The Journal of Computing Sciences in Colleges

Papers of the 30th Annual CCSC South Central Conference

April 5th, 2019
The University of Texas at Dallas
Dallas, TX

Volume 34, Number 5 **April 2019**

The Journal of Computing Sciences in Colleges (ISSN 1937-4771 print, 1937-4763 digital) is published at least six times per year and constitutes the refereed papers of regional conferences sponsored by the Consortium for Computing Sciences in Colleges. Printed in the USA. POSTMASTER: Send address changes to Susan Dean, CCSC Membership Secretary, 89 Stockton Ave, Walton, NY 13856.

Table of Contents

The Consortium for Computing Sciences in Colleges Board of Directors

Following is a listing of the contact information for the members of the Board of Directors and the Officers of the Consortium for Computing Sciences in Colleges (along with the years of expiration of their terms), as well as members serving CCSC:

Jeff Lehman, President (2020), (260)359-4209, jlehman@huntington.edu, Mathematics and Computer Science Department, Huntington University, 2303 College Avenue, Huntington, IN 46750.

Karina Assiter, Vice President (2020), (802)387-7112, karinaassiter@landmark.edu.

Baochuan Lu, Publications Chair (2021), (417)328-1676, blu@sbuniv.edu, Southwest Baptist University - Department of Computer and Information Sciences, 1600 University Ave., Bolivar, MO 65613.

Brian Hare, Treasurer (2020), (816)235-2362, hareb@umkc.edu, University of Missouri-Kansas City, School of Computing & Engineering, 450E Flarsheim Hall, 5110 Rockhill Rd., Kansas City MO 64110.

Susan Dean, Membership Secretary (2019), Associate Treasurer, (607)865-4017, Associate Editor, susandean@frontier.com, UMUC Europe Ret, US Post: 89 Stockton Ave., Walton, NY 13856.

Judy Mullins, Central Plains Representative (2020), Associate Treasurer, (816)390-4386, mullinsj@umkc.edu, School of Computing and Engineering, 5110 Rockhill Road, 546 Flarsheim Hall, University of Missouri - Kansas City, Kansas City, MO 64110.

John Wright, Eastern Representative (2020), (814)641-3592, wrightj@juniata.edu, Juniata College, 1700 Moore Street, Brumbaugh Academic Center, Huntingdon, PA 16652.

David R. Naugler, Midsouth Representative(2019), (573) 651-2787, dnaugler@semo.edu, 5293 Green Hills Drive, Brownsburg IN 46112.

Lawrence D'Antonio, Northeastern Representative (2019), (201)684-7714, ldant@ramapo.edu, Computer Science Department, Ramapo College of New Jersey, Mahwah, NJ 07430.

Cathy Bareiss, Midwest Representative (2020), cbareiss@olivet.edu, Olivet Nazarene University, Bourbonnais, IL 60914.

Brent Wilson, Northwestern Representative (2021), (503)554-2722, bwilson@georgefox.edu, George Fox University, 414 N. Meridian St, Newberg, OR 97132.

Mohamed Lotfy, Rocky Mountain Representative (2019), Information Technology Department, College of Computer & Information Sciences, Regis University, Denver, CO 80221.

Tina Johnson, South Central Representative (2021), (940)397-6201, tina.johnson@mwsu.edu, Dept. of Computer Science, Midwestern State University, 3410 Taft Boulevard, Wichita Falls, TX 76308-2099.

Kevin Treu, Southeastern Representative (2021), (864)294-3220, kevin.treu@furman.edu, Furman University, Dept of Computer Science, Greenville, SC 29613.

Bryan Dixon, Southwestern Representative (2020), (530)898-4864, bcdixon@csuchico.edu, Computer Science Department, California State University, Chico, Chico, CA 95929-0410.

Serving the CCSC: These members are serving in positions as indicated:

Brian Snider, Associate Membership Secretary, (503)554-2778, bsnider@georgefox.edu, George Fox University, 414 N. Meridian St, Newberg, OR 97132.

Will Mitchell, Associate Treasurer, (317)392-3038, willmitchell@acm.org, 1455 S. Greenview Ct, Shelbyville, IN 46176-9248.

John Meinke, Associate Editor, meinkej@acm.org, UMUC Europe Ret, German Post: Werderstr 8, D-68723

Oftersheim, Germany, ph 011-49-6202-5777916.
Shereen Khoja, Comptroller, (503)352-2008, shereen@pacificu.edu, MSC 2615, Pacific University, Forest Grove, OR 97116.
Elizabeth Adams, National Partners Chair, adamses@jmu.edu, James Madison University, 11520 Lockhart Place, Silver Spring, MD 20902.
Megan Thomas, Membership System Administrator, (209)667-3584, mthomas@cs.csustan.edu, Dept. of Computer Science, CSU Stanislaus, One University Circle, Turlock, CA 95382.
Deborah Hwang, Webmaster, (812)488-2193, hwang@evansville.edu, Electrical Engr. & Computer Science, University of Evansville, 1800 Lincoln Ave., Evansville, IN 47722.

CCSC National Partners

The Consortium is very happy to have the following as National Partners. If you have the opportunity please thank them for their support of computing in teaching institutions. As National Partners they are invited to participate in our regional conferences. Visit with their representatives there.

Platinum Partner

Turingscraft
Google for Education
GitHub
NSF – National Science Foundation

Silver Partners

zyBooks

Bronze Partners

National Center for Women and Information Technology
Teradata
Mercury Learning and Information

Foreword

Welcome to the 2019 issues of our journal for the CCSC spring 2019 conferences: Southwestern (March 22-23), Central Plains (April 5-6), South Central (April 5), Mid-south (April 12-13), and Northeastern (April 12-13).

Please plan to attend one or more conferences, where you can meet and exchange ideas with like-minded computer science educators. Each conference covers a variety of topics that are practical and stimulating. You can find detailed conference programs on the conference websites, which are listed on the CCSC conferencecalendar: `http://www.ccsc.org/regions/calendar.`

From January 2019, this journal will be published electronically on the CCSC website and links to the journal issues will be sent to CCSC members via email. Those of you who would like hard copies of journal issues can order them from Amazon. Simply search for "CCSC Journal" to find available issues. The journal will continue to be available in the ACM Digital Library.

As an author, you may post your papers published by CCSC on any website. Please make sure to use the PDF versions of your papers with CCSC's copyright box. Such PDFs can be downloaded from the ACM Digital Library or extracted from our electronic journal.

Please feel free to email me directly at `blu@sbuniv.edu` if you notice any issue with our publications.

Baochuan Lu
Southwest Baptist University
CCSC Publications Chair

Welcome to the 2019 CCSC South Central Conference

The 2019 South Central Steering Committee is very pleased to welcome everyone to our 30th annual conference in Richardson, Texas hosted by the University of Texas at Dallas. Our conference chair and host, Sam Karrah, has provided wonderful facilities for the conference and the support of his department and staff has been outstanding. Several faculty and staff at the University of Texas at Dallas have given extensive time and generous efforts to our conference and committee throughout this past year. Their efforts and congeniality are very much appreciated.

This year we have eight papers, two workshops, several lightning talks, and both student and faculty posters on the program. The Steering Committee chose 8 of 14 papers through a double-blind review process for a paper acceptance rate of 57%. We had 21 colleagues across the region and country serve as professional reviewers and we recognize their generous efforts in providing time and guidance in the selection of our conference program.

The Steering Committee invites colleagues to host the conference in the future and to join our community of computer science educators to enrich our curricula and provide innovative pedagogy for our students. We encourage other members of the South Central region to attend our Friday evening business meeting and to join in our efforts to bring in fellow colleagues who wish to be involved in the planning and execution of the conference in the future.

We extend a very warm and delightful welcome to all presenters and attendees and encourage everyone to enjoy our program and the University of Texas at Dallas. Thank you again to all members of the 2019 Steering Committee who continue to provide the necessary time and dedication to the conference with grace and cherished commitment.

Sam Karrah
University of Texas at Dallas
Conference Chair and Host

Laura J. Baker
St. Edward's University
Papers and Program Chair

2019 CCSC South Central Conference Steering Committee

Shyam Karrah, Host and Chair ..... The University of Texas at Dallas
Anne Marie Eubanks, Registrar ..... Stephen F. Austin State University
Tina Johnson, National Board Representative .. Midwestern State University
Bilal Shebaro, Treasurer ..... St. Edward's University
Laura Baker, Papers/Program Chair, ... St. Edward's University
Tim McGuire, Panels and Tutorials Chair, ... Texas A&M University
Michael Scherger, Posters Chair, ... Texas Christian University
Bing Yang Wei, Moderators Chair, ... Texas Christian University
Abena Primo, At-Large Member, ... Huston-Tillotson University
Eduardo Colmenares-Diaz, Publicity Chair, ... Midwestern State University
Michael Kart, Lightning Talks Chair, ... St. Edward's University
Vipin Menon, Webmaster, ... McNeese State University
Michael Scherger, Past Conference Chair ..... Texas Christian University

Reviewers — 2019 CCSC South Central Conference

Srinivasarao Krishnaprasad ... Jacksonville State University, Jacksonville, AL
Anne Marie Eubanks . Stephen F. Austin State University, Nacogdoches, TX
David Gurney ..... Southeastern Louisiana University, Hammond, LA
Sikha Bagui ..... University of West Florida, Pensacola, FL
Barbara Anthony ..... Southwestern University, Georgetown, TX
Kenneth Rouse ..... LeTourneau University, Longview, TX
Michael Kart ..... St. Edward's University, Austin, TX
Jose Metrolho ..... IPCB Portugal, Castelo Branco, CB
Vipin Menon ..... McNeese State University, Lake Charles, LA
Tim McGuire ..... Sam Houston State University, Huntsville, TX
Bilal Shebaro ..... St. Edward's University, Austin, TX
Jeong Yang ..... Texas AM University-San Antonio, San Antonio, TX
Ferdi Eruysal ..... Texas A&M University Central Texas, Killeen, TX
Abena Primo ..... Huston-Tillotson University, Austin, TX
Yi Liu ..... Georgia College and State University, Milledgeville, GA
Lisa Lacher ..... University of Houston-Clear Lake, Houston, TX
Mamta Yadav ..... Texas AM University-Corpus Christi, Corpus Christi, TX
Tina Johnson ..... Midwestern State University, Wichita Falls, TX
Eduardo Colmenares ..... Midwestern State University, Wichita Falls, TX
Catherine Stringfellow ..... Midwestern State University, Wichita Falls, TX
Muhammad Rahma ..... Clayton State University, Morrow, GA

Analyzing the Impact of Experiential Pedagogy in Teaching Socio-Cybersecurity: Cybersecurity Across the Curriculum*

Carlene M Buchanan Turner[1] and Claude F Turner[2]
[1]Sociology Department
[2]Computer Science Department
Norfolk State University
Norfolk, VA 23504
{cmturner,cturner}@nsu.edu

Abstract

This paper presents results from efforts to integrate cybersecurity into social science courses, for the purposes of improving the cybersecurity awareness of non-computer science students. The instruction was delivered through socio-cybersecurity modules. A quasi-experimental methodology which included pre-tests, delivery of the modules, and then post-tests was the source of the primary data. Paired t-test was used to analyze the data. Qualitative data was also collected form students' discussions, after the experiential exercises. The infusion targeted a 200-level Social Problems class across two semesters.

The paired T-Test results showed significant mean differences for two of the eight concepts taught in the Password Module, as there was a significant difference across pre- and post-test conditions for the use of special characters t(22)=-3.33, p=0.003; and for vulnerability of weak passwords t(22)=2.47, p=0.022. Additionally, one of the six concepts taught in the Phishing module demonstrated significant mean difference, as students believed that most scammers worked alone and not in organizations t(25)=3.07, p=0.005. Based on the outcome it was demonstrated

that sociology students could benefit more from experiential learning, using laboratory settings rather than a lecture-only format.

1 Introduction

This research examines the learning outcomes of non-computer science majors who were taught socio-cybersecurity modules using an experiential learning pedagogy. The analysis presented in this work is a component of a three-year project to integrate cybersecurity into the sociology and criminal justice curriculum at a minority serving institution. Socio-cybersecurity, a relatively new field, is defined as the socio-cultural aspects of cybersecurity. Within the emerging discourse there is a focus on the accompanying social problems of the phenomenon, the socio-psychological implications particularly for criminology, its role in modern bureaucracies and institutions, and the position of big data and research methodology. This article is predicated on a password module and a phishing module which were integrated in a 200-level Social Problems course.

The two hypotheses that are analyzed to measure the learning outcomes are: (1) Ho: There is no difference in the means of the Password indicators across the pre- and post-test conditions; and (2) Ho: There is no difference in the means of the Phishing indicators across the pre- and post-test conditions. The premise here is that as cybersecurity threats become an ever-present reality for social science practitioners, experiential learning can be used to re-socialize, or construct new realities about our everyday computing practices.

2 Background

2.1 Theoretical Framework

The students targeted by the project engaged in active learning. This complements the constructivist premise that hands on experience is the root of, and stimulus for, learning [4]. The curriculum development efforts analyzed in this paper are guided by a social constructivism learning theory. Vygotsky social constructivism theory of learning reinforces the value of the students' experience and their social context in the learning process [8]. Vygotsky [8] defined education as the artificial mastery of natural processes of development, so the premise of this research is that experiential pedagogy mimics the taken for granted process of learning by interacting with one's environment.

Vygotsky social constructivism of learning theory supports the pedogical thrust of the socio-cybersecurity project because the modules were created and taught with experiential techniques. The goal of the project therefore,

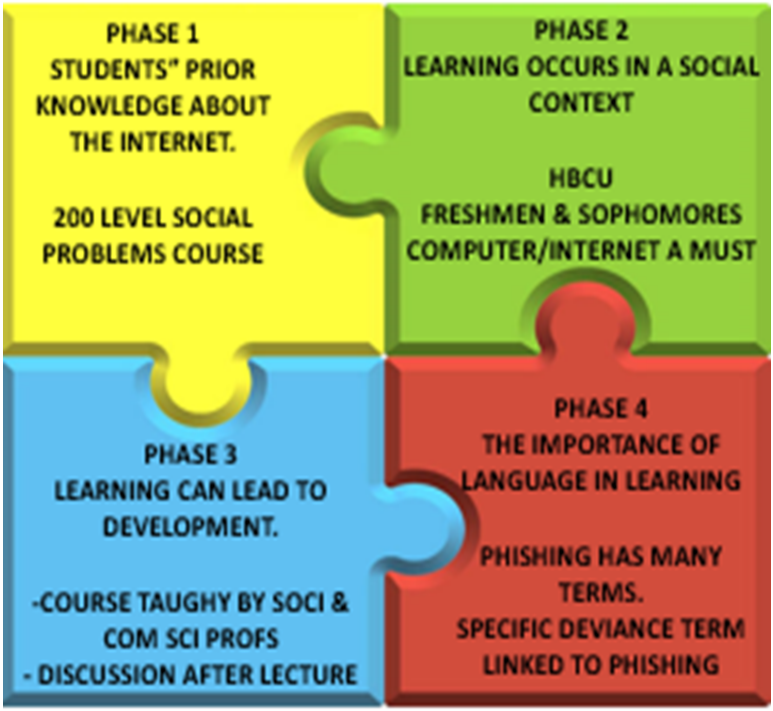


Figure 1: Constructivism Learning Theory for Socio-Cybersecurity

was to create modules that allowed students to experience cybersecurity by doing hands-on activities or engaging in discussions.

The framework for each module included: the background; a checklist; the lecture; the laboratory exercise, a discussion, and a final checklist [11]. This structure complements the Vygotsky Constructivist pedagogy which recognizes that learners encounter a lesson with a priori knowledge. The theory has four important components. First, Figure 1 points out that the Social Problems students encounter the socio-cybersecurity modules with prior knowledge about the internet. Secondly, learning occurs in a social context. The identity of the learner from the specific (their academic major) to the macro (their race and ethnicity) is thus seen as important in the learning process. Third, experiential (hands-on or active learning) can lead to development. Therefore, creating a phishing game or creating your own passphrases can lead to development. Finally, language is important in this type of pedagogy. The theoretical choice for this project is predicated on the belief that experiential learning is a pivotal

part of how human beings learn [6]. Korgen and Atkinson [5] point out in their new textbook that active earning will allow "students to do sociology through real-world activities designed to increase learning, retention, and engagement with course material."

2.2 Literature Review

The existing literature supports the use of modules to integrate new concepts into a school's or college's existing curriculum [3]. Additionally, the use of modules as instructional resources seems to occur more seamlessly in the physical and computational sciences, more-so than in the social sciences [7]. The discourse in the literature supports the practice of instructors broadening the teaching of science by enriching the syllabus with interdisciplinary modules. Gardener [3] implemented a module infusion program where she and her colleagues introduced scientific concepts to non-science majors. The results from their project demonstrated that such efforts allow for the appeal to a large general audience of students who needs a citizen's (or working) complement of science.

The cybersecurity infusion across the curriculum project is based on the development and design of security injections. The best practice is based on self-contained modules focused on specific, well-established principles (Saltzer & Schroeder, 1975; Taylor & Azadegn, 2006). As stand-alone modules, security injections are inserted seamlessly into existing courses with planned frameworks. Additionally, these new modules expose students to the relatively new practice of cybersecurity beyond computer science and information technology departments.

