Collaborative Learning in Cloud-based Virtual Computer Labs

Xiaolin Hu
Dept. of Computer Science
Georiga State University
Atlatna, GA, USA
xhu@gsu.edu

Hai Le
Dept. of Computer Science
Georiga State University
Atlatna, GA, USA
haihle181@yahoo.com

Anu G. Bourgeois
Dept. of Computer Science
Georiga State University
Atlatna, GA, USA
anu@cs.gsu.edu

Yi Pan
Dept. of Computer Science
Georiga State University
Atlatna, GA, USA
yipan@gsu.edu

***Abstract*—This Innovative Practice Work-In-Progress paper presents a collaborative virtual computer lab (CVCL) environment to support collaborative learning in cloud-based virtual computer labs. With advances of cloud computing and virtualization technologies, a new paradigm of virtual computer labs has emerged, where students carry out labs on virtualized resources remotely through the Internet. Virtual computer labs bring advantages, such as anywhere, anytime, on-demand access of specialized software and hardware. However, with current implementations, it also makes it difficult for students to collaborate, due to the fact that students are assigned separated virtual working spaces in a remote-accessing environment and there is a lack of support for sharing and collaboration. To address this issue, we develop a CVCL environment that allows students to reserve virtual computers labs with multiple participants and support remote real-time collaboration among the participants during a lab. The CVCL environment will implement several well-defined collaborative lab models, including shared remote collaboration, virtual study room, and virtual tutoring center. This paper describes the overall architecture and main features of the CVCL environment and shows preliminary results.**

Keywords—Collaborative learning, virtual computer lab, cloud computing, virtualization

I. Introduction

Computer labs where students work on assignments on computers using specialized software and/or hardware play a critical role in computing education and in STEM education in general. Traditionally these computer labs are carried out in computer centers on campus due to the need of specialized software and/or dedicated hardware. With advances of cloud computing and virtualization technologies, a new paradigm of virtual computer lab has emerged where students carry out labs on virtualized resources (such as high-end hardware or licensed software) remotely through the Internet. A virtual computer lab takes programs running on a university's hardware and beams the images to any computer desktop across the internet, giving students the ability to create and save work as though the programs were running on their own computers [1, 2, 3]. This overcomes the issue that a student has to be physically present in a campus computer laboratory in order to use the specialized software or hardware. The virtual computer lab also makes it possible for a large number of students to access a lab without being constrained by the space of a physical computer laboratory. It can significantly increase the accessibility of computer lab software and hardware, and can potentially revolutionize the manner in which computer labs are managed in educational institutions [2, 4]. Due to these advantages, there have been various efforts developing different types of virtual computer lab systems, such as cloud-based virtual laboratory for network security education [5, 6], cloud based systems lab for student remote access [7], virtual lab solutions for online cyber security education [8], and the Virtual Computing Lab (VCL) platform [9, 2, 3] that has been implemented in many universities.

Virtual computer labs bring advantages such as anywhere, anytime, on-demand access of specialized software and hardware. However, it also makes it difficult for students to collaborate, due to the fact that students are assigned separated virtual working spaces in a remote-accessing environment and there is a lack of support for sharing and collaboration in the current systems. This compares to computer labs in the physical world where students naturally feel the presence of their peers and can easily work together and help each other if needed. Collaborative labs help students to learn through experience, leverage the perceptions of their learning partners, and form their own opinions through social constructivism [10]. They have been consistently shown to have positive effects on student achievement, self-esteem, and attitude toward learning [11], [12]. This important feature of learning, however, is largely lost in the current implementations of virtual computer labs.

The importance of supporting collaboration in an online environment has long been recognized. One approach to support collaboration is to develop specialized software such as Google Docs that supports collaborative editing and Saros [13] that supports distributed pair programming. The limitation of this approach is that it depends on the availability of the collaborative software and thus cannot be generalized to software that does not provide collaboration support (such as Matlab). A more flexible approach is to support collaboration through desktop sharing technologies. Desktop sharing allows remote access and real-time collaboration and has the advantage of working with any software that runs on the "desktop". This makes it a desirable solution for supporting collaborative learning in virtual computer labs, where the types of software are open-ended. While many third-party desktop sharing tools exist, they are not originally designed for pedagogical purposes and do not provide integrated support for virtual computer labs [14, 15]. To effectively support collaborative learning in virtual

The work is partially supported by the National Science Foundation under award: DUE-1712384.

computer labs, we believe an integrated collaborative learning environment with native support for different collaborative learning models is needed.

