Learn2Mine: Data Science Practice and Education through Gameful Experiences

Clayton A. Turner, Jacob L. Dierksheide, and Paul E. Anderson

***Abstract*—We introduce Learn2Mine, an education and analysis platform that integrates state-of-the-art data mining tools with effective feedback and training mechanisms in order to lower the barrier for domain experts and computer scientists to learn data science. Data science is the combination of statistical and computer science techniques in order to extract meaningful information from domain-specific datasets. Learn2Mine is a platform where students learn and practice techniques commonly used by data scientists. The Learn2Mine platform is a novel environment for teaching data science without requiring prerequisite knowledge, and with the idea that all knowledge bases can be enhanced by data science. It applies the principles of gamification, making the learning process more engaging and rewarding. Learn2Mine has been piloted by undergraduates, which, through the ability to retry lessons and receive instant feedback, has allowed them to engage in more sophisticated data science concepts than previous semesters. The next step for Learn2Mine, which will be continuously extended with new algorithms and lessons and completely open to the public beginning January 2014 (http://learn2mine.appspot.com), is the completion of an extension framework giving international institutions and organizations of higher learning the ability to create their own lessons for students to perform.**

***Index Terms*—Data mining, data science, gamification, education.**

I. Introduction

The development of cross-disciplinary fields, such as data science, has changed scientific inquiry and business analytics in many respects. But it is still often the case that collaborations between computer scientists and domain experts are limited by the disparity in their skill sets. Attempting to bridge this gap, computer scientists have developed domain specific software and algorithms, but the use of which typically requires significant computing expertise. Modern programming languages have allowed developers to minimize platform dependencies, and attempts have been made to connect the domain expert with informatics systems (e.g., Galaxy [1]-[3], Weka [4], RapidMiner [5], and Taverna [6]); however, these applications often require additional dependencies for specific domains, lack an intuitive mechanism for feedback and training, require extensive computer science expertise, and require dedicated computing resources for large datasets.

Manuscript received November 15, 2013; revised December 27, 2013. This work was funded by the SC EPSCoR/IDeA GEAR: Research Experiences (PI: P. Anderson).

C. A. Turner, J. L. Dierksheide, and P. E. Anderson are with the College of Charleston, Charleston, SC 29424 USA (email: caturner3@g.cofc.edu, jldierks@g.cofc.edu, andersonp@cs.cofc.edu).

The amount of data collected by, available to, and of interest to the scientific and business communities has resulted in an increased demand for individuals with the skills necessary to analyze large data sets (i.e., big data). While undergraduate and graduate programs have begun to emerge in the field of data science [7]-[10], the number of open positions for computer scientists and applied mathematicians with the training and curiosity to make discoveries in the world of big data has exponentially increased over the last few years. At present, the community lacks engaging data science and analytics software aimed at teaching aspiring students to explore and find patterns in large datasets.

Learn2Mine is an integrated learning environment to introduce students to data mining, computer science, statistics, and data science. Learn2Mine increases the accessibility of data science and analytics to a diverse group of students and practitioners by reducing the need for local computing resources, removing programming as a prerequisite, and engaging the students via gameful experiences. Learn2Mine is a cloudbased application intended for use by academics, industry professionals, and students interested in the field of data science and its related sub-disciplines (data mining, statistics, etc.). It offers lessons with rewards, in the form of badges and a progressively-built skill tree that includes many common algorithms, such as K-Nearest Neighbors Classification, K-Means Clustering, Neural Networks, and Market Basket Analysis, among others. Additionally, Learn2Mine also strives to introduce students to the R programming language.21‘Learn2Mine provides easy access to tools for these techniques that can be used for user-driven projects or research.

II. Motivation and Related Work

A. Emergence of Data Science

Data science, a conglomeration of applied mathematics, statistics, computer science, and artificial intelligence, aims to glean new knowledge from existing information. Typically, data science is applied to a specific field of domain knowledge - bioinformaticians, for example, tend to work with biological datasets, such as gene sequence data. A common bioinformatics task is the identification of differentially expressed genes and patterns of genes that correlate to a treatment group, and while the algorithms used to accomplish this task are applicable to other areas, significant domain expertise is still required for this and other data science problems.