Borrowing from the lab model in traditional sciences, each security injection module contains introductory background, lecture/laboratory/homework assignments (with supporting video content), a security checklist, discussion questions, and a security scorecard [10]. The outline and content of these modules encourage students to engage in experiential learning strategies. Lab assignments provide opportunities for active learning [10]. Short video clips are included as they are beneficial in enhancing the learning process by increasing students' outcomes and instructor's development [2][9].

3 Methodology

A two-fold methodology was used to assess the outcome of the infusion of the Password Module and the Phishing Module into the 200-level Social Problem course, using two different samples of students. The modules were 'Creating Strong Passwords: A Simple National Security Tool' and 'Sociology of Deviance

and Phishing: A Scam to Steal Private Information.' The module was co-taught by a Computer Science professor and a Sociology professor.

3.1 The Pre- and Post-Test Surveys

The primary methodological tools that facilitated the evaluation of the module infusion process was a quasi-experiment based on the population of TTSU students that were enrolled in the targeted course. Pre and post-tests surveys were conducted via Blackboard. Each module had a unique questionnaire to capture the concepts being taught. For each module, identical questions appeared on both the pre and post-test instruments. The Password questionnaire had twelve items, with four demographic questions and eight password content questions. The eleven-item Phishing questionnaire had the same demographic questions, as well as seven phishing content questions. The pre-tests were collected; then the modules were taught in under a week; then a week later the post-tests were made available to the students.

While 26 students participated in the Password pre-test, there were two less participants in the post-test. This resulted in only 23 paired observations across the pre and post-tests conditions. The data demonstrates that approximately 70% of the class were female; just about half were freshmen; and approximately two-thirds were Social Work majors.

For the Phishing module, there were more students in the pre-test (40) than in the post-test (28). This resulted in 26 valid paired observations across the pre and post-tests conditions. Most of the students were female across the infusion conditions (pre-test 72.5%) and (post-test 78.6%). Most of the students in the class were sophomore (42.5% in the pre-test and 39.3% in the post-test). The students in the course were from two majors: Sociology (47.5% pre-test; 50% post-test) and Social Work (37.5% pre-test and 39.3% post-test).

The statistical techniques utilized in this paper were Paired T Tests as well as descriptive statistical analyses to compare the means on the relevant password items and data integrity items across the pre- and post-tests. The findings are presented in the results section.

3.2 Content Analysis of Students' Assignments

In the Password Module, the other methodological technique utilized was a review of students' assignments. Qualitative data was gathered from the experiential in-class laboratory exercises to complement the surveys from the quasi- experiment. The students worked in groups to build pass-phrases using the management rules taught in the lecture. For the phishing module, the qualitative data gathered was based on a phishing simulation game. The students

navigated several scenarios in the Anti-Phishing Phil game, and they reflected on the real-life lessons learnt from the game [1].

4 Results

4.1 Descriptive Statistics

In examining the descriptive statistics for the password content questions, it was noted that the infusion of the module into the Social problem class impacted student learning. There was a 20% jump in students who believe that passwords should be at least eight characters. There was a 20% drop in the number of the students who said that passwords should be memorable. There was about a 20% increase in students who said that passwords should have special characteristics. In the post-test condition, more students indicated that passwords should be changed once a year, rather than every six months. About the same number of students agreed that national security agencies had strong password protocols. There was a 20% increase in the opinion that weak passwords can compromise national security. There was a 14% increase in the option that USA's cybersecurity has been attacked a lot. Finally, more students were likely to say their passwords were somewhat secure, instead of saying secure.

Descriptive statistics for the phishing content questions demonstrates the module's impact on student's learning. 67.5% of the students said they knew what phishing was in the pre-test, and 92.9% said the same in the post-test. 57.5% of the respondents said it was appropriate to open an expected e-mail attachment from a 'known' person in the pre-test condition, compared to an increased percentage (64.3%) in the post-test condition. In the pre-test 52.4% of the students said they would trust an embedded link if it originated in the USA, while in the post-test 60.7% said they would not trust it whether it originated in the USA or not. Likewise, in the pre-test 87.5% said they are suspicious of stranger e-mails whether they originated in the USA or not, while 75% said the same in the post-test. In the pre-test condition 37.5% of scammers were described as too smart for their good, while in the post-condition they were equally described as too smart for their own good and as just greedy (46.4%). In the pre-test condition most students said that scammers worked in groups (52.5%), while this was reversed in the post condition to work alone (50%). In the pre-test condition 35% of the students believed they get about 3-4 suspicions e-mails monthly; while the estimate is 50% in the post-test condition.

4.2 The Quasi Experiment (Pre-& Post-Tests)

The hypothesis that drove the analysi of the password module was: Ho: There is no difference in the means of the password indicators across the pre- and post-test responses.

The paired T-test analysis covered eight individual password items as indicated by Table 1. Two significant means between pairs resulted from the Password Module analysis. First, there was a significant difference in the scores for whether it's important to use special characteristics in passwords before the module was taught (M=2.65, SD=1.03) and after the module was taught (M=3.43, SD=.728); t(22)=-3.33, p=0.003.

Table 1: T-TEST RESULTS FOR PASSWORD INDICATORS TAUGHT IN SOCIAL PROBLEMS CLASS SPRING 2017

Variables	Pretest		Posttest		95% CI for Mean Difference			Sig (2-tailed)
Outcome	M	SD	M	SD		t	df	
Password Length	2.78	.518	3.00	.302	-.477, .042	-1.74	22	.096
Memorable	1.17	.491	1.39	.583	-.561 .127	-1.31	22	.203
Special Character	2.65	1.03	3.43	.728	-1.27, -2.96	-3.33	22	.003
Time 2 Change	2.09	.949	1.78	.795	-.137, .745	1.43	22	.166
National Security	1.36	.727	1.27	.550	-.294, .476	.49	21	.628
Compromise	1.52	.790	1.26	.689	-.213, .735	1.14	22	.266
Attacked	1.57	.896	1.13	.344	.070, .800	2.47	22	.022
Everyday Password	2.26	.915	2.48	1.12	.931, .496	-.63	22	.534

N=23

Another significant mean difference was realized for whether students thought unsecure passwords have resulted in attacks on the USA. The before module condition resulted in (M=1.57, SD=.896) and after the module was taught (M=1.13, SD=.344); t(22)=2.47, p=0.022. The other six indicators did not result in any significant results.

For the Phishing Module, the hypothesis for the data integrity analysis was: Ho: There is no difference in the means of the phishing indicators across the pre- and post-test responses.

The paired T-test analysis for the Phishing Module covered seven indicators as displayed in Table 2. Only one paired indicator resulted in a significant mean difference. There was a significant difference in the scores for the opinion about whether scammers worked alone or in groups. The before module condition resulted in (M=2.19, SD=.634) and after the module was taught (M=1.81, SD=.849); t(25)=3.07, p=0.005. The other six indicators did not result in any other significant results.

Table 2: T-TEST RESULTS FOR PHISHING INDICATORS TAUGHT IN SOCIAL PROBLEMS CLASS FALL 2017

Variables	Pretest		Posttest		95% CI for Mean Difference			Sig (2-tailed)
Outcome	M	SD	M	SD		t	df	
Know Phishing	1.31	.549	1.08	.272	-.006, .468	2.00	25	.056
Careful	1.46	.582	1.35	.485	-.172, .403	.827	25	.416
Trust Origin	2.69	1.49	2.85	1.49	-.797, .489	-.493	25	.627
Trust Sender	2.77	.652	2.70	.884	-.177, .330	.625	25	.538
Scammer Character	1.85	.834	2.00	.980	-.491, .183	-.941	25	.356
Scammer Social	2.19	.634	1.81	1.37	.127, .642	3.07	25	.005
# Email	1.88	1.37	2.08	1.26	-.535, .151	-1.15	25	.259

N=26

4.3 Analysis of Students' Assignments

Students' outcomes from the piloting of the module in the Social Problems class can also be seen in the assignment they were given after the instructions. Directly after the Password module was taught, the students – divided into groups - were instructed to create their own passwords, based on self-selected pass-phrases. Five out of the seven groups successfully created passwords that conformed to the password rules that were taught in the lab. These were: choose a memorable phrase; pull the first letter of main words; include numbers; special characteristics; upper-case letters; and they should not simply be dictionary words. Table 3 presents the innovative passwords the student groups created. The surveys for the Phishing module were also supplemented by an interactive game and content analysis was used to catalog the student's written answers. Some of the common sentiments from the students were: "The game helped me to be more alert about certain sites." "The Anti-Phishing Phil games has helped me to better understand how many tricky ways that are to get caught up into the madness of phishing." "Another thing that caught my eye was a lot of banking sites and the msn website with the word verify in it as I didn't know that would be a scam site."

5 Discussion

The infusion of both the Password and Phishing Modules provided evidence for the continued integration for cybersecurity modules into Sociology courses. For the Password Module that was integrated into the Social Problems course, the increased awareness of the students was seen across the two evaluation tools presented. First, the paired t-test indicated that there was a significant

Table 3: STUDENTS' PASSWORDS CREATED FROM THE MODULE'S LAB- SOCIAL PROBLEM CLASS SPRING 2017

Password	Phrase-Based Mnemonics
(ig2pimp)	I got 2 phones in my pocket
Catnth3Hat	Cat in the hat_
Mdasd1gom	My dear aunt sally drinks 1 glass of milk
REMywl76win.sum	Remy was lcked 7 winters & 6 summers
sfs;of4A	stand for something; or fall for Anything
#ILmsw230c	I love my social work 230 class
Dt0uchMh-s	Don't Touch my hair - Solange

mean difference in students' opinion on use of special characteristics, and the importance of password in keeping the USA from being attacked. This is a simple yet practical skill that students can use in their everyday internet interactions.

Secondly, the students use of popular culture to create their pass phases and ultimately passwords were not only fun, but it demonstrated the power of experiential learning. The students readily used the password creation protocols from the lecture's checklist. It should be concluded that the concepts that the students interacted with helped them to understand the cybersecurity concepts. This reinforced the theoretical underpinning of the project [6].

Finally, the comparative descriptive statistics shows that the students' knowledge had grown from the pre-test to the post-test condition. Based on the analysis of the survey data, students demonstrated an improvement in the parameters of password creation, and they also knew that weak passwords could have national security implications.

The second Social Problems module was offered in Fall 2017. The Phishing and Deviance module also demonstrated some positive results for the infusion project. The concept being introduced, phishing linked to the deviance typology labeling was also abstract, yet there was a positive reaction from the students. Navigating through the simulated phishing environment was reported as the impetus for some of this learning. The students showed that they understood the assymmetrical nature of phishing attacks after the infusion [3].

6 Conclusion

In conclusion, the analysis of the learning outcomes from the socio-cybersecurity module infusion demonstrates that this is a efficient method to

teach social sciences students a STEM discipline. Incorporating experential learning techniques by allowing students to make connections in hands-on laboratory exercises to their everyday lives enhanced their cybersecurity awareness. However, there were some areas in the module infusion process that need further development.

A suggested improvement that can be made in the deployment of the modules based on the in-class surveys. The analysis demonstrates that students need to have the opportunity to practice the skills from the lecture. The project's education evaluator suggests that this can be done through cooperative learning.

The utilization of the social construction of learning theory [4] reinforced the value of experiential learning in teaching cybersecurity to sociology students. Moving forward, the expectation is to improve the modules based on the lessons from these early interventions and deploy them again in future academic term. In conclusion, the learning outcomes were enhanced not just by the modules' lecture, but also the hands-on laboratory exercise and in-class discussions.

7 Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1623201

References

[1] Anti-phishing phil game. `https://www.ucl.ac.uk/cert/antiphishing/`.

[2] Pedagogical benefits. `http://www.uq.edu.au/teach/video-teach-learn/ped-benefits.html`.

[3] M. Gardener. Modules and minicourses for integrated science. *The Sci. Teacher*, 40(2):31–32, 1973.

[4] P. Jarvis, J. Holford, and C. Griffin. *The Theory and Practice of Learning, 2E.* London and Sterling, Virginia, 2012.

[5] Kathleen Korgen and Maxine P. Atkinson. *Sociology in Action.* SAGE Publishers, New York, 2017.

[6] Chris Manolis, David J. Burns, Rashmi Assudani, and Ravi Chinta. Assessing experiential learning styles: A methodological reconstruction

and validation of the kolb learning style inventory. *Learning and Individual Differences*, 23:44 – 52, 2013.

[7] B.A. Nye and C.G. Thigpin. Examing the relationship between process-oriented staff development and classroom practices using integrated mathematics and science instructional modules. *Journal of Elementary Science Education*, 5(1):10–26, 1993.

[8] R. S. Prawat. Dewey and Vygotsky viewed through the rearview mirror and dimly at that. *Educational Researcher*, 31(5):16–20, 2002.

[9] N. Spence. The art of educational videos. `http://teche.ltc.mq.edu.au/art-educational-videos/`.

[10] Blair Taylor and Shiva Azadegan. Threading secure coding principles and risk analysis into the undergraduate computer science and information systems curriculum. In *Proceedings of the 3rd Annual Conference on Information Security Curriculum Development*, InfoSecCD '06, pages 24–29, New York, NY, USA, 2006. ACM.

[11] Blair Taylor and Shiva Azadegan. Teaching security through active learning. In *Proceedings of Frontiers in Education: Computer Science and Engineering*, 2007.

A Course Module on HTML5 New Features and Security Concerns*

Mounika Vanamala, Xiaohong Yuan, Macey Morgan
Computer Science Department
North Carolina A&T State university
Greensboro, NC 27411
{mvanamal,mlmorgan}@aggies.ncat.edu, xhyuan@ncat.edu

Abstract

This paper describes a course module developed to introduce some of the new features introduced to HTML5, and the security concerns related to the new features, and possible countermeasures. The HTML5 features considered in this paper include: local storage, Geolocation API, offline web, Web workers, and Web messaging. The course module includes power point slides on the topic, and a hands-on exercise demonstrating the HTML new features. This course module could be used in a class teaching Web programming to introduce HTML5 and security concepts related to it.

1 Introduction

The Hypertext Markup Language version 5 (HTML5) is the successor of HTML 4.01, XHTML 1.0 and XHTML 1.1. In HTML 4 the web development is mostly based on the serverside generated content and layout. In HTML5, new ways are introduced to store information on the client-side database for easy access. HTML5 introduced new elements and attributes, changed some elements and made some elements obsolete.

The new features introduced in HTML5 also brings potential security concerns. There will be new attack vectors for hackers to exploit. Some originate from elements of the standard itself, some from the implementations of the

standard in each browser, and some from the care that developers take in building their HTML5 code.

This paper introduces a course module that discusses how the new features introduced in HTLML5 have an impact on the security of the system [5]. The course module was developed to teach students about the new features of HTML5, the security concerns related to the new features, possible countermeasures, and lab exercises designed to demonstrate the new features. The HTML5 features considered in this paper include: local storage, Geolocation API, offline web, Web workers, and Web messaging. The course module includes power point slides on the topic, and a hands-on exercise demonstrating the HTML new features.

The rest of the paper is organized as follows. Section 2 describes the HTML5 new features, their security concerns, and possible countermeasures. Section 3 describes the hands-on lab exercises. Section 4 describes our teaching experience, and section 5 concludes the paper.

2 HTML5 New Features and Security Concerns

2.1 Local Storage

Before HTML5, application data had to be stored in cookies, which are included in every server request. HTML5 introduced Local storage, which is also known as Offline Storage, or Web Storage. It is more secure, and large amounts of data can be stored locally on the client without affecting website performance. Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server [4].

HTML5 local storage provides two objects for storing data on the client:

- window.localStorage – It stores data with no expiration date.
- window.sessionStorage – It stores data for one session (data is lost when the browser tab is closed).

The main security concern with Local Storage is that user is not aware of the data that is stored in Local Storage. The user is not able to control the access to data stored in Local Storage.

It is common to use cookies to track users visiting websites. With HTML5, Local Storage is another way to store information about a user visiting the website. The website can store user tracking information on the client's browse and correlate user sessions [7]. A third-party advertiser (or any entity capable of getting content distributed to multiple sites) could use a unique identifier stored in its local storage area to track a user across multiple sessions, building

a profile of the user's interests to allow for highly targeted advertising [6]. However, Local Storage is not deleted in all browsers if browsing history is deleted. Users trying to delete their browser cache may not know that data stored in Local Storage is not deleted [8].

Therefore, it's highly recommended not to store sensitive information in the local storage. Developers should use object *sessionStorage* when persistent storage is not needed. Users should pay attention to *"getItem"* and *"setItem"* calls within the HTML5 page. This helps detect when sensitive information is put into the local storage.

2.2 Geolocation API

Before HTML5 the user location was found using plugins such as Java Applets. In HTML5 the physical location of the user is tracked based on the GPS position. However, the position is not available unless the user permits sharing the location as it can affect the privacy of the user. the web application needs to trick the user to always accept sharing location information with this domain. The more precise the location information is, the more precise the user tracking can be.

Additionally, if the user has a user account with the web application, then the application knows which user is visiting. Every time the user accesses the web application, the user's position is tracked. Based on this, the website can create a profile of the user's movement and track the user's physical movement.

Users must be trained not to allow web applications to access the location information, and to share location information only to trusted service providers. If HTML5 Geolocation APIs are used, it is suggested to migrate the pages making Geolocation API calls to HTTPS. This way, the Geolocation APIs are used in a secure context.

2.3 Offline Web Application

HTML5 introduces the concept of Offline Web Applications. A web application can send files to the browser which are needed for working offline. The browser recognizes the offline mode and loads the data from the browser cache. To tell the browser that it should store some files for offline use, the new HTML5 attribute manifest in the tag has to be used.