Towards the above-mentioned goal, this paper presents a collaborative virtual computer lab (CVCL) environment to support collaborative learning in virtual computer labs. The developed environment allows students to reserve virtual computers labs with multiple participants and support remote real-time collaboration among the participants during a lab. By real-time collaboration, it means that students can not only talk and chat in real time during a lab, but also have the capability of remote desktop sharing and control so that multiple team members can work on the same "desktop" for a group project. This capability of desktop sharing and control is extremely useful in computer labs so that all team members can be "hands-on" at the same time during a lab. It also makes it effective for team members to help each other when problems arise. For example, a student stuck on a Matlab coding problem for a long time would really appreciate if another team member can touch and debug the exact code instead of just suggesting changes through voice or messaging. Due to this reason, we have integrated the function of remote desktop sharing and control into the CVCL environment. Furthermore, to support different forms of collaboration in both formal and informal settings, the CVCL environment will implement several well-defined collaborative lab models, including shared remote collaboration, virtual study room, and virtual tutoring center. The novelty of this work lies in the fact that it focuses on the aspect of collaborative learning in virtual computer labs and develops a new environment that allows students to collaborate in similar ways as they do in physical computer labs (e.g., sharing and controlling the same desktop). The developed environment is not tied to any specific system and can work with open-ended virtual computer lab systems.

The remainder of the paper is organized as follows. Section II presents related work. Section III describes the overall architecture and the three collaborative lab models of the CVCL environment. Section IV presents CVCL implementation and shows preliminary results. Section V concludes this paper.

II. RELATED WORK

The new paradigm of virtual computer lab is enabled by cloud computing technologies. In cloud computing, users acquire services that meet specific needs, such as accessing to application software or using a particular computing platform, without taking on responsibilities for the underlying structure [16, 17]. The computing resources that provide the services come to the user over the Internet. Cloud computing supports computer labs that overcome many of the limitations experienced with the traditional computer lab and student ownership models [4]. A user needs only a modest-sized computer, while the faster processing speed and larger memory capacity required for advanced software can be provided via the cloud. Many support issues are made easier because of the central control of the computing resource in the cloud [4].

Enabled by the cloud computing technologies, there are various efforts developing cloud-based virtual computer lab systems [18]. A cloud-based virtual laboratory education platform called V-Lab was developed to support network security education [5, 6]. The virtual laboratory platform provides an environment for hands-on experiments using virtualization technologies and allows students to remotely control virtual machines (VMs) and perform experimental tasks. A cloud-based solution for teaching computer networks was described in [19], where VMWare software was used to manage the virtualized computer labs and the virtual infrastructures include routers, switches and virtual machines on demand. The work of [7] built a cloud based systems lab to support students to access the computing resources remotely. A comparison of virtual lab solutions for online cyber security education was presented in [8], where the authors discuss different approaches of designing a virtual lab and address the criteria in selecting the optimal deployment method in virtual computer labs.

Perhaps the most widely used virtual computer lab system is the Virtual Computing Lab (VCL) platform [20, 21], which was initiated at North Carolina State University (NCSU) for applying cloud computing to teaching and learning. VCL is an open-source project [9] that offers a cloud computing platform with the goal of delivering dedicated, custom compute environments to users. When a user uses a program via VCL, the program runs on the server and VCL allows the user to control that program from his/her own computer. In this way, students and faculty can access software and applications that were typically only available while on campus in a lab or via individual software purchase and installation. The VCL have served students at NCSU, and have been adopted and deployed in many universities across the nation [3].

Collaborative learning is an important topic for online learning. A classification of collaboration approaches in E-learning was presented in [24]. A proposal of supporting collaborative work in online education was described in [25]. Several works also considered collaborative learning in virtual computer labs. The work of [6] supports collaborative learning for virtual labs by offering a social site for knowledge sharing and contribution. Another work [26] uses a virtual computer lab to show the benefit of collaborative learning by designing collaborative hands-on projects (i.e., projects requiring multiple students to work together) and comparing the learning outcomes with individual hands-on projects. Our work is different from these works by focusing on developing capabilities for students to collaborate in similar ways as they do in physical computer labs (e.g., sharing and controlling the same desktop).