The primary focus of data science is to manage and make sense of large and complex datasets (i.e., big data) which, as mentioned previously, tend to come with prerequisite domain knowledge. Whether in the areas of database management, cloud computing, knowledge discovery, data mining, computer vision, or language processing, this domain expertise is an inescapable, necessary component. This concept is one of the main ideas behind the design of Learn2Mine, and requires novel ways to introduce data science to domain experts with limited prerequisite technical background.

B. Learning with Game

The key to Learn2Mine's implementation is its reliance on the notion of "gamification." A concept with growing support in the education community, the goal of gamification practice is to invoke the same level of engagement and enjoyment that can be gleaned from gaming in a way that can be focused on other problem solving contexts. Gamification does not necessitate a full-fledged game, but rather is the application of game mechanics in non-game settings [11], [12]. Implementing gamification has been shown to increase user involvement with an application [13]. It is important to note the difference between gamification and so-called "edutainment", which are simply games that have a bonus educational goal. Concepts commonly implemented by gamified applications include:

- Reward Schedules[14]-[17]
- Constant Feedback [14], [16]
- Reputation [14]
- Advanced User Paths [14]
- Content Unlocks [14], [16]
- Collaboration [14], [16], [17]

Additionally, gamification grants the user the permission to fail, which can be a real boon in the learning process and encourages a trial and error approach to material that pulls on their own ability to problem solve and creates a more engaging experience[18], [17]. An example of a system with gameful experiences for the domain of bioinformatics is the platform Rosalind [19].

Learn2Mine makes use of many of these concepts in its form of gamification. Users are given immediate feedback when running tools by providing visual results. Users unlock the ability to earn a "Learned" status on more lessons with each one they complete, and have the option of continuing already completed lessons to earn "Mastery" status by learning R programming through manipulation of provided code or the creation of their own. A running leaderboard for some lessons also gives users a reputation among their peers and encourages competition amongst them, a technique shown to aid in learning and technique development [15], [20].

III. LEARN2MINE DESIGN

Learn2Mine is designed to provide a rewarding educational experience through the application of gamification techniques. Feedback and training loops capitalize on this gamification in order to minimize the divide between domain expert and algorithm developer. In order to bridge that gap, Learn2Mine has been developed to communicate between three different environments: The Google App Engine Framework, Galaxy, and RStudio. Google App Engine is an implementation of a platform as a service (PaaS) cloud computing for hosting web applications. This allows Learn2Mine to operate through a cloud-based interface. Additionally, this gives Learn2Mine a NoSQL database to work in the backend. This database is built for storing user credentials and progress. Google App Engine is flexible in that it supports many languages for populating the application including Python, Java, and Go. This distributed architecture allows other institutions to implement their own lessons and lesson verification, which can then be incorporated into the global learning framework.

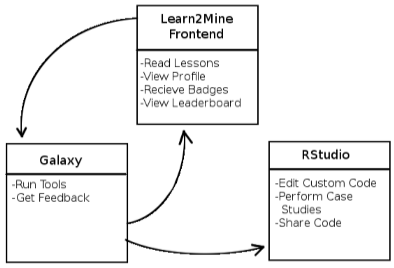

Fig. 1. The Communications between the 3 branches of Learn2Mine.

Galaxy [1]-[3] is an extensible, open source project which works as workflow management system for conducting algorithms in a local or cloud-based environment. It uses jobs to conduct these algorithms, which are typically a Python wrapper, masked by a user-friendly graphical interface. The Python files that run, in turn, can be used to call scripts in other languages, a capability that Learn2Mine utilizes to run R code that executes state-of-the-art data mining algorithms, while outputting dynamic results in HTML.

RStudio is a browser-based IDE for creating, modifying, and executing R code. RStudio has built-in mechanisms for allowing users to specialize their R installation. Users can install third-party packages, conduct any visualization task, and customize the layout of RStudio itself.

Fig. 1 shows the dynamic that exists between the 3 and the abilities of each. Through the login email and session provided by the Learn2Mine Frontend, users create their key in the Galaxy working environment. Galaxy then uses this key to send the users' lesson results back to Learn2Mine (this process is described in further detail in later sections). Additionally, Galaxy can run the custom scripts written by users in RStudio. This architecture has been designed for collaboration, where other institutions have the ability to interact with the Learn2Mine application programming interface (API) and issue and validate their own badges that can be incorporated into the skill tree.