The attribute manifest refers to the manifest file which defines the resources, such as HTML and CSS files, to be stored for offline use. The manifest file lists the files that should be cached and stored offline, the files that should never be cached, and the files that should be loaded in case of an error. This manifest file can be located anywhere on the server [6].

It is possible to cache the root directory of a website. Caching of HTTP as well as HTTPS pages is possible. Whether the browser requests permission for the user to store data for offline browsing, and when this cache is deleted vary from one browser to another.

The Offline application cache stays on the browser until either the server sends an update, or the user deletes the cache manually. If the "recent history" is deleted, the offline application cache of some browsers may not be deleted. These are threats to browser protection and secure caching.

Web Application information stored offline can also be used for user tracking. Web applications can include unique identifiers in the cached files, which can be used for user tracking and correlation.

2.4 Web Workers

With Web Workers, it is possible for a web application to do some processing work, like refreshing data or accessing network recourses, while the web application is still responding to the user. Web Workers are JavaScripts running in the background. They are like the Threads in other programming languages.

Web Workers do not directly introduce new vulnerabilities but make exploiting vulnerabilities easier. For example, Web Workers makes establishing and using Botnet easier to implement and are less likely to be detected by the user.Web Workers could also use excessive CPU for computation, leading to Denial of Service condition. To mitigate the security risks of Web Workers, web application developers should ensure the Web Worker scripts are not malicious, don't allow creating Web Worker scripts from user supplied input, and validate messages exchanged with a Web Worker.

2.5 Web Messaging

Web Messaging is the way for documents to separate browsing context to share the data without Document Object Model. The main problem with Web Messaging is that the content of a web page is no longer limited to content from its origin domain and the server cannot control all data sent and received by its web pages. With Web Messaging the web page may receive content of other domains without the server being involved and the data is exchanged within the browser between the Iframes. The receiving Iframe does not check the origin. The target of postMessage() is set to * because both Iframes receive input and are designed to handle input correctly. So, they are able to influence how the input is rendered in the receiving Iframe. Sensitive data may be sent to the wrong Iframe. This is a threat to the security requirement of confidentiality

3 Lab Assignment

In this lab assignment, students are given a web application "MysteryApp" in which students will solve different mysteries by answering the questions related to them. The Students can watch a video and obtain hints related to the mystery. The student will receive a score when he/she solves the mystery. The lab assignment includes five parts, which correspond to the HTML5 new features and security concerns in section 2. The five parts are explained below.

3.1 Local Storage

The main objective of this part of the lab assignment is to teach students how the Html5 local storage is built and how it is stored and retrieved even after ending the session. Local storage is built on the concept of key-pairs. Data is stored based on a named key and later it is retrieved by using the same key. The referred named key is implemented as string [9].

A simple example that shows how this works is shown below:

```
// store data record
localStorage.setItem("lastname", "XYZ");
// retrieve data
document.getElementById("result").innerHTML =
  localStorage.getItem("lastname");
localStorage.removeItem("lastname"); //removes data record
```

In the above example, a localStorage name/value pair is created with name="lastname" and value="XYZ"; afterwards the value of "lastname" is retrieved from the local storage and inserted into the element with id="result".

In this exercise, once the student starts the application he/she will find the score of playing the game on the top let corner. The score is stored locally, and students were asked to observe the score by closing and reopening the session. the students were also provided with the source code and asked to change local storage to session storage and observe if there is any difference [2].

3.2 Geolocation

This part of the assignment demonstrates to the students the use of geolocation API to request sharing of location. The geolocation API is implemented through the navigator.Geolocation object. The getCurrentPosition() method initiates an asynchronous request to detect the user's position. When the position is determined, the defined callback function is executed.

Once the student opens MysteryApp, a dialogue box will be poped up asking the student's permissions to access his/her location at top left corner of

the web page. The student will give permission to access his/her location and go to the console to see the location of the computer he/she is using to access MysteryApp.

3.3 Offline Web

The HTML 5 specification provides two methods for using the web application offline:

1. An SQL-based database API for storing data locally
2. An offline application HTTP cache for ensuring applications are available even when the user is not connected to the network

The cache manifest file is a simple text file that lists the resources the browser should cache for offline access. Resources are identified by URI. Entries listed in the cache manifest file must have the same scheme, host, and port as the cache manifest [3].

In this part of the lab assignment the offline application HTTP cache is used to demonstrate the offline web specification of HTML5. Students will disconnect their computers from the network and observe that the application can function the same as online. The students will then disable this feature by deleting the cache manifest file and observe that the students could no longer use the application offline after disconnecting the computer from the network [1].

3.4 Web Workers

The worker thread can perform tasks without interfering with the user interface. The Worker interface of the Web Workers API represents a background task that can be easily created and can send messages back to its creator. Creating a worker is as simple as calling the Worker() constructor and specifying a script to be run in the worker thread.

In this part of the lab assignment, a Web worker is implemented which takes a prime number returned by a function and assign to the "result" variable. The web worker function searches through numbers from until it finds a prime number, and then returns the prime number with a post message call. Students were asked to click on "Find Highest Prime Number" button to display the highest prime number. The number keeps changing to larger one. It is a JavaScript that runs in the background, independent of other scripts. Without affecting the performance of the page, the user can continue to do whatever he wants: clicking, selecting things, etc., while the web worker runs in the background. The students were asked to examine the code and explain how the web worker function is implemented.

3.5 Web Messaging

The Window.postMessage(message, targetOrigin) method safely enables cross-origin communication when called, this method causes a MessageEvent to be dispatched to the target document. The parameter targetOrigin specifies the origin of the target document.

The web messaging is implemented in Demoapp.html and in Demo.html. demoapp.html gets the message and checks that it is not blank, then sends the message to demo.html. demo.html has an event listener that listens for the message to be received. It then checks that it is coming from the correct domain, and there are no illegal characters. If so, it posts the message in the element with id "messages".

Students were asked to click on the button "Cross Domain Web Messaging" and enter a message in the text box and click on send and the message will be received. They were asked to review the source code in Demoapp.html and Demo.html to explain how web messaging was implemented.

4 Teaching Experience

Course Module was taught in Spring 2017 and received positive feedback from students. Initially students were provided with a PowerPoint presentation and then they were given a lab assignment along with a lab manual document and had one week to submit. There were 21 students, and everyone submitted the assignment with the average grade of 97 and a feedback survey was conducted, 15 Students from a class of 21 students participated in this survey.

The learning objectives were met	60% Strongly Agree, 33.3% Agree
Powerpoint presentation was useful to help you understand the material	60% Strongly Agree, 33.3% Agree
Hands-on lab exercises helped you better understand the material compared to having only Powerpoint presentation describing the concepts.	74% Strongly Agree, 6% Agree
You enjoyed doing the lab exercises	46.67% Strongly Agree, 33.33% Agree
The instructions in the lab manual were clear	46.67% Strongly Agree, 33.33% Agree

Table 1: Question Response

Students commented that this course module was informative, and it was useful to learn about the security risks. From the survey we can conclude that

most students were able to complete the lab assignment without any hurdle and they were also interested in learning some more modules based on HTML5.

5 Conclusion

This paper describes a course module for learning the new features of HTML5 and their security concerns. The new features of local storage, Geolocation API, offline web, Web workers, and Web messaging are introduced. Security concerns related to these features, and possible countermeasures are discussed. A hands-on assignment for demonstrating these concepts is also described. The assignment was used in a class. Most of the students were able to finish the assignments and they gave positive feedback on the course module. This course module could be used in a class teaching Web programming to introduce HTML5 and security concepts related to it.

References

[1] W3C HTML5: Offline web applications, 2011. `http://www.w3.org/TR/2011/WD-html5-20110525/offline.html`.

[2] W3C web storage., 2013. `http://www.w3.org/TR/webstorage/`.

[3] Statcounter globalstats. offline web applications, 2014. `http://caniuse.com/#search=offline\%20web\%20applications`.

[4] Statcounter globalstats. web storage – name/value pairs, 2014. `http://caniuse.com/#search=Web\%20Storage`.

[5] J. Jenkov. Html5 local storage. `http://tutorials.jenkov.com/html5/local-storage.html`.

[6] Pilgrim M. Dive in: Let's take this offline, 2013. `http://diveintohtml5.info/offline.htm`.

[7] S. Z. Naseem and F. Majeed. Extending HTML5 local storage to save more data; efficiently and in more structured way. In *Eighth International Conference on Digital Information Management (ICDIM 2013)*, pages 337–340, Sep. 2013.

[8] M. Pilgrim. The past, present & future of local storage for web applications., 2013. `http://diveintohtml5.info/storage.html`.

[9] William West and S. Monisha Pulimood. Analysis of privacy and security in HTML5 web storage. *J. Comput. Sci. Coll.*, 27(3):80–87, January 2012.

A Study of Evolutionary Algorithms for the Degree-Constrained Minimum Spanning Tree Problem*

Anthony Bloch[1] *and Robert Owens*[2]
[1]*Computer Engineering*
[2]*Computer Science Department*
St. Cloud State University
St. Cloud, MN 56301
{ajbloch,rtowens}@stcloudstate.edu

Abstract

The degree constrained minimum spanning tree problem (DCMSTP) is an NP-hard problem which seeks to find a minimum spanning tree on a complete graph, whose degree does not exceed a given maximum. A number of different algorithms are tested to find solutions represented by one of two different data structures, Prüfer strings and adjacency lists. Two versions of Prüfer string decoding are used, the standard Prüfer decoding and the dandelion decoding. Four mutation operators, four crossover operators, and three operators which combine mutation and crossover are implemented in evolutionary algorithms (EAs) along with a greedy algorithm, and the fitnesses of the solutions found by the algorithms are compared. The greedy algorithm produces the best solutions, but the EAs perform almost as well with a smaller number of vertices. For a larger number of vertices, the greedy algorithm performs better than the mutation EAs and significantly better than the crossover EAs. Dandelion decoding for Prüfer strings results in better solutions than normal Prüfer decoding, especially as the number of vertices increases. Some of the input parameters such as staleness, population size, and number of runs are examined and demonstrate an improvement in the solutions found by the EAs as they become larger, with the largest improvement resulting from an increase in staleness.

1 Introduction

A combinatorial optimization problem can be complex to solve, and, as the problem scales in size, the time required to find the best answer can become impractical. The need to solve such problems has inspired the development of evolutionary algorithms (EAs), which strive to find a good solution in a reasonable amount of time to a problem that would otherwise take too long to solve to an optimal solution. EAs simulate ideas inspired by biology, such as mutation and survival of the fittest, to implement algorithms that can produce good solutions to NP-hard problems. Put very simply, an EA generates a population of chromosomes, which are representations of possible solutions, and then operates on those chromosomes to form a new generation of chromosomes. The process is repeated on the newest generation, forming many generations of chromosomes, and, if the operators used are chosen well, the fitness of the solutions that the chromosomes represent should improve as more generations are run, resulting in a good solution in the end.

Two popular operators are crossover and mutation. Crossover selects parents from the current generation and, drawing from elements of both the parent chromosomes, generates an offspring chromosome for the next generation. Mutation chooses a single chromosome from the current generation and makes a small change to generate an offspring chromosome for the next generation.

The idea of survival of the fittest is often used to select parents from the current generation that will be operated on, choosing chromosomes with better fitness to be parents. This typically results in a next generation that has better average fitness than the previous generation and can often lead the way to generating a chromosome with a new best fitness overall. These are the main ideas driving EAs, but how they are implemented for a given problem can vary, impacting the effectiveness of an EA.

The degree-constrained minimum spanning tree problem is an NP-hard combinatorial problem.[3] A variety of EAs and a greedy algorithm will be implemented to solve this problem. The solutions provided by the EAs will be analyzed to determine the best one(s). Some of the aspects of the EAs that will be studied include the operators used to create future generations, the data structures of the chromosomes, and input parameters such as staleness, populations size, and number of runs of each algorithm.

2 The Problem

The degree-constrained minimum spanning tree problem (DCMSTP) seeks to find a minimum spanning tree (MST) on a complete graph. The degree of the MST should not exceed a given maximum, while its weight should be minimized. DCMSTP has been shown to be NP-hard.[3] Thus, as the number of vertices in the complete graph grows, the time to find the best answer increases, making DCMSTP a good candidate problem for EAs to solve.

3 Representations of the Solutions

The first element of an EA to consider is what a chromosome looks like. A chromosome represent a potential solution, and it must be possible to evaluate a chromosome for its fitness, i.e. how well it solves the problem. For DCMSTP, chromosomes represent spanning trees (STs) on the base graph. Two representations of STs are implemented in the EAs described below. The first chromosome is an adjacency list which is an array of lists, one for each vertex on the graph. Each list holds an array of pointers to the other vertices it is connected to as well as an array of integers corresponding to the weights of the connected edges. The index of a pointer to a vertex is the same as the index of the edge's weight that connects that vertex. The fitness of such a chromosome is found by adding up the weights of all the edges in the ST. This is the structure used to hold the base graph on which the EAs run.

The second chromosome is a Prüfer string. For a graph of n vertices, a Prüfer string, consisting of n-2 integers, can represent an ST for those n vertices. Where the number corresponding to a given vertex appears in the Prüfer string x times, the degree of that vertex in the ST is x+1. When the fitness of the Prüfer string is being evaluated, it can be decoded into a ST referencing the original complete graph for edges' weights. For example, if a Prüfer string is 12141, it represents a tree with vertices 0 through 6. Two methods can be used to decode the Prüfer string: normal decoding (ND) and dandelion decoding (DD).[1] The corresponding trees are shown below (ND on the left and DD on the right). In the implementations of the EAs,

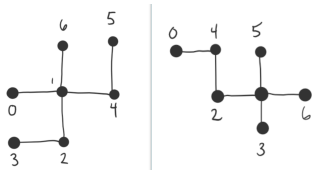


Figure 1: Normal Prüfer Decoding (Left) and Dandelion Prüfer Decoding (Right) for Prüfer String 12141

a Prüfer string is an array of integers to represent the string itself and a separate integer representing the fitness of the Prüfer string which needs to be calculated only once when the Prüfer string is generated, not each time it is referenced.

4 Non-Evolutionary Heuristic

A greedy algorithm is implemented to compare against the EAs. A greedy algorithm builds its solution one step at a time, making the most beneficial choice it can at each step, ignoring what happens in the other steps. To solve the DCMSTP, an empty ST is initialized that will hold the solution. The graph is searched for the shortest edge which is added to the ST. If there is more than one shortest edge, one is selected randomly to be added. After the first edge is added, the graph is then searched for

the shortest edge connecting a vertex already on the ST and a vertex not already in the ST. The ST is checked to determine if adding the selected edge would create a cycle or violate the degree constraint. If either of these conditions are met, the graph is searched for the next shortest edge. Once a valid edge is found, it is added to the ST. The graph is searched in this way for a shortest edge to add to the ST n-1 times where n is the number of vertices in the base graph. This forms an ST of n-1 edges.

5 EAs: General Structure

The EAs begin by generating an initial population of a number of chromosomes equal to the population size. Using simple tournament selection to choose parents and crossover and/or mutation operators, new generations are formed. Each time a new chromosome is generated, its fitness is compared to the best overall fitness which is updated whenever a new best fitness is found. The best fitness is carried over to each new generation so that the best fitness of any generation never decreases. The algorithm ends when the fitness of new generation becomes stale (i.e. when the fitness of several consecutive generations does not improve). Staleness refers to the number of consecutive generations allowed with no improvement in the best fitness before terminating the current run of the algorithm. Each EA is typically run thirty times (with a new, random initial population each run), and the best fitness across those runs is reported as the solution of that EA.

6 EAs

6.1 Crossover Operators

Crossover is one of the two primary operators used in genetic algorithms. Crossover creates an offspring chromosome for the next generation using elements from two parent chromosomes. Four crossover operators are implemented in nine crossover based EAs.

The first crossover operator used for Prüfer strings, called "similar-alternating" (P SA), examines the parent Prüfer strings element by element (i.e. integer by integer through the Prüfer string), copying any element that is similar between the parents to the corresponding element of the offspring. Next, the remaining elements are copied from one parent or the other, alternating which parent is referenced from one element to the next. If adding the given element from the chosen parent would violate the degree constraint, a valid random number is assigned to that element so that the degree constraint is met. Example with max degree of 4: 1134565 (parent 1) + 6135662 (parent 2) = 1135562 (offspring).

The second crossover operator used for both types of chromosomes, called "similar-random" (P SR for Prüfer string and AL SR for adjacency list) is fairly similar to SA crossover. It copies the elements or edges that are similar in the parent Prüfer string or adjacency list to the offspring but then, for the remaining elements or edges, assigns a valid random number or edge while still observing the degree constraint.

Example with max degree of 4: 1134565 (parent 1) + 6135662 (parent 2) = 5136263 (offspring).

The third crossover operator used for Prüfer strings, called "similar-random, copy fifty percent" (P SRCF), is similar to the second, but, after creating a new offspring, one of the parent chromosomes is copied over as a whole to the new generation. Thus there are only half as many new chromosomes each generation, while better chromosomes of a given generation tend to be preserved.

The final crossover operator used for Prüfer strings, called "n-point" (P NP), divides the offspring chromosome into n sections where n is the number of parents it has. For each section, the corresponding elements of a given parent are copied. If the degree constraint would be violated, a valid random number meeting the constraint is assigned instead of copying the parent. Example with max degree of 4: 1134565 (parent 1) + 6135662 (parent 2) = 1134662 (offspring).