III. COLLABORATIVE VIRTUAL COMPUTER LAB ENVIRONMENT

A. Overall Architecture

The CVCL environment under development follows a layered architecture to run on top of an existing virtual computer lab system such as VCL. As shown in Fig. 1, the layered architecture includes two layers: a virtual computer lab layer and a collaboration layer. The two layers are loosely coupled: the collaboration layer uses services from the virtual computer lab layer; it builds on top of an existing virtual computer lab system and is not tied to the specific system. This layered design brings flexibility to set up CVCL environments for many existing virtual computer lab systems and thus supports easy adoption of the tool.

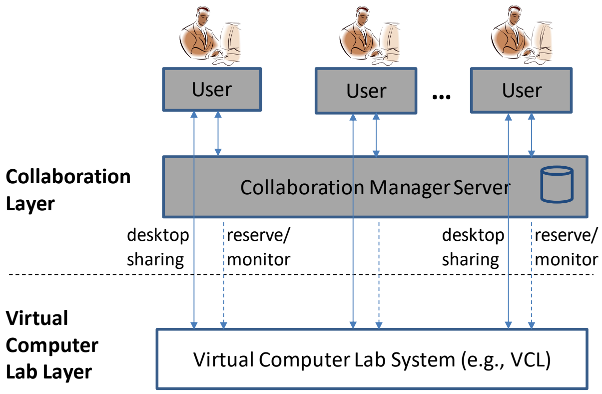

Fig. 1. Layered Architecture for the CVCL Environment

More specifically, at the virtual computer lab layer, an existing virtual computer lab system (e.g., VCL) provides all the necessary resources and functions for enabling virtual computer labs. These include hardware resources such as a cluster server, and software resources such as baseline operating system, hypervisor software, desired middleware or application that runs on the operating systems, and end-user access solution that is appropriate (see [3, 20] for examples of software needed in a virtual computer lab). Through virtualization, different instances of virtual machines are created and each instance includes an operating system and the desired application software (e.g., Matlab) requested by the user.

The virtual computer lab layer by itself does not support collaboration among users. To support collaborative labs, a collaboration layer is added on top of the virtual computer lab layer. The collaboration layer follows a client-server architecture and includes a collaboration manager server and a web client. The collaboration manager server is responsible for managing the collaborative labs, including lab reservation, session management, real time monitoring, and desktop sharing request/response among users. It includes a database to store information related to collaborative labs. The web client is responsible for connecting with the server and providing a GUI for users. To reserve a collaborative lab involving multiple users, a student uses the web client to send a request to the collaboration manager, which then makes the reservation through the underlying virtual computer lab system, e.g., the VCL system. The manager then passes the reservation information to the corresponding users. To join a reserved lab, a student uses the web client to connect to the collaboration manager. The client interface displays information such as participants of the lab, status of participants. It also offers functions such as presentation, audio, Web Cam, chat, and other web conferencing functions. We leverage an existing open source web conferencing system BigBlueButton to support these functions (more details in Section IV). Besides the web conferencing capabilities, an important function the CVCL environment supports is the *remote desktop control* function. Through remote desktop control, each user will have a virtual computer lab viewer that is connect to the reserved virtual resource provided by the virtual computer lab system and can work on the desktop in a "hands-on" fashion. Depending on the specific collaborative lab models (described below), multiple students may be connected to the same virtual desktop or have their own desktops but can request to access others' desktops in real time.

B. *Collaborative Lab Models*

To enable multiple forms of collaboration in virtual computer labs, the CVCL environment develops and integrates three collaborative lab models: *shared remote collaboration*, *virtual study room*, and *virtual tutoring center*. These three lab models are chosen because they represent important forms of collaboration and are often used in computer labs.

Shared Remote Collaboration – The main feature of the shared remote collaboration is that two or more students reserve and log into the same virtual computer lab and share the same desktop view. One student will be assigned as the driver to control the desktop cursor, while the others are the observers. They can request to switch roles at any time. The shared remote collaboration makes it possible for two or more students to synchronously collaborate with each other while working remotely on virtualized resources. It is similar to the software development paradigm pair programming [22, 23]. However, it is not limited to pair programming as it supports collaborative learning with any software running on virtual computer labs. For example, many proprietary software systems (such as Matlab) do not support collaboration between users. With the CVCL environment and the shared remote collaboration model, students not only can access these proprietary software systems remotely, but also can conduct collaborative labs in a well-defined manner. The shared remote collaboration model can be used in settings where an instructor sets the learning goals and explicitly groups students to use the virtual computer lab.