A. Virtual Portfolio

Learn2Mine allows users to build up a virtual portfolio, which is a culmination of many different gameful experiences. Users will be able to progress through a skill tree, earn varying levels of badges representing analagous achievements, and compete for high scores on leaderboards. This virtual portfolio is hosted on Google App Engine. The implementation used for Learn2Mine uses Python and the webapp2 framework. The webapp2 framework allows Learn2Mine to use Python code to populate HTML pages and to interact with the backend NoSQL database. This database stores all the information for each user's virtual portfolio.

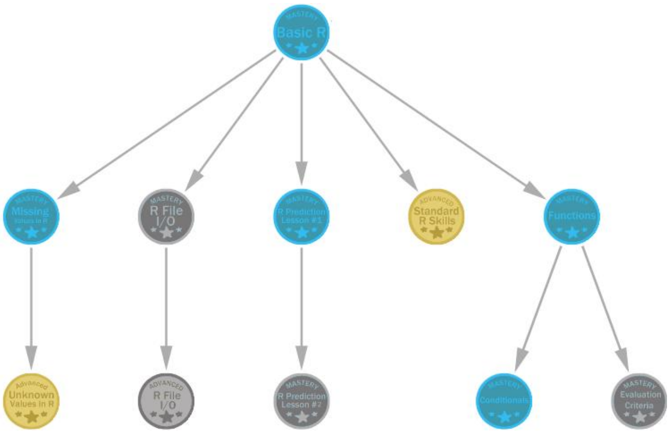

Fig. 2. An example of a user's R subtree.

The skill tree provided to each user, subtrees shown in Fig. 2 and Fig. 3, is integral to the gameful learning process promoted by Learn2Mine. The skill tree acts as a window into what a user has learned, mastered, and even what they will soon learn. The color of skills within the skill tree reflects the difficulty of a lesson.

In Fig. 2 and Fig. 3 there are gray, green, blue, and gold skills. A skill is gray if the user has not completed the associated lesson for the skill. In Fig. 2, the R File I/O lessons have not been completed so they show up as this gray. The simplest lessons are basic learning lessons and they appear green upon completion. In Fig. 3, the first K-Nearest Neighbors lesson, the Partial Least Squares Regression lesson, and Neural Networks lesson have all been completed at the basic learning level. Typically, there is a follow-up lesson being created or already residing in the skill tree for these lessons as to allow the user to achieve mastery or advanced mastery. Blue skills represent more difficult lessons that were completed and they are referred to as mastery lessons. In Fig. 2 there are a multitude of skills that have been mastered, such as the Functions lesson and Conditionals lesson. Lastly, there are gold lessons that represent an advanced understanding of specific topics. In Fig. 2 there are two of these lessons: Unknown Values in R and Standard R.

The skill tree assumes a hierarchy of skills, however Learn2Mine does not hold users to this hierarchy, meaning that a user could complete the advanced lessons lower in the skill tree without ever completing the more basic versions of the lessons. This allows users that are already familiar with certain aspects of data science to learn exactly what they desire to learn. So users are allowed to traverse the tree in any manner, giving users the all-important feeling of free will within Learn2Mine.

Further advances in these data science skills can be shown in the Learn2Mine Leaderboards. A select set of lessons have scores that relate to the techniques-at-hand. For example, the K-Nearest Neighbor Classification lesson scores a user based upon a percentage of correct classifications in the testing set provided in the lesson. More elaboration on how users improve their score in Section III-D. The leaderboards give users a way to build confidence in their skills and incite healthy competition with their peers.

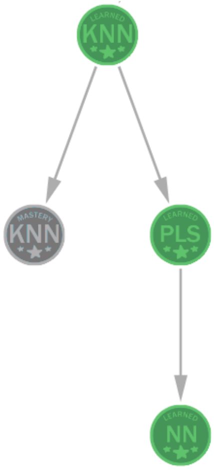

Fig. 3. An example of a user's classification subtree.

B. Open Badges

After completing lessons in Learn2Mine, users can earn badges that provide incentives and mark their progress. This takes advantage of Mozilla's OpenBadges in order to easily issue and distribute badges. After completing a lesson, a user is issued a corresponding badge for that lesson and ranking. Badges are saved as a JSON file on the Learn2Mine server, a static link to which is provided to Mozilla for them to keep as a reference. Badges are presented to the user upon visiting the Home page, and once accepted can be viewed in the user's Mozilla Backpack, and can be displayed on their Learn2Mine homepage by creating an appropriate group in their backpack. The badges can also be displayed on other platforms, such as LinkedIn.