DD and ND are used on separate instances of algorithms using these four operators and Prüfer strings to show which decoder is better. These combined with the AL SR crossover make nine crossover based EAs.

6.2 Mutation Operators

Seven EAs using mutation are also implemented. The mutation operator involves selecting a chromosome from the parent population and copying its elements to the offspring chromosome, while making a small change to its elements.[2]

The adjacency list structure is used in some of the implementations of the mutation EA. For this type of chromosome, mutation involves copying the ST of the parent to the offspring, removing a random edge from the offspring's ST, and then adding a random edge so that the degree constraint is met, no cycles are created, and a connected ST is formed that includes all the vertices from the base graph. Three operators are used to generate similar EAs that differ only in the number of edges that are changed: a single edge (AL 1), three edges (AL 3), and five edges (AL 5).

Prüfer strings are used for chromosomes in combination with two mutation operators which copy the elements of the parent Prüfer string to the offspring. For the first of these mutation EAs, called single-random (P SR), one random element of the offspring Prüfer string is changed to a valid random number while observing the degree constraint. For the second, called single-parent (P SP), another parent chromosome is selected, and a single random element in the offspring is changed to the corresponding element in the second parent. While this method involves pulling information from from two chromosomes in the parent generation, it is categorized under mutation since it only references one parent for a single element and the other parent for the rest of the elements in the Prüfer string. P SR and P SP mutation based EAs are implemented using both ND and DD.

6.3 Mixed Operators

One more method that is implemented for the adjacency lists is a mixed version of mutation and crossover. This version uses a weighted percent to determine if

a chromosome is generated using crossover or mutation. This percent is changed each generation based on what the best chromosome's method was. Thus, if the best chromosome was made using mutation, one percent is added to the chance that future chromosomes will be made using mutation, and one percent is subtracted from the chance for crossover being used. There is also a limit of a ninety-ten split so that a method is never eliminated, meaning that there should always be some chromosomes in each generation that use either mutation or crossover. There are three variations of this mixed operator (AL 1, AL 3, and AL 5) based on the number of edges mutated by the mutation operator. AL SR crossover is used for the crossover operator.

7 Testing Procedure

For each of the operators described above, tests are run. Each test is run on a complete graph whose edges' weights are generated randomly ranging from one to fifty. There are three different levels of vertices tested: twenty with max degree five, fifty with max degree ten, and a hundred with max degree twenty five. Then at each level, a normal test case is done where the staleness and the population size are equal to the number of vertices in the graph, and the number of runs of each individual algorithm is thirty. Six more test cases are done at each level with either half or double either staleness, population size, or number of runs. Finally, a test case is run on fifty vertices with staleness, population size, and runs quadruple the normal value to see better the affect of adjusting these parameters.

8 Results and Analysis

To analyze the results, the best solutions for each individual algorithm is divided by the best solution across all algorithms for that given test case. This gives a ratio that expresses how well each algorithm does in relation to the best solution that was achieved. These ratios are averaged for the test cases run multiple times (up to three times), and the average is analyzed. In all test cases except an instance of twenty vertices graph test case, the greedy algorithm produces the best solution. Overall, the best performing mutation operator based on the lowest achieved ratio is DD P SR mutation, beating the other mutations 68% of the time in the given test cases. For the other 32% where DD P SR is not the best, DD P SP is the best. Comparing AL 1 versus AL 3 versus AL 5 mutation, all three typically produce similar results. AL 1 is the best in 46%, AL 3 is the best in 36%, and AL 5 is best in 18% of the test cases. None of these did noticeably better than the others as the number of vertices change. As the number of vertices increases, the ratio of the fitness of the solutions produced by the mutation EAs to the solution of the greedy algorithm increases but remains for the most part within a factor of two of the greedy algorithm, with the exception of ND P SR and ND P SP which remain within a factor of six for the given test cases. The average time per generation for the Prüfer string mutations is significantly longer than the average time per generation for either of the adjacency

			Mutation							Crossover								Mix			Greedy
Variable	Adjustment	Vertices	AL 1	AL 3	AL 5	ND P SR	ND P SP	DD P SR	DD P SP	AL SR	ND P SA	ND P SR	ND P SRCF	ND P NP	DD P SA	DD P SR	DD P SRCF	DD P NP	AL 1	AL 3	AL 5	Greedy
Staleness	Half	20	1.845	1.751	1.797	1.469	1.802	1.203	1.403	3.736	2.731	3.509	3.757	3.824	2.346	2.562	3.329	3.649	2.453	2.734	2.633	1.000
Staleness	Normal	20	1.378	1.241	1.377	1.487	1.705	1.074	1.170	3.534	2.768	3.055	3.764	3.482	2.375	1.853	3.412	3.669	1.923	2.020	1.958	1.000
Staleness	Double	20	1.152	1.182	1.191	1.398	1.601	1.041	1.136	3.977	2.915	2.381	4.007	3.877	2.343	1.455	4.135	3.743	1.434	1.330	1.403	1.007
Staleness	Half	50	2.700	2.767	2.965	3.857	3.552	1.548	1.514	11.111	6.850	10.814	11.129	10.948	4.697	10.722	10.797	10.012	4.561	4.675	4.495	1.000
Staleness	Normal	50	1.734	1.900	1.960	2.958	3.234	1.228	1.289	10.509	7.119	9.796	11.027	10.401	4.635	9.741	11.109	10.217	2.934	3.029	2.825	1.000
Staleness	Double	50	1.388	1.399	1.403	2.591	2.574	1.151	1.120	9.492	6.089	8.808	9.798	9.190	4.238	8.794	9.559	9.568	1.897	1.873	1.730	1.000
Staleness	Half	100	2.707	2.836	2.776	5.310	5.138	1.517	1.750	16.871	9.491	16.250	16.802	16.293	6.017	16.431	17.293	15.940	4.698	5.233	4.241	1.000
Staleness	Normal	100	2.008	1.690	1.730	4.738	4.563	1.389	1.317	15.516	8.611	15.325	15.929	15.365	5.603	14.698	16.111	15.635	2.881	3.024	2.968	1.000
Staleness	Double	100	1.411	1.484	1.476	3.653	4.194	1.202	1.177	15.460	7.976	14.952	16.113	14.903	5.984	15.121	16.355	15.234	1.992	1.855	1.581	1.000
Pop Size	Half	20	1.683	1.829	1.921	1.655	1.922	1.204	1.293	5.302	4.514	2.852	4.977	4.945	4.125	2.420	4.775	4.743	2.302	2.605	2.562	1.000
Pop Size	Normal	20	1.378	1.241	1.377	1.487	1.705	1.074	1.170	3.534	2.768	3.055	3.764	3.482	2.375	1.853	3.412	3.669	1.923	2.020	1.958	1.000
Pop Size	Double	20	1.356	1.288	1.356	1.344	1.741	1.038	1.179	4.056	2.963	3.354	4.230	3.988	2.251	1.694	4.055	3.800	2.106	1.774	2.116	1.000
Pop Size	Half	50	1.887	1.832	1.850	2.619	3.399	1.310	1.354	9.801	7.559	9.482	10.316	9.580	5.851	8.655	10.047	9.542	2.506	2.712	2.568	1.000
Pop Size	Normal	50	1.734	1.900	1.960	2.958	3.234	1.228	1.289	10.509	7.119	9.796	11.027	10.401	4.635	9.741	11.109	10.217	2.934	3.029	2.825	1.000
Pop Size	Double	50	1.774	1.765	1.760	2.389	2.600	1.163	1.257	9.538	5.939	9.195	10.066	9.325	4.111	9.272	10.432	9.306	2.698	2.810	2.708	1.000
Pop Size	Half	100	1.850	1.903	1.982	4.566	5.708	1.354	1.212	16.522	11.938	16.690	17.584	17.345	9.230	16.726	18.292	17.283	2.726	2.876	2.885	1.000
Pop Size	Normal	100	2.008	1.690	1.730	4.738	4.563	1.389	1.317	15.516	8.611	15.325	15.929	15.365	5.603	14.698	16.111	15.635	2.881	3.024	2.968	1.000
Pop Size	Double	100	1.881	2.085	1.763	4.000	4.525	1.169	1.763	15.898	8.636	15.017	15.975	14.983	5.466	14.958	16.720	15.703	2.737	2.576	2.500	1.000
Runs	Half	20	1.535	1.530	1.438	1.693	2.220	1.149	1.224	4.562	3.445	3.930	4.529	4.900	2.760	2.041	4.653	4.469	2.429	2.187	2.395	1.000
Runs	Normal	20	1.378	1.241	1.377	1.487	1.705	1.074	1.170	3.534	2.768	3.055	3.764	3.482	2.375	1.853	3.412	3.669	1.923	2.020	1.958	1.000
Runs	Double	20	1.341	1.447	1.464	1.589	1.572	1.121	1.210	3.813	3.024	3.216	4.092	3.788	2.447	1.772	3.718	3.558	1.869	1.847	2.119	1.000
Runs	Half	50	2.034	2.125	1.934	3.151	3.472	1.309	1.529	10.560	6.963	10.030	11.115	10.407	4.513	9.897	11.295	10.392	3.221	3.169	2.754	1.000
Runs	Normal	50	1.734	1.900	1.960	2.958	3.234	1.228	1.289	10.509	7.119	9.796	11.027	10.401	4.635	9.741	11.109	10.217	2.934	3.029	2.825	1.000
Runs	Double	50	1.789	1.710	1.776	2.760	3.026	1.234	1.202	9.615	6.193	9.349	10.365	9.625	4.510	9.388	10.125	9.464	2.839	2.760	2.687	1.000
Runs	Half	100	1.931	1.793	1.853	4.595	3.500	1.448	2.164	16.017	9.526	16.586	17.440	16.052	6.172	15.328	17.647	16.474	2.828	2.871	2.974	1.000
Runs	Normal	100	2.008	1.690	1.730	4.738	4.563	1.389	1.317	15.516	8.611	15.325	15.929	15.365	5.603	14.698	16.111	15.635	2.881	3.024	2.968	1.000
Runs	Double	100	1.964	1.830	1.839	4.357	4.241	1.438	1.125	17.107	8.830	16.188	16.938	16.750	5.688	16.723	17.679	16.643	2.973	2.661	2.929	1.000
Staleness, Pop Size, and Runs	Quadruple	50	1.145	1.160	1.158	2.117	2.196	1.052	1.079	8.618	5.673	8.449	9.455	8.941	3.456	8.067	9.400	8.850	1.314	1.328	1.372	1.000

Figure 2: Ratios of Algorithm Results vs Best Result of All Algorithms

list mutation EAs, so the best mutation solutions require much more time for graphs with a large number of vertices.

Of the four crossover operators, DD P SA produces the best solution for higher numbers of vertices (at least fifty), always beating the other crossover operators. When solving for a solution to a smaller graph of twenty vertices, the crossover operators perform relatively equally, with DD P SA winning 14% of the cases and DD P SR winning the other 86%. As the number of vertices increases, AL SR, ND P SR, ND P SRCF, ND P NP, DD P SR, DD P SRCF, and DD P NP crossovers perform much worse than ND P SA and DD P SA.

For the debate of crossover versus mutation, these instances show that, for DCMSTP, mutation is better. The data collected shows that Prüfer strings make for slightly better chromosomes than adjacency lists for mutation, but neither perform very well with crossover. An explanation could be that the crossover operators modify the STs too much, often changing many edges. Thus, while the parents may be good solutions, the child ST can be so different that it is less likely to be better. Another potential reason that the crossover operators are worse than the others is that, in a Prüfer string crossover, copying an element of the parent strings does not necessarily mean that an edge is copied to the child from the parents since the rest of the Prüfer string affects what the ST looks like. Changing more than one element from parent to child can change the ST greatly.

For the mutation operators that use Prüfer strings, DD produces better results than ND. This difference is magnified as the number of vertices increases, where ND produces solutions with fitnesses up to almost five times greater than those produces by DD of the same operator. For crossover, the only noticeable difference between ND and DD is seen in the P SA operator where fitnesses from DD are about two thirds that of ND. The other crossover operators do not show a tendency that favors ND or DD.

Increasing the staleness, the population size, and the number of runs allow the EAs to get closer to the fitness of the solution obtained by the greedy algorithm. Increasing staleness typically improves the results of the EA more than population size or number of runs.

9 Conclusion

EAs are implemented to solve DCMSTP for a good solution. EAs with the mutation operator performs better than those with the crossover operator. DD P SR mutation produces the best solutions of the mutation operators, while DD P SA crossover performs best of the crossover operators. Some of the EAs produce good solutions, but the greedy algorithm beats out the EAs in all but one of the test cases. Dandelion decoding produces much better results for mutation operators than does normal Prüfer decoding, while, for crossover operators, dandelion decoding is only noticeably better than normal Prüfer decoding for similar alternating crossover. DCMSTP seems to lend itself to mutation based EAs and greedy algorithm solutions for general heuristic solutions, while crossover does not seem to be a powerful operator to solve

DCMSTP. Time permitting, increasing the staleness, the population size, and the number of runs tend to improve the fitness of the solutions produced by the EAs, but, of these three, increasing staleness gives the best improvements.

Acknowledgement

We would like to thank Dr. Bryant Julstrom from whom we learned many of the concepts discussed in this article in his course on evolutionary computation.

References

[1] Tim Paulden Evan Thompson and David K. Smith. The dandelion code: A new coding of spanning trees for genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 11(1), 2007.

[2] Nacira Ghoualmi-Zine and Rachid Mahmoudi. Crossover and mutation based cloning parent for degree constrained minimum spanning tree problem (d-mstp). *2010 Second International Conference on Engineering System Management and Applications*, 2010.

[3] Sheng Lin Guangping Xu Kai Shi, Qingfeng Song and Zhanxu Cao. An improved genetic algorithm for degree constrained minimum spanning trees. *28th Chinese Control and Decision Conference (CCDC)*, 2016.

Handwritten Digits Recognition Using Convolution Neural Networks*

Sam Pratt, Ana Ochoa, Mamta Yadav, Alaa Sheta, Mahmoud Eldefrawy
Department of Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX 78412, USA
{mamta.yadav alaa.sheta}@tamucc.edu

Abstract

Convolution Neural Networks (CNNs) have been successfully used to solve variety of problems in computer vision and pattern recognition applications. In this paper, we explore the use of CNN to provide a model for handwritten digits recognition (HCCR). Various CNN architecture were explored with various data size to develop the minimum required data size that can produce optimal performance. We utilized the MNIST database of handwritten digits. The recognition results are promising.

1 Introduction

Automatic handwriting recognition has a diversity of applications. It is considered as one method of communication between man and machine. Many methods were provided to handle this problem with various accuracy [11, 8]. The recognition method performance always depends on many attributes such as the size of the digit, the writing style, and the rate of recognition. One of the main challenges of handwritten digit's recognition is the inconsistency of the person handwriting style (i.e., width and shape), the type of device that collects the handwriting such as tablet or papers. Therefore, there is a need to have a system that can automatically recognize handwriting patterns

with a high recognition rate. In the past, handwritten digit recognition with a Back-propagation Neural Network was explored [4]. A hybrid optimization approach based evolution strategy and gradient descent method of Bayesian classifiers were explored in [2]. Many research results were reported for recognizing handwriting for Chinese [6, 3], Arabic [1, 9] and Japanese [7] characters. The recognition of handwritten Japanese characters using CNN was recently reported [10]. By using a deep learning technique we can reduce the number of classification errors in transcribing these handwritten digits and reduce the time taken to complete this task.

2 Handwritten Digits Database

In this research, we utilized the MNIST database of handwritten digits, available at [5]. The MNIST database has 60,000 images assigned as training examples, and another 10,000 examples assigned as testing examples. The dataset was developed as part of various scanned document dataset provided by the National Institute of Standards and Technology (NIST). Thus it was named as the Modified NIST or MNIST dataset. The database has images of size 28×28 pixel square of 784 pixels per image. This data set is used as a benchmark for various application models. The digits have been size-normalized and centered in a fixed-size image. These images are pre-normalized to 20×20 pixels and then overlay to a 28×28 pixel field to prevent any edges being to close to the end of the field allowing for a complete picture of the digit in question. This dataset was created by 500 different users to diversify the output of the data set and give it a more realistic picture of the real world uses it may face. In Figure 1 we show a sample of the training data set.

3 CNN Architecture

CNN are a special type of Artificial Neural Network (ANN). A CNN is made up of five base layers that include an input layer, a convolution layer, a pooling layer, a fully connected layer, and an output layer. CNN operates in two main processing stages: a feature learning stage and a classification stage. Each stage consists of one or more layers. The feature learning stage is implemented by combining two types of layers (i.e., Convolution layers and pooling layers). This stage provides the most significant features extracted from the training example. These features are fed to a fully connected ANN layer.

The proposed input layer will receive an image with the handwritten digits. To ensure that no subset is incomplete, in this layer padding is added to the image. This enables the ability of the subsets to fill in any missing data

Figure 1: Sample example of the MNIST data set examples.
Image from Yann LeCun, Corinna Cortes, Christopher J.C. Burges, *http://yann.lecun.com/exdb/mnist/* (MNIST Dataset Sample, 1999)

points with zeros. The first CNN layer is typically a convolution layer. The convolution layer uses a set of the filter to slid over the input images. The results of each subset are then mapped to one single point. These calculations are repeated for the entire image. Once this process is done the output is sent to a pooling layer. The pooling layer is used to make the calculations of the convolution layer easier by reducing its size. In our module, we are using max-pooling to reduce the image size by half. The pooling layer consists of filters of sizes 2×2. It takes a subset of 4 pixels and deduces that to one pixel that holds the highest value of that subset. That final decision is implemented in the fully connected layer. This layer takes the output of the last pooling layer as an input. The output layer shall have a vector of 10 neurons, one for each possible digit outcome.