Virtual Study Room – The main feature of the virtual study room model is that each student works on his/her own virtual desktop but can share their desktop views with others and request to access others' desktops in real time when needed. Students can reserve a study room to study together and to help each other when working on their own virtual computer labs. A virtual study room brings two important advantages compared to the current practices of virtual computer labs where students are isolated. First, it fosters a sense of community amongst students because students are connected with each other in the same study room. Second, it makes it convenient for students to help each other because they can easily share the desktops in real time when needed. The virtual study room model can be used in informal settings associated with a course, a group project, or an interest group, etc. and gives each student the flexibility to collaborate as much or as little as they prefer.

Virtual Tutoring Center – The main feature of the virtual tutoring center model is that students can log into the virtual tutoring center and have "one-to-one" interactions with a tutor (or multiple tutors) who also logs into the system. A tutor can directly access the desktop of a student and interact on this virtual platform with the student. This capability is especially useful for tutoring computer labs, which typically involve answering questions regarding how to use software and/or how to fix bugs in programming code. Connections in the virtual tutoring center would be one-to-many connections. The tutor will have access to the desktops for all students active in the virtual tutoring center, however, students will not be able to view each other. The virtual tutor center has the advantage that students can access help from a tutor from any location at any time (provided a tutor is active at this time). In this way, a student is not obligated to be on campus in order to receive assistance.

IV. CVCL IMPLEMENTATION

Following the layered architecture described in Fig. 1, implementation of the CVCL environment mainly focus on the collaboration layer that runs on top of the virtual computer lab layer. In our implementation, the collaboration layer includes three functional components: 1) web conferencing support; 2) remote desktop sharing and control; 3) collaborative lab model management and data management.

To achieve effective collaboration, real time communication such as voice call, video call, and chatting among the participants of a virtual computer lab is essential. To support these functions, we leverage an existing open source web conferencing system BigBlueButton (https://bigbluebutton.org/) and integrate it as part of the CVCL environment. BigBlueButton offers many of the functions that are important for the CVCL environment, including real-time sharing of audio, video, slides (with whiteboard), polling, emoji icons (including raise hand), chat, and the presenter's desktop. As part of this implementation effort, a local BigBlueButton server has been set up. This server acts as the collaboration manager server of the CVCL environment to allow students to connect to the CVCL environment through a web browser.

While BigBlueButton does a lot of the heavy lifting for supporting collaboration among the participants, it misses an important function that is critical for collaborative virtual computer labs: remote desktop control. In fact, BigBlueButton allows a presenter to share his/her desktop during web conferencing. However, it does not support remote desktop control. This means a team member can only view another member's desktop but cannot touch and work on the other member's specific problem in a "hands-on" fashion. To address this issue, we integrate another software package called tightVNC (https://www.tightvnc.com/) into the CVCL environment. TightVNC is a lightweight free remote control software package that allows a user to see the desktop of a remote machine and control it with his/her local mouse and keyboard, just like he/she would do it sitting in the front of that computer . This capability makes it possible for multiple team members to work on the same "desktop" in a virtual computer lab, just like they do in a physical computer lab. We integrated the tightVNC package with the BigBlueButton web interface, so that a user can share or request to access the desktop of another user. The remote desktop will be displayed in a VNC viewer (a separated window from the web client window of BigBlueButton), through which remote desktop control is enabled. Fig. 2 shows an example of a VNC viewer displayed on top of a web client for the BigBlueButton-enabled CVCL environment. There are two participants (named as *Student1* and *Student2*) in this CVCL lab, which are displayed on the *Users* panel (the top left panel of the web interface). The VNC viewer running on *Student1*'s computer displays S*tudent2*'s desktop, through which *Student1* can access *Student2*'s desktop and work on the code that *Student2*'s is working on.