As seen previously, these badges are the nodes in a user's skill tree, so the coloring of the badges follows all of the aforementioned rules with learning, mastery, and advanced lessons. To better view a set of these badges, see Fig. 4. Fig. 4 shows a side-by-side comparison of the "Learned" and "Mastery" badges for the "Stock Market Case Study" lesson. Once awarded to the users, badges can be showcased additionally on social media and r ́esum ́es, according to Mozilla Open Badges specifications, in order to publicly show proficiency or mastery of data science skills.

(a) Learned Badge (b) Mastery Badge
Fig. 4. Stock case study badge style comparison.

C. Lessons and Data Mining on the Web

The execution of algorithms within Learn2Mine resides in the Galaxy environment. This is where users will be running the tools we have provided them and viewing the corresponding analyses. Users can come here and run any algorithm, whether it is a simple data scaling or as complicated as running a customized Neural Network. In order for users to perform lessons, however, they have to set up the communication from Galaxy to Learn2Mine.

D. Customizing and Designing R Code

Once Learn2Mine users have developed a familiarity with the algorithms, they are given the opportunity to take their experimentation with data science further by trying for lesson "mastery" through the manipulation of the R code that runs each tool in Galaxy. Each Learn2Mine user is given access to their own RStudio account on our server, with a personal home directory within which they can write and save R scripts to be run by Galaxy. Users are also given access to the code each Learn2Mine tool uses in the form of a git repository that they can pull in directly to their RStudio workspace. Advanced lessons are designed to make users tamper with this code in the hopes of optimizing their results. Additionally, users are encouraged to utilize this RStudio workspace in order to compose the code for the R programming lessons.

For example, in their intended first mastery lesson, users are asked to weight the features of a previously used dataset to improve the success rate of a K-Nearest Neighbors Classification run. Implementing gamification often requires a balancing act between too many and too few extrinsic rewards and feedback. Holding the user's hand too much will make them feel like they have lost autonomy, but too little guidance devalues their experience [21]. Mastery lessons on Learn2Mine provide a concrete goal, but give users the freedom to test and learn through RStudio on their own terms. Approaches suggested to them through Learn2Mine include hill-climbing algorithms, Simulated Annealing, and genetic algorithms, but, ultimately, users are free to experiment with whichever optimization techniques they desire.

The Learn2Mine Galaxy environment also provides users the additional freedom to run any other scripts they want to make themselves in the form of "Custom Tools". We have created multiple custom tools within Galaxy so that the user can pick which tool caters to their custom, specialized code. The tools have arguments that can be passed to the R script and the tools have varying outputs, giving the user as much flexibility as needed. This gives the users an open-ended experience with our application, and between the RStudio and Galaxy systems have all the tools at their disposal to sandbox any other data science projects they wish to undertake.

E. Security and Deployment

Currently, Learn2Mine's Galaxy and RStudio servers are deployed on a virtual machine specially designated to run them. Starting in January of 2014, we will release the Learn2Mine API and the GAE code base to the open source community. As a system that encourages users to run their own personalized scripts, there are some inherent security concerns in Learn2Mine. To combat this, we are in the process of migrating the Galaxy and RStudio components from a traditional virtual machine architecture to the lightweight virtualization solution known as ZeroVM. This will allow us to provide each individual with their own container, thus greatly improving security. This is possible due to ZeroVMs ability to efficiently virtualize applications and not machines.

When a user first goes to Learn2Mine, they are assigned a unique key, referred to as a "session" on the backend. We have created a Galaxy tool called "Create Learn2Mine Key" which is important for the synchronization of the 3 branches of Learn2Mine. Users provide their email address to this tool, which then remotely retrieves their "session" from the Google App Engine Datastore. When a user utilizes this tool, then Learn2Mine, in the background, creates an account for that user in RStudio complete with a personal working directory using their Learn2Mine email and session as the username and password, respectively.

