
8	1.002	5.002	1.002
9	1.002	6.002	2.002
10	0.002	3.002	2.002
11	1.002	10.002	0.002
12	0.002	3.002	0.002
13	0.002	4.002	2.002
14	0.002	4.002	1.002
15	1.002	9.002	2.002
16	0.002	2.002	1.002

4.002 observed respectively in run 2, run 4 and run 15; the three smallest values were 0.002, 0.002 and 0.669 observed respectively in run 1, run 3 and run 5.

The observed values in runs 1 and 2, runs 10 and 11, and runs 7 and 8 were consistently small. This corresponded to the two most efficient carving tools — *Foremost* and *Scalpel*. In fact, *Scalpel* was developed based on *Foremost*. Besides the choice of tools, the inclusion of jpeg files also had significant impact on the false positives.

In this test, the first group of observations provided much inaccurate information as the DFTT testing image has a broad mixture of files; however, we managed to achieve the correct results by introducing extra observations with

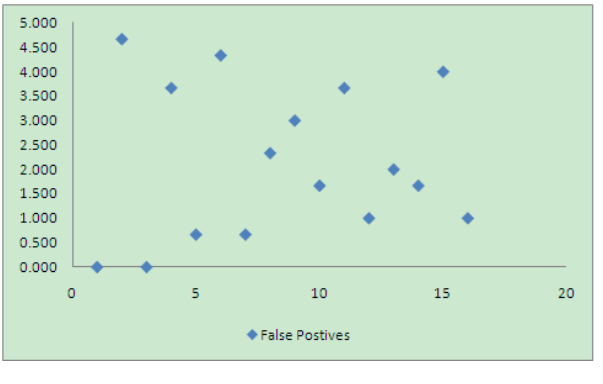

Fig. 1. Average False Positives of Data Carving Tools Observed in the 16 Testing Runs

better quality. Therefore, the overall quality of testing data was good enough to reveal accurate information about the correctness of the tools.

4 Conclusions and Future Work

A robust correctness testing method for digital forensic tools was presented in this paper. In order to deliver the most accurate result, we demonstrated that multiple observations are necessary. Moreover, we have shown that the choice of OA determines the testing pattern and provides the number of testing cases; our adaptive experimental approach guarantees that testers achieve testing results of good quality through a limited number of observations.

Our approach enables general testers to compare the performance of digital forensic tools without using sophisticated equipment or spending a large amount of time. The testing results have proved the validity and the effectiveness of our methodology. Most importantly, our methodology can be fully automated so that a computerized tester may be developed to replace human testers in the future.

Acknowledgment

The authors would like to acknowledge the constructive comments made by the anonymous referees and the help from the conference organizers.

References

1. Beckett, J., Slay, J.: Digital Forensics: Validation and Verification in a Dynamic Work Environment. In: Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS 2007), p. 266a (2007)
2. CFTT group from NIST. Disk Imaging Specifications. NIST technial report (2001)
3. CFTT group from NIST. Digital Data Acquisition Tool Specification. NIST technial report (2004)
4. CFTT group from NIST. Digital Data Acquisition Tool Test Assertions and Test Plan. NIST technial report (2005)
5. Digital Forensics Research Workshop. DFRWS06 Forensic Challenge, `http://www.dfrws.org/2006/challenge/index.shtml`
6. Brian Carrier. Digital Forensics Tool Testing Images, `http://dftt.sourceforge.net/`
7. Hedayat, A.S., Sloane, N.J.A., Stufken, J.: Orthogonal Arrays: Theory and Applications. Springer, Heidelberg (1999)
8. NIST/SEMATECH. e-Handbook of Statistical Methods, `http://www.itl.nist.gov/div898/handbook/`
9. Mohay, G.: Technical Challenges and Directions for Digital Forensics. In: Proceedings of the 1st International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE 2005), pp. 155–161 (2005)
10. Pan, L., Batten, L.M.: A Lower Bound on Effective Performance Testing for Digital Forensic Tools. In: Proceedings of the 2nd International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE 2007), pp. 117–130 (2007)

11. Pan, L.: A Performance Testing Framework for Digital Forensic Tools. PhD thesis Deakin University (2007)
12. Sloane, N.J.A.: A Library of Orthogonal Arrays, http://www.research.att.com/~njas/oadir/index.html
13. Taguchi, G.: Introduction to Quality Engineering: Designing Quality Into Produces and Processes. White Plains (1986)
14. Taguchi, G., Chowdhury, S., Wu, Y.: Taguchi's Quality Engineering Handbook. Wiley, Chichester (2004)
15. Wilsdon, T., Slay, J.: Digital Forensics: Exploring Validation, Verification and Certification. In: Proceedings of the 1st International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE 2005), pp. 48–55 (2005)
16. Youden, W.J.: Statistical techniques for collaborative tests. In: Statistical Manual of the Association of Official Analytical Chemists, p. v–63 (1975)
17. Zhou, J., Zhu, H.: Robust Estimation and Design Procedures for the Random Effects Model. The Canadian Journal of Statistics 31(1), 99–110 (2003)