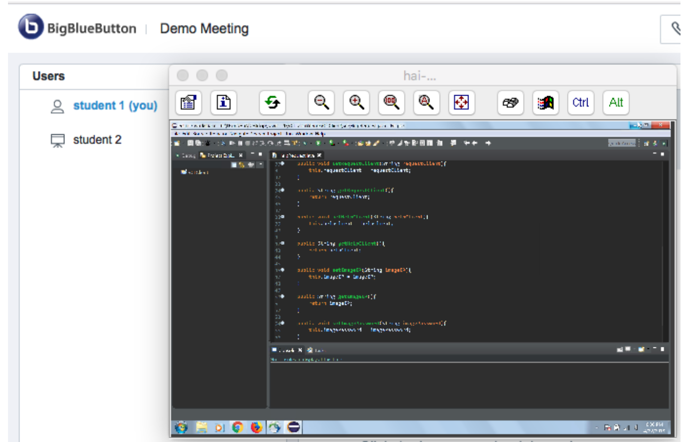

Fig. 2. Snapshot of CVCL client interface and remote desktop viewer

The third component of the collaboration layer is collaborative lab model management and data management. We have three different collaborative lab models as described in Section III. Each lab model has its own features and different rules regarding how participants can interact with others. For example, a participant in a virtual study room can share his/her desktop with all other participants, but in a virtual tutoring center only the tutors can view and access the desktop of a student. The collaborative lab model management takes care of all the different aspects related to collaborative labs, including reservation and login procedure of computer labs, status tracking and updating of participants during a lab, remote desktop sharing and control rules, as well as customized user interfaces for the different lab models. Besides the collaborative lab model management, data management is also important. The data management supports storing all the data related to the sessions of collaborative labs, such as participants, user login, start time and end time of the labs. These data are useful for analyzing the usage of the CVCL environment and for evaluating the effectiveness of the different lab models. We have set up a MySQL database and will integrate the database with the BigBlueButton server to support the data management.

We have implemented all the major functions described above and are at the stage of integrating them together. The implemented system has been utilized and tested by a small group of students and received positive feedbacks from the students. Our plan is to have an integrated CVCL environment by the end of the summer. Then we'll use the system in several computer science courses and will set up a virtual tutoring center in the department to evaluate the system.

V. CONCLUSION

This paper presents a collaborative virtual computer lab (CVCL) environment to support collaborative learning in cloud-based virtual computer labs. The developed CVCL environment follows a layered architecture that includes two layers: a virtual computer lab layer and a collaboration layer. It integrates three collaborative lab models to support different forms of collaboration. After the system is developed, we will evaluate the system in courses and by setting up a virtual tutoring center.

REFERENCES

[1]. R. Buyya, C.Y. Shin , S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platforms:Vision, Hype, and Reality for Delivering Computing as the 5th Utility," *Future Generation Computer Systems*, vol. 25, no. 6, June 2009.

[2]. H. E. Schaffer, S. F. Averitt, M. I. Hoit, A. Peeler, E. D. Sills and M. A. Vouk, "NCSU's Virtual Computing Lab: A Cloud Computing Solution," in *Computer*, vol. 42, no. 7, pp. 94-97, July 2009.

[3]. M. A. Vouk *et al*., "Using VCL technology to implement distributed reconfigurable data centers and computational services for educational institutions," in *IBM Journal of Research and Development*, vol. 53, no. 4, pp. 2:1-2:18, July 2009.

[4]. S. Stein, and H. Schaffer, "Cloud with a long tail: the VCL in support of pedagogy," *EDUCAUSE Review*, vol. 45, no. 3, pp. 10-11, 2010 Available: https://er.educause.edu/articles/2010/6/cloud-with-a-long-tail-the-vcl-in-support-of-pedagogy [Accessed: Jun. 18, 2018].

[5]. L. Xu, D. Huang, W. T. Tsai, "V-Lab: A Cloud-based Virtual Laboratory Platform for Hands-on Networking Courses," *Proc. 17th Annu. ACM ITiCSE*, pp. 256-261, 2012.

[6]. L. Xu, D. Huang and W. T. Tsai, "Cloud-Based Virtual Laboratory for Network Security Education," in *IEEE Transactions on Education*, vol. 57, no. 3, pp. 145-150, Aug. 2014.

[7]. J. Alexander, D. Aaron, H. Jonathan, H. Damen, and S. Mark, "Building a cloud based systems lab," *In Proceedings of the 13th annual conference on Information technology education*, *ACM* 2012, pp. 151-154,

[8]. J. Son, C. Irrechukwu, and P. Fitzgibbons, "Virtual lab for online cyber security education," *Communication of the IIMA*, vol. 12, no. 4, art. 5, 2012.