IV. Sample Lesson

Learn2Mine allows users to learn and practice data science through a variety of structured lessons in commonly used algorithms. A runthrough of "K-Nearest Neighbor", a typical Learn2Mine lesson, would look as follows:

A. The User Logs into Learn2Mine

When a user first visits the Learn2Mine front end website, they are asked to log in with a Google account. At the point of successful login, a User object is created in the Google Datastore, with fields that include their status in each lesson, and the unique 21-digit session value.

B. The User Visits the Lessons Page

A user may then proceed to the "Lessons" page on the Learn2Mine website, and select the "KNearest Neighbors" link from the panel of options on the right side of the page. After reading the associated explanation of the algorithm and some examples of its use, the user is presented with instruction for running their own K-Nearest Neighbor test on 2 provided datasets, Diabetes Train and Diabetes Test.

C. The User Proceeds to Learn2Mine's Galaxy Server

Following the "Try Our First KNN Lesson" link on that same page, users are then brought to Learn2Mine's galaxy server. Instructions for using Galaxy are provided on the Lessons page, as well as in the form of an introductory data upload and submission lesson to serve as a tutorial.

D. The User Uploads Required Datasets

At this point users can upload the provided datasets using the "Upload Data" tool that is built in to Galaxy. For the K-Nearest Neighbor lesson they are provided with a DiabetesTrain.csv and a DiabetesTest.csv to upload.

E. The User Runs the K-Nearest Neighbor Tool

Once both datasets have been uploaded they are ready to run the K-Nearest Neighbor test, located in the Learn2Mine Toolset. To use the tool, users provide the "Training File", "Testing File", and a *K*-value of their choosing. Pressing "Execute" will run the tool and provide them with two outputs in their Galaxy history: an html file (which contains formatted output from their tool including a graphical representation of the classifications) and csv results (which is how their test will be graded). An example of the HTML output from the *K*-Nearest Neighbors lesson can be seen in Fig. 5. In this example, the Diabetes Training data and Testing data were evaluated with K=10. In order to allow for viewable results, we conduct principal component analysis in order to reduce the dimensionality of the data that the user is utilizing so a cross-section of the data can be viewed in a formatted graph.

F. The User Sends Results Back to Galaxy

Upon test completion, the user is ready to have their results graded. Here, the "Create Learn2Mine Key" tool comes into play again. Aside from RStudio user creation, this key is how Galaxy links a lesson submission to a user. Users enter the output of this key tool as input for the "Submit to Learn2Mine" tool, along with their results, and the lesson they want it graded as, in this case "*K*-Nearest Neighbor". The csv file is then evaluated based on the criteria for the lesson they are submitting it as, and the outcome is sent via a POST request to the Learn2Mine website, where the appropriate user is queried from the Datastore by the "Key" they submitted, and the virtual portfolio can be updated appropriately.

G. The User Receives Feedback

Upon successful submission, users can view the formatted feedback of their lesson results in the "Learn2Mine Submission..." history item, which tells them whether or not they successfully met the lesson's criteria and gives hints if not. If they have achieved a "Learned" or "Mastered" ranking, they can proceed back to the Learn2Mine frontend website to observe the changes to their skill tree as well as accept any badges due to them.

V. Discussion

A. Implications for Teaching

Learn2Mine is currently being utilized in Dr. Paul Anderson's Data Science 101 class for the fall semester of 2013.

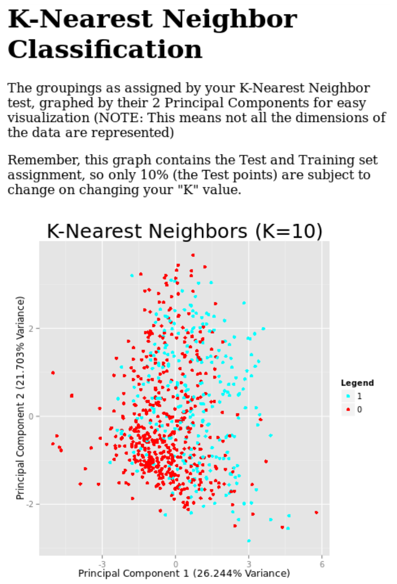

Fig. 5. Sample HTML output of KNN lesson.

During this semester, the system has allowed students to learn and practice the methods taught by the class in ways never afforded to those of previous semesters. These students are able to get hands-on experience programming even with little to no background. For example, students with no prior programming skills have written code in the R programming language to predict the stock market and classify biological samples from experimental data using algorithms such as Random Forests and Support Vector Machines. Prior semesters have focused on using boxed or canned software solutions to process the data, such as RapidMiner [5] or Weka [4]. Their progress through these applications have been driven by incremental lessons and instant feedback.