3.1 CNN Simulation

Data is an integral part of the learning process. It provides the main guidance for conversion and building an accurate CNN. In our case study, we created the training as random samples of 30,000 examples out of the 60,000 examples provided by the MNIST database as training examples. And, another 10,000 examples for testing purposes. The proposed CNN will have gray-scale images of size 28×28 as inputs. We trained the CNN for 14040 iterations shown in Figure 3. The prediction curves show the training progress where the accuracy increased near iteration 8,000 and became stable later. The computed confusion matrix, in this case, is shown in Figure 4. It is clear that the overall

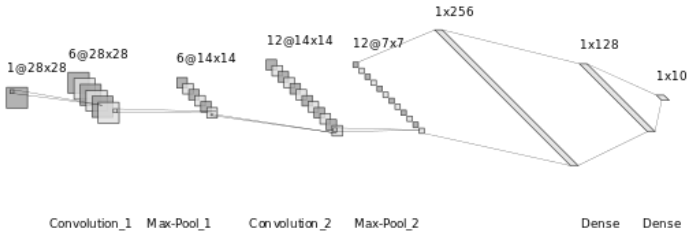


Figure 2: Proposed CNN Architecture

recognition accuracy is over 98%. The confusion matrix indicates that a representative small data training set can achieve accurate results after enough iterations.

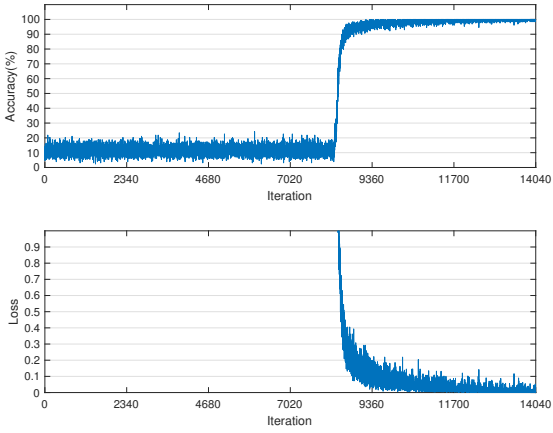


Figure 3: Training and Loss using 30000 examples

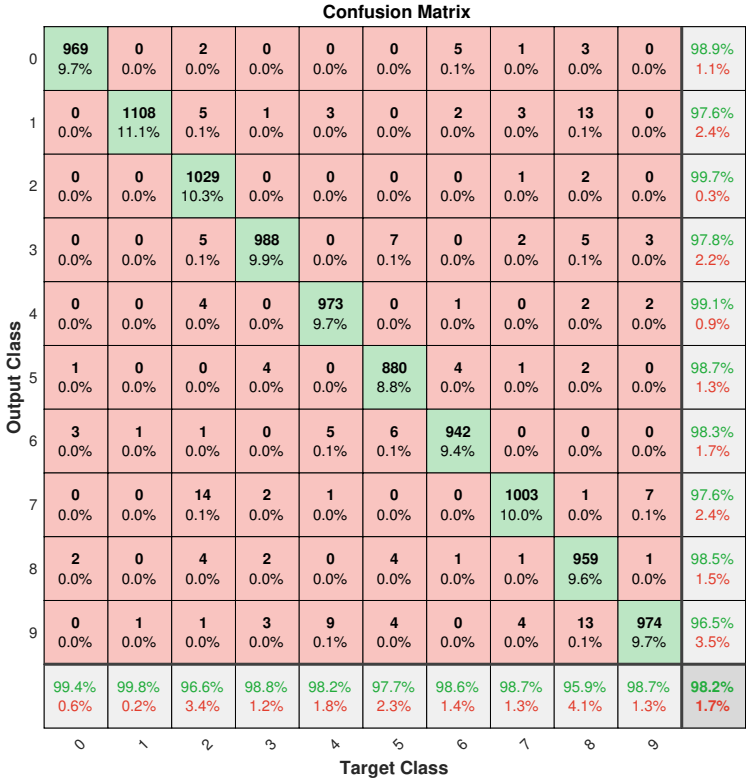


Figure 4: Confusion Matrix using 30000 examples

4 Conclusion and Future Work

In this paper, we explored our initial ideas of using CNN to recognize various handwriting digits. We utilized the MNIST database of handwritten digits. Various scenarios were adopted with a different number of digit images. We discovered that the developed CNN have a steep learning curve. While our network may achieve very high accuracy, it is not perfect, and there are ways that it could be improved. Finally, we would like also to explore the performance of our model in identifying not only English digits, but also Chinese and Arabic digits.

References

[1] Badr Al-Badr and Sabri A. Mahmoud. Survey and bibliography of arabic optical text recognition. *Signal Processing*, 41(1):49 – 77, 1995.

[2] Xuefeng Chen, Xiabi Liu, and Yunde Jia. Combining evolution strategy and gradient descent method for discriminative learning of bayesian classifiers. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, GECCO '09, pages 507–514, New York, NY, USA, 2009. ACM.

[3] Z. Chen, C-W. Lee, and R-H. Cheng. Handwritten chinese character analysis and preclassification using stroke structural sequence. In *Proceedings of the International Conference on Pattern Recognition (ICPR '96) Volume III-Volume 7276 - Volume 7276*, ICPR '96, pages 89–, Washington, DC, USA, 1996. IEEE Computer Society.

[4] Y. Le Cun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson. Advances in neural information processing systems 2. chapter Handwritten Digit Recognition with a Back-propagation Network, pages 396–404. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[5] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. `http://yann.lecun.com/exdb/mnist/`.

[6] Cheng-Lin Liu, Stefan Jaeger, and Masaki Nakagawa. Online recognition of chinese characters: The state-of-the-art. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):198–213, January 2004.

[7] Masaki Nakagawa, Junko Tokuno, Bilan Zhu, Motoki Onuma, Hideto Oda, and Akihito Kitadai. Recent results of online japanese handwriting recognition and its applications. In David Doermann and Stefan Jaeger, editors, *Arabic and Chinese Handwriting Recognition*, pages 170–195, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[8] Jan Richarz, Szilard Vajda, Rene Grzeszick, and Gernot A. Fink. Semi-supervised learning for character recognition in historical archive documents. *Pattern Recogn.*, 47(3):1011–1020, March 2014.

[9] Maad Shatnawi and Sherief Abdallah. Improving handwritten arabic character recognition by modeling human handwriting distortions. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 15(1):3:1–3:12, November 2015.

[10] Charlie Tsai. Recognizing handwritten japanese characters using deep convolutional neural networks. 2016.

[11] V. Vijaya Kumar, A. Srikrishna, B. Raveendra Babu, and M. Radhika Mani. Classification and recognition of handwritten digits by using mathematical morphology. *Sadhana*, 35(4):419–426, Aug 2010.

Selection of WSNs Inter-Cluster Boundary Nodes Using PSO Algorithm*

Mamta Yadav[1], Basma Fathi[2], Alaa Sheta[1]
[1]Department of Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX 78412, USA
[2]Department of Research and Testing
New and Renewable Energy Authority
Cairo, Egypt

Abstract

The emerging area of Wireless Sensor Networks (WSNs) has been extensively studied in many real-life applications and became a very essential tool in surveillance. Grouping the sensor nodes into clusters support the data aggregation and enhance the overall system scalability. Inter-cluster communication technique is an emerging research interest in recent WSN applications. The position of Cluster Head (CH) in a cluster determines the overall energy consumption and the choice of Boundary Nodes (BNs) determines the communication energy consumption. The BNs recognition of the clusters in WSNs is a crucial issue. These nodes should be used to make the communication possible between two clusters with the minimum communication overhead. The optimum selection of BNs increases the network robustness against BNs failure. In this paper, we propose a Hybrid Particle Swarm Optimization (HPSO) algorithm to select appropriate BNs of WSNs to save energy and prolong network lifetime.

1 Introduction

WSNs were used in diverse real-life applications such as battlefield surveillance, health care systems, habitat monitoring [6], visual surveillance for automatic

object detection [9], environment monitoring [7], and noise pollution monitoring. These applications are identical life threatening and need a wide range of coverage for the area under observation. It is therefore not surprising that researches in different disciplines have contributed to the developments in WSNs. A WSN is composed of a large number of sensor nodes communicating with each other using radio signals with the objective to sense, monitor, and explain some phenomena from the surrounding environment. The sensor nodes process the measured data and send them to a remote data collector called sink node.

The WSN lifetime is a critical issue because it is an indication of network performance degradation. The best solution to increase the network lifetime is efficient routing. This problem is resolved by partitioning a WSN in several smaller sub-networks called clusters.A cluster is consists of a cluster head (CH), intra-cluster sensor nodes, and boundary nodes (BNs). The intra-cluster nodes communicate with the CH in single hop while the inter-cluster nodes communicate via BNs of clusters. Inter-cluster communication allows one cluster to send its data to another CH using its BN. The same cluster can receive data from another cluster using the other cluster's BN. Therefore, the selection of the cluster BN is an emerging challenge design for WSN.

Now a day, designing a WSN is a challenge because we need to solve some optimization problems such as: What is the optimal number of the clusters and BN we should have? If fewer clusters are to be produced, then cluster overloading occurs. If the number of clusters is high, then more nodes will connect to the sink node. The optimal selection of BNs facilitates many basic operations of WSNs such as routing, topology control, and network health.

PSO is an optimization technique inspired by the movement of bird swarms. The inspired technique consists of particles that move in the search space and communicate with each other in an attempt to search for the best location. The best location represents the solution to the optimization problem. The quality of the solution obtained is evaluated according to a fitness function proposed depending on the nature of the optimization problem. Details on PSO technique can be found in [10, 4, 1] and [3]. In this paper, we focus on the problem of developing an efficient method for selecting the optimal BNs for inter-cluster communication using PSO algorithm.

This paper is organized as follows. In Section 2, we present the proposed a Hybrid Particle Swarm Optimization (HPSO) algorithm for selecting optimal BNs. The fitness function of our proposed algorithm is discussed in Section 3 followed by simulation results in Section 4. Finally, we summarize the results in the paper in Section 5.

2 Proposed Algorithm

An HPSO algorithm is proposed to select the optimum boundary nodes of WSN clusters. The proposed algorithm is provided into two phases:

- **Phase 1: Clustering WSNs Using K-means**
  K-means is a simple unsupervised, heuristic clustering algorithm that efficiently solved a variety of clustering problems [5, 2]. It classifies a given data set into a predefined number of clusters by mainly defining a centroid for each cluster, maximizing the inter-cluster distance and minimizing the intra-cluster distance [8]. In this phase, K-means first outputs the optimum number of clusters, k_{opt}, for the network layout given based on minimizing the total cluster communication distance. Then, K-means is applied to cluster the network to the calculated k_{opt} clusters. K-means records the nearest node to the cluster center as the CH.

- **Phase 2: Inter-Cluster Boundary Nodes Selection Using PSO**
  In our work, PSO is developed to obtain the optimum boundary nodes (BNs) for each cluster obtained from the K-means phase. For a WSN of N nodes divided into k_{opt} clusters, a maximum of 10% nodes of each cluster are allowed to be boundary nodes. For example, if the network is divided into three clusters having member nodes as $\{12, 20, 50\}$, then the maximum allowed a number of boundary nodes will be $\{1, 2, 5\}$. Thus, the size of PSO particles depends on the size of each cluster. The PSO proposed particle structure to obtain the optimal inter-cluster boundary node for a WSN network is presented as follows:

BN_{11}	BN_{21}	BN_{22}	BN_{31}	.....	BN_{35}

To see how the proposed idea works, we show an example in Figure 1. The WSN is first divided into the optimum number of clusters by using the first phase of the proposed HPSO algorithm. In this example, we are showing three clusters: A, B, and C having $8, 10$, and 8 nodes respectively. The selection of the ratio of BNs to cluster nodes is decided by the designer according to the required level of reliability. In this example, 30% of the nodes are selected to be the maximum possible number of BNs. Thus, the maximum number of BNs in Clusters A, B and C are: $2, 3$, and 2 respectively.

The next step is to search for the best BNs to reduce the communication distance between any two clusters. The chosen BNs must provide the shortest path between any two clusters. The PSO-selected BNs for clusters A, B and C are: $(a_1,\ a_2)$, $(b_1,\ b_2,\ b_3)$, and $(c_1,\ c_2)$ respectively. In Figure 1, the best inter-cluster routing between clusters B and C is performed by using the BNs

b_3 and c_1, as they are close. Therefore, each BN has one intra-cluster communication with its CH and two inter-cluster communication with the CHs of other clusters.

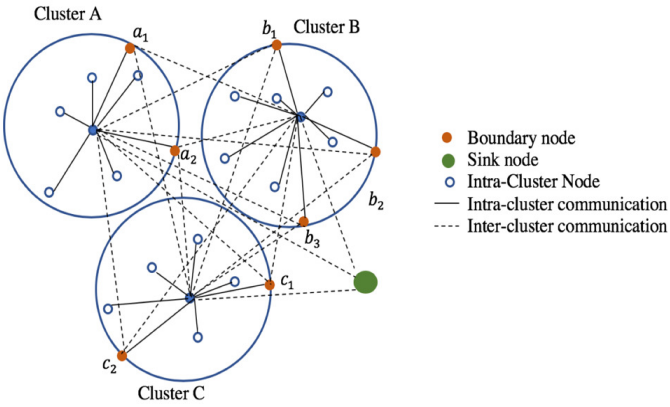


Figure 1: Nodes naming in WSNs

3 Evaluation Criteria

The fitness function is designed to choose dispirsed cluster BNs that are close to other BNs of their neighbor clusters. This choice will help in minimizing communication energy consumed. Our proposed fitness function focuses on minimizing the inter-cluster distance between the boundary nodes of different clusters and at the same time maximizes the intra-cluster distance between the boundary nodes of the same cluster. The proposed fitness is calculated as:

$$F = \frac{Total\ Intercluster\ Distance}{Total\ Intracluster\ Distance}$$

The total inter-cluster distance is the sum of the Euclidean distances of each BN and the other BNs that are not in the same cluster. The total intra-cluster distance is the sum of Euclidean distances between each CH and its BNs.

4 Simulation Results

Our HPSO algorithm is tested for a randomly generated network of 100 nodes located in a geographic area of $100\times100m$ wide. The HPSO algorithm first

decides the best number of clusters needed for the given layout then clusters the network using the K-means algorithm. Then the PSO algorithm locates the BNs for each cluster obtained from the K-means phase; Figure 2(a) displays the clusters layout and the BN within each cluster. The BN is marked with a filled square. The figure shows that the boundary nodes for each cluster are dispersed to cover the whole cluster and its neighbors. Figure 2(b) shows the conversion curve for the PSO algorithm to the optimum fitness value.

5 Conclusion

In this paper, we presented and implemented an algorithm to select the optimum number of boundary nodes in each cluster of WSNs. Our proposed HPSO algorithm gets the optimum clusters and CHs in the first phase and then selects the optimal boundary nodes for each cluster in the second phase. The fitness function of our proposed algorithm focused on minimizing the inter-cluster communication distance. It also aimed to select distantly separated BNs within the same cluster to avoid redundancy. Also depending on the WSN layout, the same boundary node could be redundantly chosen. Then, the optimal boundary nodes could be less than the maximum allowed. Therefore, PSO manages to select the boundary nodes sufficient for the given layout.

References

[1] D. Bratton and J. Kennedy. Defining a standard for particle swarm optimization. In *Proceedings of the IEEE Swarm Intelligence Symposium*, pages 120–127, 2007.

[2] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881–892, 2002.

[3] J. Kennedy. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *Proceedings of the 1999 Congress on Evolutionary Computation*, volume 3, pages 1931–1938, 1999.

[4] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.

[5] S. J. Vinay Kumar and S. Tiwari. Energy efficient clustering algorithms in wireless sensor networks: A survey. *International Journal of Computer Science Issues*, 8(5):259–268, 2011.

[6] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, WSNA '02, pages 88–97, New York, NY, USA, 2002. ACM.

[7] Bushra Rashid and Mubashir Husain Rehmani. Applications of wireless sensor networks for urban areas. *J. Netw. Comput. Appl.*, 60(C):192–219, January 2016.

[8] G. A. Wilkin and X. Huang. K-means clustering algorithms: Implementation and comparison. In *Proceedings of the second International Multi-Symposiums on Computer and Computational Sciences*, pages 133–136, 2007.

[9] S. E. Yoo, P. K. Chong, T. Kim, J. Kang, D. Kim, C. Shin, K. Sung, and B. Jang. Pgs: Parking guidance system based on wireless sensor network. In *IEEE 2008 3rd International Symposium on Wireless Pervasive Computing*, pages 218–221, New York, NY, USA, 2008. IEEE.

[10] C. Zhang and S. Xi. K-means clustering algorithm with improved initial center. In *Proceedings of the second International Workshop on Knowledge Discovery and Data Mining*, pages 790–792, 2009.