[9]. "Apache VCL.", 2012 [Online] Available: https://vcl.apache.org [Accessed: Jun. 18. 2018].

[10]. H. Avi, and V. N. Lunetta. "The laboratory in science education: Foundations for the twenty-first century," *Science education*, vol. 88, no. 1, pp: 28-54, 2004.

[11]. L. Springer, M. E. Stanne, and D.S. Samuel, "Effects of small-group learning on undergraduates in science, mathematics, engineering, and technology: A meta-analysis," *Review of Educational Research*, vol. 69 vol. 1, pp. 21-51, 1999.

[12]. L. Williams and L. Layman, "Lab Partners: If They're Good Enough for the Natural Sciences, Why Aren't They Good Enough for Us?," *20th Conference on Software Engineering Education & Training (CSEET'07)*, Dublin, pp. 72-82, 2007.

[13]. Saros, "Distributed Collaborative Editing and Distributed Party Programming.", 2018 [Online] Available: http://www.saros-project.org/ [Accessed: Jun. 18, 2018].

[14]. L, Lipponen, "Exploring foundations for computer-supported collaborative learning," *Proceedings of the Computer-supported Collaborative Learning Conference*, pp. 72-81, 2002.

[15]. I. Papapanagiotou, J. V. Devide., Devetsikiotis, and M. Devetsikiotis, "Networked 3d virtual computing for collaborative environments in science and education: Towards VCL 3.0," 2008 [Online]. Available: https://pdfs.semanticscholar.org/29e1/8a1642d7fc0ec6901c5e80dab891874155e3.pdf [Access: Jun. 18, 2018].

[16]. B. Greg, M. Padma, and Q. Dennis. "Cloud Computing: High Performance On Demand Solutions (HiPODS)," 2007 [Online]. Available: http://download.boulder.ibm.com/ibmdl/pub/software/dw/wes/hipods/Cloud_computing_wp_final_8Oct.pdf [Access: Jun. 18, 2018].

[17]. D. Ferraiolo, "NIST Cloud Computing," 2017. Available: http://csrc.nist.gov/groups/SNS/cloud-computing/ [Access: Jun. 18, 2018].

[18]. H. A. Ali, and H. A. El-Ghareeb, "Implementation of Cloud-based Virtual Labs for Educational Purposes", *IJCSNS International Journal of Computer Science and Network Security*, Vol.14, No.7, pp. 45- 49, July 2014.

[19]. R. I. Dinita, G. Wilson, A. Winckles, M. Cirstea and A. Jones, "A cloud-based virtual computing laboratory for teaching computer networks," *2012 13th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, Brasov, 2012, pp. 1314-1318.

[20]. S. Averitt, M. Bugaev, A. Peeler, H. Schaffer, E. Sills,S. Stein, J. Thompson, and M. Vouk, "The Virtual Computing Laboratory", *Proceedings of the International Conference on Virtual Computing Initiative,* IBM Corporation, Research Triangle Park, NC, 2007, pp. 1–16.

[21]. M. Vouk, "Support of Student Learning in a Technology-Rich Environment: Remote Application Access for Specialized and Collaborative Software," *IBM Supported University Research Proposal*, 2003.

[22]. W. Laurie & R. Kessler, *Pair Programming Illuminated*. Addison-Wesley, 2002.

[23]. L. Williams, "Integrating pair programming into a software development process," *Software Engineering Education and Training, 2001. Proceedings. 14th Conference on*, Charlotte, NC, 2001, pp. 27-36.

[24]. A. Al-Abri, Y. Jamoussi, N. Kraiem, and Z. Al-Khanjari, "Comprehensive classification of collaboration approaches in E-learning," *Telematics and Informatics*, vol. *34*, no. 6, pp. 878-893, 2017.

[25]. W. Luna., and J.L.C. Sequera, "Collaboration in the cloud for online learning environments: an experience applied to laboratories," *Creative Education*, vol. 6, no. 13, pp. 1435-1445, 2015.

[26]. A. Konak, and M. R. Bartolacci, "Using a virtual computing laboratory to foster collaborative learning for information security and information technology education," *Journal of Cybersecurity Education, Research and Practice*, vol. 2016, no.1, art. 2, 2016.