By building upon research into gamified learning, the forgiveness of Learn2Mine's skill trees, instant feedback, and unlimited retries (or lives) allows them to learn in a reinforced and productive manner through trial-and-error[14], [16]. We believe that Learn2Mine is an effective resource for those looking to introduce themselves to the field of data science in either a classroom guided or independent study environment.

Students in the course are also actively contributing valuable feedback for continual improvement that is being incorporated before opening the Learn2Mine system to the wider community, which is planned for January 2014.

B. Implications for Data Scientists

While Learn2Mine has the potential to impact how data science is introduced and taught, it also provides a full feature environment for academics and enterprise users. RStudio's ability to let users dynamically write and test code, as well as its integration with Git, facilitates a collaborative atmosphere among users. The ability of an individual user to write and integrate custom R scripts in to the Galaxy environment provides a novel way to increase the toolset available to an established community of users. It also provides a flexible way to incorporate the wide variety of preexisting R packages.

C. Future Development

New features will continue to be added to Learn2Mine as the classroom deployment progresses. Upcoming plans include the addition of greater security measures to futher isolate users, and a more seamless method of authentication for users between Learn2Mine, Galaxy, and RStudio. The platform will be available to the general public in January of 2014, and the API documention will be published to allow third party institutions the ability to customize their experience and extend the platform.

The easy scalability of Learn2Mine through the creation of Galaxy tools means that there is significant potential for the application's growth if it proves to be successful. New lessons can be invented and integrated with relative ease, and would also allow further development to be geared toward building a more social aspect of the application to encourage a more developed Learn2Mine community.

Acknowledgment

We would also like to thank College of Charleston for the use of their facilities during the period of this research.

References

[1] D. Blankenberg, G. V. Kuster, N. Coraor, G. Ananda, R. Lazarus, M. Mangan, A. Nekrutenko, and J. Taylor, "Galaxy: A web-based genome analysis tool for experimentalists," *Current protocols in molecular biology*, ch. 19, 2001.

[2] B. Giardine, C. Riemer, R. C. Hardison, R. Burhans, L. Elnitski, P. Shah, Y. Zhang, D. Blankenberg, I. Albert, J. Taylor, W. Miller, W James Kent, and A. Nekrutenko, "Galaxy: A platform for interactive large-scale genome analysis," *Genome Research*, vol. 15, pp. 1451–1455, 2005.

[3] J. Goecks, A. Nekrutenko, and J. Taylor, "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences," *Genome Biology*, vol. 11:R86, 2010.

[4] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The WEKA data mining software," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.

[5] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler, "YALE: Rapid prototyping for complex data mining tasks," in *Proc. the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, vol. 2006, pp. 935–940, 2006.

[6] K. Wolstencroft, R. Haines, D. Fellows, A. Williams, D. Withers, S. Owne, S. Soiland-Reyes, I. Dunlop, A. Nenadic, P. Fisher, J. Bhagat, K. Belhajjame, F. Bacall, A. Hardisty, A. N. de la Hidalga, M. P. Blacazar Vargas, S. Sufi, and C. Goble, "The Taverna workflow suite: designing and executing workflows of web services on the desktop, web or in the cloud," *Nucleic Acids Research*.

[7] P. Anderson. Data Science Program. [Online]. Available: http://datascience.cofc.edu/

[8] S. Argamon. Data Science @ IIT. [Online]. Available: http://www.iit.edu/csl/cs/programs/undergrad/datascienceundergrad.shtml.

[9] C. Cheshire. UC Berkeley School of Information: Data Science. [Online]. Available: http://www.ischool.berkeley.edu/research/datascience.

[10] P. L. Dubois. UNC Charlotte: Data Science and Business Analytics Initiative. [Online]. Available: https://dsba.uncc.edu/academic-programs

[11] S. Deterding, M. Sicart, L. Nacke, K. O'Hara, and D. Dixon, "Gamification using game-design elements in non-gaming contexts," in *Proc. the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA '11*, pp. 2425–2428, New York, New York, USA, ACM Press, 2011.