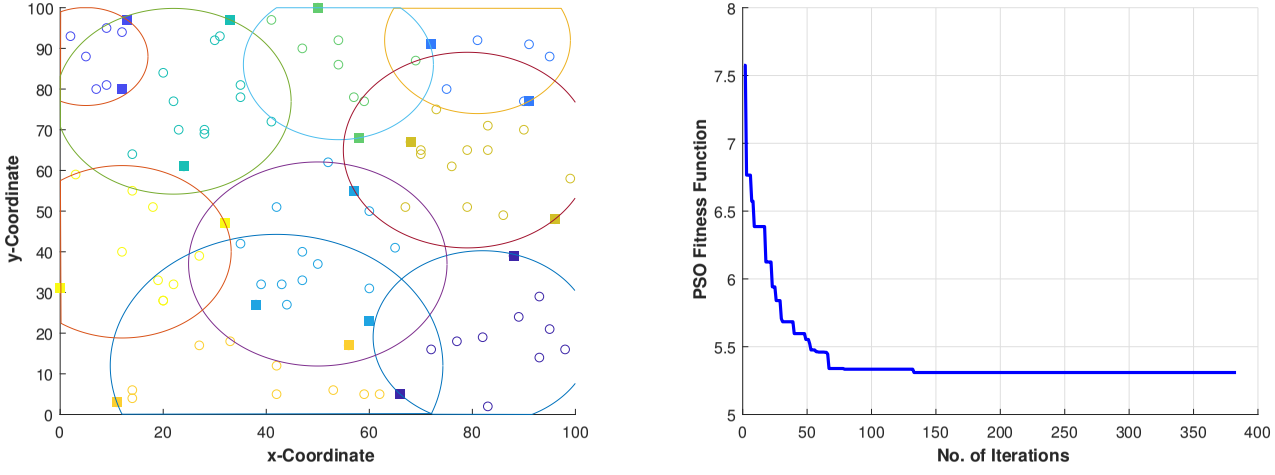


Figure 2: (a) Obtained Boundary Nodes distribution for 100 sensors of 9 clusters (b) PSO conversion curve

Crime in the 21st Century: A Co-Teaching Experience*

Bilal Shebaro[1], *Casie Parish Fisher*[2]
[1]*Department of Computer Sciences*
St. Edward's University
Austin, TX 78704
bshebaro@stedwards.edu
[2]*Department of Forensic Science*
St. Edward's University
Austin, TX 78704
casiep@stedwards.edu

Abstract

Crime in the 21st century has brought about new challenges for the discipline of forensic science. As technology has become an integral part of our everyday lives, we not only need to see it as an investigative tool, but also work to understand its value as evidence in a crime scene. This co-teaching experience utilizes the expertise of a computer scientist and a forensic scientist to merge the two worlds together in an effort to educate students in both fields. Passive and active learning strategies were employed to conduct this course of diverse topics. The interdisciplinary nature of the course provided a valuable and positive learning experience for the students.

1 Introduction

As the world we live in becomes more technologically advanced and dependent on technology, the electronic devices encountered at crime scenes have become increasingly valuable components of criminal investigations. As a result, crime

scene investigations have evolved to include technology-based work and evidentiary items that may now also contain a digital footprint. Therefore, there is a need to educate forensic science personnel about the technological changes that can significantly affect their daily tasks, and to incorporate the evidentiary processing of digital evidence into the field of forensic sciences [12, 1]. As college educators, we realized the need to introduce this knowledge into a course that connects the fields of forensic sciences and computer science so that personnel from both fields can easily communicate and work with one another.

In this paper, we introduce an interdisciplinary course that reflects the changing crime scene investigation environment of the 21st century. "Crime Scene in a Digital World" is a course co-taught by faculty from the Computer Science and Forensic Science departments that seeks to connect the expertise of both fields to solve crimes involving digital devices and evidence. Not only does this course introduce students to the technologies involved in crime scene processing and the analysis of electronic evidence, but it also provides them with a fundamental understanding of the legal, technical, management, and behavioral factors associated with conducting digital crime investigations. The purpose of this paper is to discuss the curriculum design for an interdisciplinary course in the field of digital forensics that aids all students in understanding the process of criminal investigations in the 21st century.

This paper is organized as follows: Section 2 introduces the idea and motivation behind planning this course. Section 3 highlights on the context of the course from preparing the syllabus to the choice of course topics and project, and finally to how we did our assessments and examinations. Section 4 expands on the student vocation exploration on how we focused on developing a deeper understanding of the connection between education and careers. Finally, we highlight on the student experience in Section 5, followed by our concluding thoughts in Section 6.

2 How it all started

The idea behind this course was to allow students to be able to conduct a criminal investigation involving digital devices. Commonly, the forensic science students become well-educated about criminal and civil laws, and become specialized in performing either crime scene investigations or laboratory analysis, while sparing the extraction of digital evidence from electronic devices for the digital forensic technicians. Similarly, computer science students specializing in digital forensics usually learn about various techniques for data retrieval from digital devices without going through the process of how such devices are retrieved and recovered from a crime scene. As a result, we (the authors), as professors from Computer Science and Forensic Science departments, com-

bined efforts to develop and co-teach a course to introduce students to both worlds of criminal investigations and digital evidence retrieval, highlighting on the emerging technology reformation on crimes in the 21st century.

When the course was offered in Fall 2017, it attracted students from the computer science and forensic science fields. We had a total of 28 students, 15 students were Computer Science majors and 13 were Forensic Science majors. Given the varying skill sets and prior knowledge brought by the students, teams were composed of both computer science and forensic science students for the course project as we detail in Section 3.1.4.

3 Context

In this section, we highlight how we prepared the course material, sharing our experience on the chosen course topics, assessments, and team projects.

3.1 Course Preparations

3.1.1 Syllabus

We prepared our syllabus by focusing on the learning outcomes that we wanted students to accomplish upon taking this course [4, 10, 3]. Our course description and objectives were derived based on preparing our students with the knowledge, skills, and experiences so they would become capable of performing all tasks related to crime scene investigations, starting with arriving at the crime scene and following proper procedures for electronic device acquisition until the presentation of digital evidence in court [9, 7, 11]. Even though the course was offered at the junior and senior level, we did not put course prerequisites in place given the diversity of students' backgrounds and experiences. Ethical considerations were also taken due to the recovery nature of some of the digital material. Students did sign ethical agreements in regards to not using learned techniques for malicious purposes.

3.1.2 Lecture Topics

Given our familiarity with the students' background knowledge from their major courses, we structured our lesson plans in such a way that the lessons were accessible and engaging to students while following a logical sequence [6, 8]. The course topics were arranged in the order that students would need to perform the various tasks in a full criminal investigation process and recovery of digital evidence. Figure 1 displays an outline of some of the course topics.

As instructors of the course, we met before and during the semester to ensure the flow of the course topics were as planned and made adjustments

Theme	List of Topics
Crime Scene Investigations	▪ Overview of Technologies used in Crime Scene Investigations: Digital cameras, laser measuring tools, video, 3-D scanners
Evidentiary Guidelines and Protocols	▪ Organizations and protocols associated with digital evidence and forensic images (SWGIT/SWGDE/NIST)
The Project: Mock Crime Scene	▪ Crime Scene Investigations: The mock crime scenes gave students a hands-on experience in working a crime scene and collecting evidence within the scene. The evidence included fingerprints and 'blood', but focused on mobile phones, computers, USB drives, etc.
	▪ Collection and Documentation of Evidence: Review of proper documentation and collection techniques
	▪ Recovering evidence from Digital Evidence: The recovery of fingerprints, bloodstains, and other types of evidence that potentially could be on electronic evidence.
Analysis of Evidentiary Items	▪ Recovery of deleted files
	▪ Extraction Data from USB drives & external storage
	▪ Extracting Data from Internet Browsers
	▪ Extracting metadata from digital photographs
	▪ Computer Network Forensics
	▪ Computer Memory Forensics
	▪ Mobile Forensics
	▪ Recovering footprints from using Anonymity Networks & the Dark Web
	▪ Steganography & Watermarks
Digital Forensics in Pop Culture, News, & Media	▪ CSI: *Cyber*
	▪ News Media video clips/ Cases in the media
Vocation Exploration (Guest Speakers)	▪ Technology Used in Law Enforcement
	▪ Recovery of Information of Evidentiary Items

Figure 1: Course themes and lecture topics

as needed [4]. During the planning phase of the course, it was important to split the topics according to the perspective being discussed. For example, for the Monday/Wednesday course, Mondays were dedicated to recovering evidence from the crime scene and the technology currently being used in investigations, in addition to the recovery of secondary evidence from electronic devices, and the collection/preservation guidelines for obtaining electronic evidence [5]. Wednesdays were dedicated to the recovery of digital information from a variety of different types of softwares and electronic devices that contain additional information that may be probative to an investigation. Activities were also designed during the semester to bridge both topics, including scenario driven engaging activities that required reflection on proper collection as well as recovery from the items.

3.1.3 Assignments and Examinations

Assessment activities for the course included in-class and homework assignments, as well as two examinations. Homework assignments were designed to link lecture topics with real world cases and research. One such assignment

was to write a summary and critically evaluate a digital-based case which has been covered in the media, while another assignment was related to reviewing a published research article in the field of digital forensics. Two assignments were specifically focused on learning skills for the course projects as described in Section 3.1.4. Exams consisted of multiple choice questions, short answer responses, essay questions, and case scenarios. In total, students conducted 4 homework assignments, a course project of two phases, one midterm exam and one final exam.

3.1.4 The Project

While designing the group project, it was important to consider the student groups' different skill sets and background. The goal of the project was to give students a hands-on, engaging experience that incorporated both a crime scene component and a digital component to the activity, allowing each student group the opportunity to share their particular skill set. Therefore, the student teams were formed to include at least one computer science major and at least one forensic science major. The project assignment was composed of two phases where students worked in teams of 3 or 4 students. Phase 1 of this project consisted of working on a mock crime scene, which included taking photographs of the scene, and recording measurements of the items within the scene, and their inter-spatial relationship with the evidence. Phase 2 consisted of using these measurements and sketches to build a 3D crime scene model using the computer software SketchUp [2], in addition to recovering digital evidence from a computer device placed in the mock crime scene. SketchUp is a software program that is mostly used in the architecture field to build 3D models of houses and building structures, but we choose to use it for this purpose as it is becoming an industry tool for reconstructing crime scenes in a more graphical form that can be admissible in the court. Students were given a short lecture on how to use SketchUp, but 3D sketching and modeling was entirely a student self-learning experience. A sample output of project phase 1 (shadow box made out of cardboard), and phase 2 (3D model using SketchUp) is shown in Figure 2.

4 Vocation Exploration

Vocational exploration is important for students to be exposed to during their course of study. This course specifically allowed students from two different disciplines to see how their areas of interest could merge into a job in the future. During the course, two sets of guest speakers were asked to come to campus to share their job experiences. The first was a Technology Unit (TU) from a

(a) Phase 1: Shadow Box cardboard (b) Phase 2: 3D SketchUp model

Figure 2: Sample student project output.

local law enforcement agency. The TU personnel described the different types of technology and software utilized by law enforcement to track and record law enforcement activities and actions. The second was personnel from the Multimedia and Questioned Documents section of a state forensic laboratory. The personnel from the laboratory discussed cases that involved electronic evidence and the data recovered from such items. Cellular phones, computers, and other mobile devices were some of the topics discussed in detail. The guest speakers visits were purposely planned to be at the end of the semester so topics and techniques discussed and conducted throughout the semester were reinforced with real world experience validation.

The social perception of the abilities of forensic scientists has been somewhat exaggerated by the sensationalizing of the discipline by popular television shows such as CSI, Dexter, NSCI, and others. Faculty also incorporated clips of television shows/news stations into specific topics covered to reinforce the truth behind the processes conducted to recover evidence from electronic devices. A lecture anchored by CSI: Cyber which explored the shows techniques

and technologies and whether they were 'fact or fiction' was incorporated into the course topics. Students were able to assess some of the job duties required of a forensic analyst who recovers data from electronic evidence as well as identify misconceptions portrayed in the show.

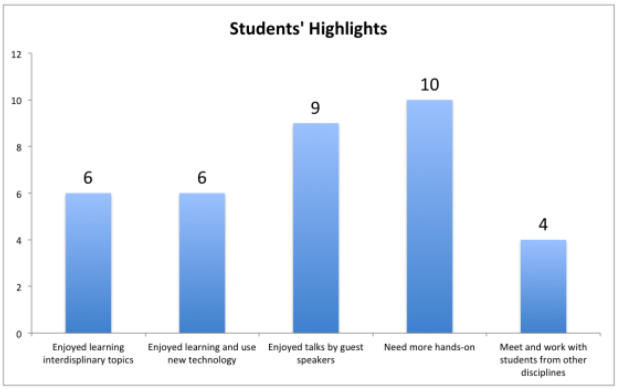
In addition to television shows and news station clips, students were also exposed to a variety of scientific articles that supported the research, techniques, and technologies discussed during the course. Peer reviewed articles are an important resources for keeping up-to-date on the most current technologies and trends. This also gave each student group another avenue to explore the opposite discipline.

5 Student Experience

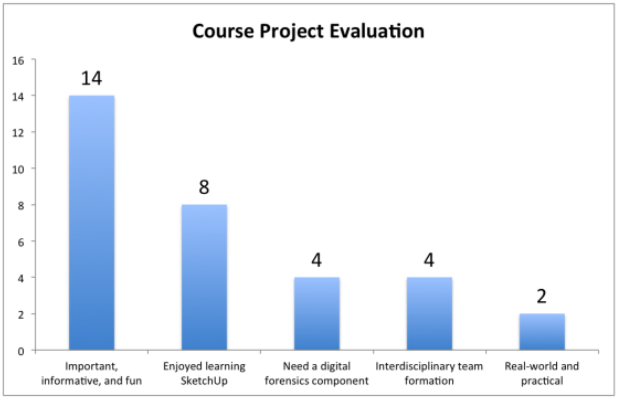
Overall student evaluations of the course were positive as shown in Figure 3. When reflecting on the highlights of the course, students overwhelmingly remarked that they enjoyed the talks by guest speakers, and in the future they would like even more hands-on activities to support topics covered in the course. Students also enjoyed the interdisciplinary nature of the course including learning new interdisciplinary topics and working with students from a different discipline. Overall, the majority of students thought the course project was important, informative, and fun, and they enjoyed learning the 3D SketchUp software (Figure 3). Given the student interest in the course project and on performing more hands-on experiments, some students mentioned the need of more digital components in the course project. Even though the project had a digital component in relation to recovering deleted data from flash drives as well as finding footprints on external drives used on computers, we plan to include even more digital components to the project in the future as soon as we are able to provide the more expensive equipment and software that would be needed for the students' experiments and investigations. Overall, the project was an important component to the course, and the goal of merging the two disciplines was successful by utilizing an assignment of this nature.

6 Conclusion

In conclusion, we (the authors) feel the course was successful in exposing students to digital forensics in a fun and engaging course design. While noting some weaknesses that will be addressed in the future, the pilot course conducted successful projects while incorporating research and social media into exploring Crime Scene in a Digital World.



(a) Student responses to questionnaire regarding the highlights of the course.



(b) Student evaluations of the course project which included both Phase 1 and Phase 2.

Figure 3: Sample survey results.

References

[1] Olga Angelopoulou and Stilianos Vidalis. An academic approach to digital forensics. *Journal of Information Warfare*, 13, 01 2015.

[2] Elissa St. Clair, Andy Maloney, and Albert Schade. An introduction to building 3d crime scene models using sketchup. *Journal of the Association for Crime Scene Reconstruction*, 4(18):29–47, 2012.

[3] J. Crow and L. Smith. Using co-teaching as a means of facilitating interprofessional collaboration in health and social care. *Journal of Interprofessional Care*, 17(1):45–55, 2003.

[4] Jacqueline Ferguson and Jenny C. Wilson. The co-teaching professorship: Power and expertise in the co-taught higher education classroom. *Scholar-Practitioner Quarterly*, 5(1):52–68, 2011.

[5] Ross M Gardner. *Practical crime scene processing and investigation.* Boca Raton, Fla. : CRC Press, second edition edition, 2012.

[6] S. Kiltz, J. Dittmann, and C. Vielhauer. Supporting forensic design - a course profile to teach forensics. In *2015 Ninth International Conference on IT Security Incident Management IT Forensics*, pages 85–95, May 2015.

[7] Anthony Lang, Masooda Bashir, Roy Campbell, and Lizanne DeStefano. Developing a new digital forensics curriculum. *Digital Investigation*, 11(S2):S76–S84, August 2014.

[8] L. McDaniel and B. Hay. Teaching digital forensics techniques within linux environments. In *2014 47th Hawaii International Conference on System Sciences*, pages 4848–4856, Jan 2014.

[9] Ashraf Mozayani and Casie Parish-Fisher. *Forensic Evidence Management: From the Crime Scene to the Courtroom.* CRC Press, 2017.

[10] W. M. Roth and K. G. Tobin. Coteaching: From praxis to theory. *Teachers and teaching: Theory and practice*, 10(2):161–180, 2004.

[11] S. Srinivasan. Computer forensics curriculum in security education. In *2009 Information Security Curriculum Development Conference*, InfoSecCD '09, pages 32–36, New York, NY, USA, 2009. ACM.

[12] Alec Yasinsac, Robert Erbacher, Donald G. Marks, Mark M. Pollitt, and Peter Sommer. Computer forensics education. *Security and Privacy, IEEE*, 1:15 – 23, 08 2003.