[12] H. Gerber. (2012). Can Education be Gamified: Examining Gamification, Education, and the Future. [Online]. Available: http://shsu.academia.edu/HannahGerber/White-Papers

[13] J. Thom, D. R. Millen, J. Dimicco, and R. Street, "Removing Gamification from an Enterprise SNS," *Incentives*, pp. 1067–1070, Seattle, WA, 2012.

[14] N. Direkova. (2012). Game On: 16 Game Mechanics for User Engagement. [Online]. Available: http://www.slideshare.net/gzicherm/nadya-direkova-game-on-16-design-patterns-for-user-engagement.

[15] I. Glover. (2003). Play As You Learn: Gamification as a Technique for Motivating Learners. [Online]. Available: http://www.editlib.org/p/112246?nl

[16] C. I. Muntean, "Raising engagement in e-learning through gamification," in *Proc. The 6th International Conference on Virtual Learning ICVL 2011*, pp. 323–329, 2011.

[17] F. Fui-hoon Nah, V. R. Telaprolu, S. Rallapalli, and P. Venkata. Gamification of Education Using Computer Games Background: Gamification and Its Application to Education. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-39226-9 12#page-1, 2013.

[18] K. Kapp, *The Gamification of Learning and Instruction*, Pfeiffer, San Francisco, CA, 2012.

[19] N. Vyahhi, P. Compeau, A. Balandin, A. Kladov, E. Sosa, M. Dvorkin, S. Alexandrova, and M. Rayko. Rosalind. [Online]. Available: http://rosalind.info/

[20] A. Dom ́inguez, J. Saenz-de Navarrete, L. De-Marcos, L. Fern ́andez-Sanz, C. Pag ́es, and J. J. Mart ́ınez-Herr ́aiz, "Gamifying learning experiences: Practical implications and outcomes," *Computers & Education*, vol. 63, pp. 380–392, 2012.

[21] N. Asaj, K. Bastian, M. Poguntke, F. Schaub, M. Weber, D. Ritter, D. Deuter, and F. Groh, "Gamification: state of the art definition and utilization," *Research Trends in Media Informatics*, pp. 39–45, Ulm, Germany, 2012.

Clayton A. Turner was born in Dillon, South Carolina (USA) on September 11, 1992. He is an undergraduate student at the College of Charleston. He plans on completing his bachelor's degree in data science and computer science in May 2014 at the College of Charleston. He has worked for the College of Charleston in the past as an Undergraduate Researcher. His interests are grounded in the development and tuning of algorithms within the realm of data science along with the refining and utilization of existing techniques. Mr. Turner is a member of IEEE.

Jacob L. Dierksheide was born in Myrtle Beach, South Caroline (USA) on May 3, 1992. He is currently enrolled at the College of Charleston, and will complete his bachelors degree in computer science in May of 2014. He has worked for the Medical University of South Carolina developing an android-based survey to store and track patient information for their Family Medicine Center. He has also worked for the College of Charleston as a Teaching Assistant in their Programming I Lab, where he assisted in lab administration and tutored students in programming.

Paul E. Anderson graduated in 2004 from Wright State University with a B.S. degree in computer engineering. He received his masters in computer science in 2006 and his Ph.D. in computer science & engineering in June 2010. His masters work was directed at developing a framework for analyzing chemical modification and limited proteolysis experimental data used for high confidence protein structure prediction. After receiving his masters, his dissertation research switched focus to developing algorithms for the field of metabolomics in collaboration with the Magnetic Resonance Lab at Wright State University (Department of Biochemistry and Molecular Biology, School of Medicine). This collaboration involved the development of novel algorithms for spectroscopic quantification and biomarker pattern recognition, which led to the development of a cloud-enabled platform for data mining and spectral processing. Throughout his graduate work, Paul was also intricately involved in the implementation of the Wright State University model for engineering mathematics education. After graduation, Paul worked as a Bioinformatics Research Scientist for the Air Force Research Laboratory (AFRL) under the Consortium of Universities Research Fellows Program. While at the AFRL, he worked on a variety of research projects, including a computational model of the human immune system, a proteomics study of mustard sulfur exposure, feature selection techniques for quantitative structure-activity relationship (QSAR), and metabolomics studies of human fatigue and performance. At present, Paul is an assistant professor in the Computer Science Department at the College of Charleston. He is also the Director of the Data Science Program and the Faculty Advisor to the ACM Club.