A Case Study on the Dialect Identification of Twitter Tweets Using Natural Language Processing and Machine Learning*

Kari Djuve, John W. Burris
Department of Computer Science
Southeastern Louisiana University
Hammond, LA 70402
{kari.djuve,jburris}@southeastern.edu

Abstract

Dialectology is the linguist field that focuses on the study of regional variances among speakers of a language. Although linguists primarily use spoken data for dialect studies, written language also shows marked differences between members of different dialectal groups. This study aims to use a supervised decision tree classifier to predict the dialect class of some written texts. Since the context in which the writer is writing in – i.e. formal letter versus journal entry – plays a major role in the style used by writers, a data set of texts pulled from the social media platform Twitter will be used. Tweets, the messages users post on the social media platform Twitter, are short and informal, thus very likely to be written in the users' dialect. Ten thousand tweets are pulled from two dialectal regions – Southern American English and New England American English dialects. However, these texts are also noisy. This study goes over how to normalize, or clean, these noisy texts so that Natural Language Processing (NLP) tools can be used for feature selection. Common feature selection methods will be applied to the texts to select the most information rich features while reducing the number of features that are redundant or too common to be useful in differentiating between

the different dialect classes. Two feature selection pipelines are applied for comparison. The Feature Selection 1 pipeline removes punctuation, non-alphabetic tokens, and stop-words. The Feature Selection 2 pipeline extends the Feature Selection 1 pipeline by adding a stemmer. Additionally, Term Frequency-Inverse Document Frequency (TF-IDF) weighting will be used for comparison. After the feature selection pipelines, a decision tree classifier is used to classify the texts.

1 Introduction and Background

Dialectology is a field of linguistics that studies regional differences among speakers of a language. A dialect is the systematic usage of a group of speakers [11] or simply as variations of the same language – this includes the written form of a language as a dialect of that language [16]. Linguists define dialects of a language when there are consistent differences in the lexicon, phonology, and grammar [11] [16].

When studying dialects, primary attention is given to speech rather than writing [11]. The written language does not offer the phonological information that is a major focus in linguistics. Also, writing is a more restricted skill. That is, everyone learns to speak before they learn to read or write. Focusing on written, then, can potentially omit a large portion of a population who are illiterate. For these reasons, there has not been many studies of dialects using written texts as the source material.

However, the same linguistic features used to define spoken dialects can be used to determine different written dialects of a language, with the exception of phonological variations. Instead of phonemes (units of sound), graphemes (or units in writing) are used, along with lexical and grammatical clues, in distinguishing between dialects and making a prediction of the dialect of some written text. For this study, the texts are tweets from the Twitter social media platform.

Features are properties that describe the data [8]. For example, in a data set of different species of iris flowers, the features are sepal length, sepal width, petal length, petal width. For NLP and machine learning (ML) tasks involving text, words themselves are features [3] [15]. A common problem in NLP is having high feature dimensionality relative to the sample size of the data set. It has been termed the "Curse of Dimensionality" [17] [1] [5]. The dimensionality of a data set is equal to the number of features. Given that the features for a NLP data set often includes the words themselves, the dimensionality can become quite high for such a text data set. This is extremely problematic for numerous reasons, especially when it comes to the computational complexity [17]. An additional problem is that there are likely many redundant and ir-

relevant features, which also contribute to the computational complexity and do not offer much, if any, gain in correctly separating the data into the target classes. The goal, therefore, is to reduce the number of features as much as possible in order to train a classifier with good generalization capabilities [8] [17] [9].

Feature selection, also called normalization [1] or the preprocessing stage [17], is used to filter out redundant and irrelevant features producing a subset of original features that are meaningful to the NLP/ML task at hand. This reduction in features naturally leads to a reduction in dimensionality and improvements to the computational complexity and the classifier's generalization capabilities [5]. There are many common methods for selecting the features of a feature set for NLP studies.

2 Method For Dialect Identification

For this study, a data set of tweets taken from geographical locations located within the two dialect regions outlined by William Labov – a renowned Linguistic researcher best known for his work on U.S. dialect studies – on his map outlining U.S. dialect regions, see [4]. In order to collect tweets for the data set, Python was used to write a wrapper for the Twitter API. Tweets are objects with many attributes. One such attribute is the Geocode, which returns tweets by users' location within a given radius of the given latitude, longitude coordinates [2]. Tweets were pulled from the Southern dialectal region and from the New England dialectal region. Regular expressions were used also used to make sure that none of the tweets pulled were retweets, which is the reposting of someone else's tweet. One caveat is that tweets pulled from the geographical region are not guaranteed to be from a native speaker of that region or its affiliated dialect. Tweets are geotagged depending on the user's location when they post the tweet. Therefore, there is no way of knowing for sure if a tweet is truly written by someone of that dialectal region, but we will assume that, for the most part, the majority of tweets are by residents of that dialectal region and exhibit a notable use of that region's dialect in their writing.

In all there are six separate python code files: Case 0.a, Case 0.b, Case 1.a, Case 1.b, Case 2.a, and Case 2.b:

- Case 0.a and Case 0.b only normalize the tweets, with the addition of TF-IDF weighting on Case 0.b. These two files test how well a classifier – decision trees in this case – performs without additional feature selection methods (other than the term frequency-inverse document frequency weighting on Case 0.b). The specific normalization done will be explain in the next section.

- Case 1.a and Case 1.b apply the same normalization as the above two cases in addition to three feature selection techniques: punctuation removal, non-alphabetic character removal, and stopword removal. TF-IDF weighting is applied to Case 1.b.
- Case 2.a and Case 2.b add one additional feature selection technique, stemming, to the feature selection pipeline used for Case 1.a and Case 1.b.

Table 1 below offers a summary of the normalization and feature selection pipelines taken for each case. Each case builds from the previous one. Performing the experiment in this fashion will allow for a comparison to be made between not using any feature selection (Case 0) and the two feature selection pipelines (Cases 1 and 2). Also, the effect of TF-IDF weighting on performance will be compared for each of the Cases versus not using any weighting.

A goal of this study is to build a ML model to predict the correct regional dialect class labels for a data set of tweets. After normalizing the tweets and performing feature selection the next step is to train and test a ML model. The data set of normalized tweets needs to be split into a training set and a testing set [8] [7]. The training set will be used to build the ML model, also called fitting. The testing set, also known as the hold-out set, contains previously unseen samples which will be used to evaluate how well the model classifies the samples. The testing set will be passed into the trained ML model without any true class labels. Instead, the model will predict the class label for each sample in the testing set. The model can then be evaluated by comparing the predicted class values to the true class values for the testing set [8].

The remaining parts of this section explain in detail how the data set is split into the training set and testing set, how the data sets are transformed from textual representations to a sparse matrix, how the ML model is built, and how the model predicts class labels for the testing set.

A library function [14] was used to split the data into training and testing sets. By default, this function shuffles the full data set then randomly extracts 75% of the rows in the data set as the training set, including their corresponding true class labels. The remaining 25% of the data set plus their corresponding true class labels becomes the testing set. These parameters can be adjusted depending on the particular data set and requirements [8]. For this study, these default values were adequate.

The training and testing sets consists of tweet text, or natural language, which needs to be transformed into a matrix of tokens. This process is called vectorization [13]. Text data needs to be vectorized in order to perform the NLP and ML tasks since computers do not understand human language. There are two scikit-learn functions that are used to do this: CountVectorizer and TfidfVectorizer.

Table 1: Summary of Normalization and Feature Selection Pipelines

Case 0.a : Normalized Only	Case 0.b : w/ TD-IDF
Escape HTML chars	Escape HTML chars
Decode data	Decode data
Tokenize	Tokenize
Lowercase	Lowercase
	Apply TD-IDF weighting
Case 1.a : Feature Selection 1	**Case 1.b : w/ TD-IDF**
Escape HTML chars	Escape HTML chars
Decode data	Decode data
Tokenize	Tokenize
Lowercase	Lowercase
Remove punctuation	Remove punctuation
Remove non-alphabetic tokens	Remove non-alphabetic tokens
Remove stop-words	Remove stop-words
	Apply TD-IDF weighting
Case 2.a : Feature Selection 2	**Case 2.b : w/ TD-IDF**
Escape HTML chars	Escape HTML chars
Decode data	Decode data
Tokenize	Tokenize
Lowercase	Lowercase
Remove punctuation	Remove punctuation
Remove non-alphabetic tokens	Remove non-alphabetic tokens
Remove stop-words	Remove stop-words
Stemmer,lemmatize	Stemmer,lemmatize
	Apply TD-IDF weighting

CountVectorizer essentially transforms the text data into a sparse matrix using a bog-of-words technique. That is, a count of each of the unique words is made and stored in the matrix where each position in the matrix corresponds to a word. The other vectorizing function used is TfidfVectorizer. This function converts a collection of raw text documents to a matrix of TF-IDF features [13]. TF-IDF is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus [6]. The goal is to give higher weightings to terms that appear often in a particular document, but not in many documents. If a word appears often in many of the documents, it is not a good feature for discriminating between classes. Likewise, if a word appears often in some documents and not in others, it is likely a good word for discriminating between classes.

The ML model used in this study is built using a decision tree. A decision tree is a ML model that essentially builds a hierarchy of if/else questions which lead it to a decision [8]. In the case of a classification task, the decision is a class label. Decision trees are one of the most widely used and practical methods for inductive inference since it is robust to noisy data [7]. This means that decision trees are relatively insensitive to errors in classification of the training samples and errors in the attribute values that describe the examples. For this reason, a decision tree classifier was chosen.

3 Analysis and Discussion

A confusion matrix is commonly used when evaluating the performance of a ML model. A confusion matrix summarizes the classification performance of a classifier by comparing the predicted class label for each sample to its actual true class label which was withheld when making predictions [12].

Four important counts are derived from a confusion matrix: true positive, false negative, false positive, and true negative. By convention when constructing a confusion matrix, labels are designated as either positive or negative [10]. For this study, class labels with a value of 1, that is the class label is Southern, are positive and class labels with a value of 0, that is the class label is New England, are negative. Thus, the true positive count represents how many of the testing set samples whose true class is positive were correctly predicted positive by the model; the false negative count represents how many of the true positive samples were incorrectly predicted negative by the model; the false positive count shows the number of samples incorrectly predicted positive when the true class was negative; and true negative count is the number of samples whose true class is negative and were correctly predicted by the model.

Several performance measures are calculated using the true positive, false negative, false positive, and true negative counts. This paper looks at four common measures used for evaluation: accuracy, recall, precision, and f-score.

The confusion matrix values were computed for the decision tree model of each of the six cases. Table 2 shows the results.

Accuracy, recall, precision, and f-score measures were computed for each case as well. The results are in Table 3 along with the mean, median, and range for each measure.

The accuracy measure calculated for each of the Cases differed by about 3%, with a mean of 56.17%, and an average median of 55.73%. Case 0.a, the Case that was normalized only and used CountVectorizer to transform the data into a sparse matrix, had the highest accuracy score at 57.86%. Case 1.b, the Case that used the Feature Selection 1 method and TfidfVectorizer weighting,

Table 2: Confusion matrix values for the decision tree model for each of the six cases. Italicized difference indicated the increase or decrease for that measure by including Tfidf weighting

Case	True Pos	False Neg	False Pos	True Neg
Case 0.a	1353	983	1540	1124
Case 0.b	1365	1092	1431	1112
Effect of Tfidf	12	109	-109	-12
Case 1.a	1509	1172	1351	968
Case 1.b	1485	1261	1262	992
Effect of Tfidf	-24	89	-89	24
Case 2.a	1478	1224	1299	999
Case 2.b	1526	1272	1251	951
Effect of Tfidf	48	48	-48	-48

had the lowest accuracy at 54.94%.

Recall varied by approximately 7% among the six different Cases, with a mean of 58.65% and a median of 59.81%. Case 2.b, the Feature Selection 2 + TfidfVectorizer weighting, had the highest recall score at 61.61%. Case 0.a had the lowest at 54.62%.

Precision varied by approximately 4%. The mean precision score was 55.51% and the median was 55.13%. Similar to the accuracy score, Case 0.a had the greatest precision score at 57.92%, while Case 1.b had the lowest at 54.08%.

F-score, which strikes a balance between recall and precision, was highest for Case 1.a, the Case that used the Feature Selection 1 method, with an f-score of 58.51%. Case 0.b, the Case that normalized only with TfidfVectorizer weighting used, had the lowest f-score at 55.33%. The f-scores differed by approximately 3%, with a mean of 56.98% and a median of 56.97%. Many ML practitioners favor the f-score over accuracy since models with a high f-score also has good recall and precision. Note how the accuracy for Case 1.a, which has the highest f-score, has the second highest accuracy score with only a 0.66% difference between the two.

With the exception of the precision score, all cases performed fairly similarly. Case 0.a had the most performance measures with the highest scores. This could be indicative of overfitting as stricter feature selection methods are applied. Overfitting occurs when the model is fit too closely to the training data [12]. So when new, previously unseen samples from the testing set are passed into the over-trained model, it does not make good generalizations and performance decreases [8]. Figure 1 illustrates this.

As accuracy on the training set increases when training the model, the ac-

Table 3: Performance measures for the decision tree model for each of the six Cases

Case	Accuracy	Recall	Precision	F-score
Case 0.a	57.86%	54.62%	57.92%	56.22%
Case 0.b	55.92%	55.11%	55.56%	55.33%
Case 1.a	57.20%	60.92%	56.28%	58.51%
Case 1.b	54.94%	59.95%	54.08%	56.86%
Case 2.a	55.54%	59.67%	54.70%	57.08%
Case 2.b	55.54%	61.61%	54.54%	57.86%
Mean	56.17%	58.65%	55.51%	56.98%
Median (Averaged)	55.73%	59.81%	55.13%	56.97%
Range	2.92	6.99	3.84	3.18

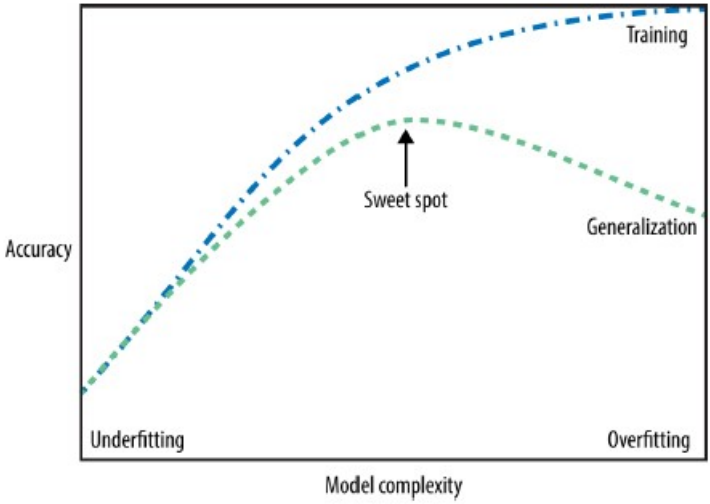


Figure 1: Trade-off of Model Complexity Against Training Set and Testing Set Accuracy [3]

curacy on the testing set increase until it reaches a maximum point. After accuracy on the testing set reaches it max, further accuracy gains on the training set lead to poorer generalizations and, thus, accuracy decreases when the model is tested with previously unseen samples from the testing set.

All of the cases start with the same data set and split that data set in the

exact same way. Therefore, the only difference between the six cases is the feature selection methods that are applied to the data set and for Cases 0.b, 1.b, and 2.b, the use of TF-IDF weighting, as illustrated in Table 2. Each case progressively refines the features selected for training their decision tree model. This refinement could be leading to a model that is finely tuned to the features of the training set and performs poorly when given new, previously unseen samples in the testing set.

4 Conclusions

The problem this study set forth to solve was identifying the dialect of written text. To accomplish this, a data set of tweet texts was compiled. The data set was then normalized, and feature selection techniques were used. The data set was then split into a training set and a testing set and transformed into sparse matrices. Next a decision tree classifier was initialized using the training set. Class predictions were made by testing the model with the testing set. Model evaluations were done by comparing the model's class predictions to the true class labels for the samples.

The model with the highest accuracy and precision was Case 0.a, the case that only normalized the text. The model with the highest recall was Case 2.b, the case that used the feature selection 2 pipeline plus TF-IDF weighting. And the model with the highest f-score was Case 1.a, the case that used the feature selection 1 pipeline.

References

[1] Steven Bird, Ewan Klein, and Edward Loper. Nltk book. `http://www.nltk.org/book/`.

[2] Steven Bird, Ewan Klein, and Edward Loper. Standard search api. `https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets.html`.

[3] W. Etaiwi and A. Awajan. The effects of features selection methods on spam review detection performance. In *International Conference on New Trends in Computing Sciences*, ICTCS '17, pages 116–120, 2017. `https://doi.org/10.1109/ICTCS.2017.50`.

[4] W. Labov, S. Ash, and C. Boberg. Phonetics, phonology, and sound change : a multimedia reference tool. *The Atlas of North American English*, 2006.

[5] H. Liu, H. Motoda, R. Setiono, and Z. Zhao. Feature selection: An ever evolving frontier in data mining. *Feature Selection and Data Mining*, 10:4–13, 2010.

[6] C. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[7] T. M. Mitchell. *Machine learning*. McGraw-Hill, Singapore, 1997.

[8] A. Muller and S. Guido. *Introduction to machine learning with Python*. OŔeilly, 2017.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] D. Powers. Evaluation: From precision, recall and f-factor to roc. *Informedness, Markedness & Correlation*, 24, 2011.

[11] J. R. Rickford. How linguists approach the study of language and dialect. `https://www-leland.stanford.edu/~rickford/papers/l73_reading_1.doc`.

[12] C. Sammut. *Encyclopedia of machine learning and data mining*. Springer, New York, NY, 2016.

[13] Scikit-learn. Scikit-learn: Feature extraction and normalization. `https://scikit-learn.org/stable/modules/preprocessing.html#preprocessing`.

[14] Scikit-learn. Scikit-learn: Model selection and evaluiation. `https://scikit-learn.org/stable/model_selection.html#model-selection`.

[15] S. K. Srivasatava, R. Kumari, and S. K. Singh. An ensemble based nlp feature assessment in binary classification. In *International Conference on Computing, Communication and Automation*, ICCCA 2017, pages 345–349, 2017. `https://doi.org/10.1109/CCAA.2017.8229840`.

[16] J. C. Stalker. Written language as a dialect of english. *College Composition and Communication*, 25(4):274–276, 1974.

[17] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, Burlington, MA, 2009.

[Mapping=tex-text, Scale=MatchUppercase]Courier Courier[Mapping=tex-text,Scale=1.0]

A System to Support a Test-Centric Mindset in Early Programming Courses*

Michael Kart
Department of Computer Sciences
St. Edward's University
3001 S. Congress Avenue
Austin, TX 78704
michaelkart@stedwards.edu

Abstract

Test cases play an important role in software development and instructors want students to gain experience employing test cases when working on assignments. Moreover, research shows that instructors can improve student-learning outcomes by adopting a "Test-first" perspective. However, many automated test systems make test cases opaque to students. This prevents students from being exposed to the internals of test cases and the mechanics of execution, from being able to examine their code against a failing test case using a debugger, and from learning how to write their own test cases. In addition, students are not in a position to explore how various concepts from the assignment are being used. In this paper we detail the anatomy of a test-centric system that is free from these shortcomings.

1 Introduction

Test First Programming (TFP) is the software development practice of writing an automated test case before the corresponding code that is intended to satisfy it. TFP has been embraced by several agile software development

methodologies, including Extreme Programming, and has been shown to improve software quality in industry [1]. TFP has also been shown to improve student understanding of programs [6] and performance [4].

Many systems have been invented in order to support a TFP development process for student assignments. One such system is Web-CAT, which was developed by S. Edwards [3, 2]. Students submit their code to this web-based system, which evaluates the submission against instructor test case data and generates a report that indicates test case success and failure. Students also submit test cases, which are evaluated against an instructor reference implementation.

Instructors trying to use a TFP approach in introductory programming courses face a couple of significant issues when using this kind of system. First, students in these courses are learning object-oriented programming basics, while writing effective test cases relies on these basics as well as a conceptual understanding of interfaces. Second, students are unable to interact with test cases using a debugger when trying to analyze test case failures. Without being able to trace through test case execution, students are often unable to determine the root cause of a failure. For instance, a test case that is attempting to exercise the behavior of an instance method can fail due to a problem elsewhere (e.g., NullPointerException thrown from a constructor call).

Giving students a trusted set of test cases provides many other benefits, including:

- Students have examples of what good test cases look like for the problem at hand
- Students can consider these test cases to be part of the specification

2 System

We have developed a system to support using TFP in introductory programming courses. Minimum system requirements include Java 1.8, Eclipse Luna, and JUnit 4 [5].

2.1 Components

The system components shown above in Figure 1 are detailed here. Some components are specific to each assignment (e.g., AssignmentImpl_Student) and these are displayed in the "Assignment-Specific" box. Note that in practice there are as many AssignmentImpl_Student/AssignmentInterface pairs as required by the particular assignment, but only one such pair is listed above. Several components are distributed to students (see the "*Starter Kit*" box above).

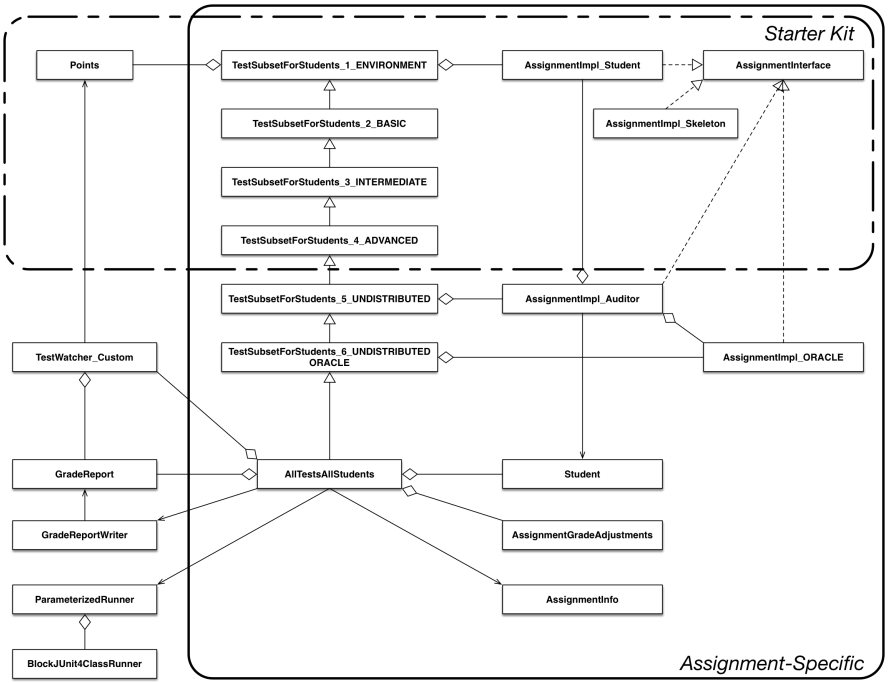


Figure 1: Component Diagram

In particular, notice that the students receive a subset of the test cases, the interface, a deficient implementation (AssignmentImpl_Skeleton), and the point values associated with each test method.

2.1.1 AssignmentInterface

This is the target interface that each student is being asked to implement.

2.1.2 AssignmentImpl_Student

This is the student submission.

2.1.3 AssignmentImpl_ORACLE

This is the standard for correct behavior and is written by the assignment author. This enables test cases to be written which pit student submissions directly against an oracle. Moreover, pairing an oracle with a simulator can

effectively produce test methods that run a submission through thousands of scenarios.

2.1.4 AssignmentImpl_Skeleton

This is a stripped-down version of AssignmentImpl_ORACLE. It is distributed to students to serve as a starting point for their implementation. In addition, the author has the ability to expose the private interface of the oracle implementation.

2.1.5 Student

The system only needs to know three pieces of information from the student:

```
public interface Student {
        public String getFirstName();
        public String getLastName();
        public String getFileNameSuffix();
}
```

An enumeration that implements Student will be defined for each section of a course and will contain one instance per student. A package and enumeration naming convention is used to keep section information separate (e.g., fall2018.CS101_01_STUDENT). Notice that the Student interface has the concept of file name suffix. This suffix is used to programmatically locate a student's submission via a class name calculation similar to:

ASSIGNMENT_NAME + "_" + student.getFileNameSuffix()

While this suffix typically is simply set to the student's last name, it can be used to handle last name collisions. For example, two students in the same section, M. Smith and T. Smith, can be told to use the file name suffixes, "MSmith", and "TSmith", respectively, for their submissions.

2.1.6 AssignmentImpl_Auditor

This class combines the student and oracle implementations in order to support elegant and structured student feedback, which is ultimately delivered via a PDF file. Such feedback is difficult to produce using only the student submission, especially in exceptional cases. As an example, the auditor's constructor uses Java reflection when attempting to construct an instance of the student's implementation and is informed when the student instance cannot be constructed. This allows for reasonable test method failure feedback (e.g., "student instance could not be constructed for parameters...") as opposed to reporting that the test method failed due to a NullPointerException (since the AssignmentImpl_Student instance was null).

2.1.7 AssignmentInfo

This class is used to house simple metadata about the assignment such as the course name (e.g., "CS101 Section 01"), semester ("Fall 2018"), assignment name ("HW 4"), etc.

2.1.8 Test Case Hierarchy

The test cases are segregated into several classes ranging from the one with suffix 1_ENVIRONMENT to the one with suffix 6_UNDISTRIBUTED_ORACLE. The test class with suffix 1_ENVIRONMENT checks whether the student's environment has assertions enabled, the correct Java version, and correct package names. The test class with suffix 6_UNDISTRIBUTED_ORACLE contains those test cases that pit the student's submission against an error-free implementation. This hierarchy is designed to encourage students to work through the more fundamental test cases first.

2.1.9 Points

Each method of the test case hierarchy is annotated with a point value (e.g., @Points(value=5)).

2.1.10 AssignmentGradeAdjustments

Student deductions resulting from late submissions, misnamed artifacts, etc., are recorded here. Student bonuses can also be recorded here. Ultimately, this information is incorporated into the PDF file given back to students.

2.1.11 GradeReport

An instance of this class is associated with each student and contains information about test case success and failure, as well as deductions and bonuses.

2.1.12 GradeReportWriter

An instance of this class is supplied with a desired ordering of the test methods that is used to organize the test feedback to the students. This class makes heavy use of the class PdfWriter from the third party code contained in iText-5.0.4.jar (or above).

2.1.13 TestWatcher_Custom

An instance of this class is used to monitor JUnit test method success and failure via an @Rule annotation. This enables a connection between the JUnit

framework and a GradeReport instance; test method successes and failures are recorded in the GradeReport instance on an event-driven basis.

2.1.14 ParameterizedRunner (and BlockJUnit4ClassRunner)

The ParameterizedRunner class extends org.junit.runners.Suite and, with a BlockJUnit4ClassRunner class, runs JUnit test cases in a "parameterized" fashion, where the parameters are determined by the test case. The system under consideration uses this to run a test case, which was designed to use a single student submission, against a whole course section's worth of student submissions.

2.1.15 AllTestsAllStudents

This class is the JUnit test case that the instructor will ultimately run and is effectively the cross product of all students (in a given section) with all of the test methods. It is a JUnit test case that is parameterized by all of the Student instances from a given section. Specifically, this test case contains a method that returns all these Student instances; this method is annotated with @Parameters. Moreover, AllTestsAllStudents is responsible for initializing all of the grade reports, installing a TestWatcher_Custom instance to monitor test method completion, and lastly, for producing PDF files containing test method feedback for students.

3 Student Workflow and Experience

From the student's perspective, software development proceeds according to the following steps:

1. Students work with instructor during class to develop interface.
2. Each student receives assignment handout.
3. Each student receives "Starter Kit" (See Figure 1) electronically.
4. Each student works on "Level 1" test cases, then "Level 2" test cases, etc.
5. At the submission deadline, the student has a good idea of what their submission grade will be and decides whether to submit on time.
6. Any student who does not submit on time can submit at the late deadline.
7. At the next class meeting, each student receives detailed feedback about the performance of their submission.

4 Instructor Workflow and Experience

From the instructor's perspective, assignment development proceeds according to the following steps:

1. Instructor comes up with idea for a new assignment.
2. Instructor designs interface and writes test cases including those which evaluate:
    - performance,
    - executable preconditions, and
    - correctness.
3. Instructor scaffolds test cases into various levels (e.g., 1:ENVIRONMENT, 2:BASIC, 3:INTERMEDIATE, 4:ADVANCED, 5:UNDISTRIBUTED).
4. Instructor writes the "Oracle" implementation(s).
5. Instructor writes the Auditor(s).
6. Instructor writes TestsAbstract for this assignment (this is mostly a copy, paste, tweak activity).
7. Instructor writes up handout.
8. Instructor produces Skeleton(s) by starting with the corresponding "Oracle" implementation(s) and deleting desired parts.
9. Instructor produces Starter Kit and provides to students electronically.
10. Instructor downloads submissions.
11. Instructor adds late information and other deductions to TestAbstract.

In support of the above, the instructor must implement the Student concept for each class section every semester. Also, the instructor must write the following classes once (these classes can be used unchanged for any new assignment):

- GradeReportWriter
- ParameterizedRunner
- Watchman

5 Conclusions

This test system provides an alternative to many other test systems and allows students to experience JUnit test cases in a hands-on way. Many students begin by reading test cases to understand the design and mechanics of the corresponding interfaces (test cases are part of the specification). Test cases can be scaffolded to encourage students to adopt a structured software development methodology. In particular, lower level test cases can be used to expose problems that might otherwise really confuse students if such a problem manifested itself in a higher level test case failure (e.g., assertions are not enabled!). Productive conversations with students often begin with the instructor request: "Show me the simplest test case that you're failing."

References

[1] Gerardo Canfora, Aniello Cimitile, Felix Garcia, Mario Piattini, and Corrado Aaron Visaggio. Evaluating advantages of test driven development: A controlled experiment with professionals. In *Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering*, ISESE '06, pages 364–371, New York, NY, USA, 2006. ACM.

[2] Stephen H. Edwards. Improving student performance by evaluating how well students test their own programs. *J. Educ. Resour. Comput.*, 3(3), September 2003.

[3] Stephen H. Edwards. Rethinking computer science education from a test-first perspective. In *Companion of the 18th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications*, OOPSLA '03, pages 148–155, New York, NY, USA, 2003. ACM.

[4] David Janzen and Hossein Saiedian. Test-driven learning in early programming courses. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '08, pages 532–536, New York, NY, USA, 2008. ACM.

[5] JUnit. About junit 4. `https://junit.org/junit4/`.

[6] M. M. Muller and O. Hagner. Experiment about test-first programming. *IEE Proceedings - Software*, 149(5):131–136, Oct 2002.

Cloud Computing and Running Your Code on Google Cloud*

Conference Workshop

Wesley Chun
Developer Advocate, Google Cloud

Cloud computing has taken over industry by storm, yet there aren't enough new college grads who know enough about it. This session begins with a vendor-agnostic, high-level overview of cloud computing including its three primary service levels. This is followed by an introduction to Google Cloud, its developer platforms, and which products serve at which service levels. Attendees will learn how to run applications on Google Cloud serverless platforms (in Python & JavaScript; other languages are supported) as well as hear about the teaching and research grants available to engineering (and non-engineering) faculty for use in the classroom or the lab. Whether you're a professor, researcher, edtech consultant, IT staff, TA grad student, or lecturer you'll know how to run code on Google's cloud and help enable the next-generation cloud-ready workforce.

Leveraging Technology to Scale Student Learning in Computer Science Courses*

Conference Workshop

Alynda Armstrong
Account Executive Turnitin/Gradescope

As demand for computer science programs continue to rise, challenges with scaling CS class processes have emerged. How can instructors assess hundreds of students effectively, efficiently, and fairly? How can they leverage the grading process to drive academic success? Learn how instructors at over 600 universities use Gradescope[1] to dramatically reduce the pain and time associated with grading all types of student work, including exams, homework, and programming projects.

[1]Gradescope is now part of Turnitin! - Gradescope is an assessment platform that optimizes grading workflows for STEM, Economics, and Business courses.

CyberReady StL Curriculum: Tutorial, Best Practices, and Results from Initial Deployment*

Conference Tutorial

Rebecca Dohrman[1], Paul Gross[2], Steve Coxon[3], Dustin Nadler[4], Chris Sellers[5], Christi Demuri[6], Robyn Ray[5]
[1]Department of Communication, Maryville University
rldohrman@maryville.edu
[2]Gross Code and Education
[3]Department of Education, Maryville University
[4]Department of Pyschology, Maryville University
[5] Jennings High School
[6] Ritenour High School
St. Louis, MO 63141

This workshop will walk participants through the development and recent deployment of the CyberReady StL curriculum which is built on the Raspberry Pi platform to introduce students to the basics of coding in Python, the Raspberry Pis platform (with SenseHat), and networking in order to help students be more cyber ready and to prepare them for subsequent computing curricula (i.e. CyberPatriot). The tutorial will be presented by a team of researchers from Maryville University, a computing expert who was on development team for the curriculum, and three educators who deployed the curriculum in Fall 2018. The team will talk through results from the pre and post tests about attitudes related to computing as well as cyber readiness skill.

Preparing for the New ABET-CAC Computing and Cybersecurity Criteria*

Panel Discussion

Tim McGuire[1], Rob Byrd[2], Deborah Dunn[3]
[1]Department of Computer Science
Sam Houston State University
Huntsville, TX 77341
mcguire@shsu.edu
[2]School of Information Technology and Computing
Abilene Christian University
Abilene, TX 79699
rbyrd@acu.edu
[3]Department of Computer Science
Stephen F. Austin State University
Nacogdoches, TX 79699
ddunn@sfasu.edu

Multiple ABET Commissions are modifying their criteria in significant ways. With input from the computing community, ABET's Computing Accreditation Commission has updated the Criteria for Accrediting Computing Programs to reduce the assessment burden and take into account the CS2013 curricular guidelines. These revised criteria will be fully effective for site visits from 2019 onwards. Programs must plan for these changes, especially how they affect both curriculum and assessment. Additionally, criteria for Cybersecurity programs are now in the final stages of approval. What do these changes mean for your computing program(s)? This panel session is an effort to inform computing faculty about the recent changes made to the criteria and how these changes may potentially impact current assessment processes and curriculum implementations. The panelists are ABET Commissioners and they will provide an overview of the changes and their rationale. Together, they will engage the audience in discussion, allowing time for interaction and clarifying questions.