WiPiEC Journal

Works in Progress in Embedded Computing Journal

Special Issue on selected papers of
Works in Progress Session within 26th Euromicro Conference on Digital System Design (DSD) and 49th Euromicro Conference Series on Software Engineering and Advanced Applications (SEAA),
Durres, Albania, Sept. 6th – Sept. 8th, 2023.

September 2023 · Volume 9 · Issue 2 · ISSN 2980-7298

http://wipiec.digitalheritage.me

Message from the General Chair

It is my great pleasure to welcome you to the renewed session of Works in Progress (WiP) within *the 26th Euromicro Conference on Digital System Design (DSD'2023*) and *the 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA'2023)*.

I am honored to host the DSD'2023 and SEAA'2023 events in Durres, a city with 2500 years of history, located on the Albanian coast of the Adriatic. Today, it is a modern city with beautiful beaches, vibrant life and an important port that connects Albania and a significant part of the Balkans with Western Europe. But, city of Durres is always proud of its history and offers guests and visitors with memorable cultural and archaeological heritage. This is the first time that DSD and SEAA conferences are held in the Balkans that confirms an inclusive approach of the Euromicro community towards colleagues from developing countries, as well as the rapid advancement of this Region in advanced technologies.

First, I would like to thank all the authors of selected papers for WiP Session. Then to Karl-Erwin Grosspietsch Director of Euromicro Association for supporting me to renew this Session.

A high-quality SEAA and DSD academic program would not have been possible without the contribution of student volunteers to establish conference infrastructure, and produce relevant materials. One of them was Jovan Djurkovic, Dipl. Ing., which technically completed material for WiP Session. Thanks to him.

Finally, I would like to say thank you to all enthusiastic colleagues from DSD'2023 and SEAA'2023 Committees as well as to everyone who helped us in any way.

I am sure that WiP Session will grow during next events, because of its good fundaments and visionary idea.

Welcome to DSD'2023, SEAA'2023 and Works in Progress (WiP) Session!

Prof. Dr Radovan Stojanović
University of Montenegro, Montenegro
Euromicro DSD/SEAA 2023 General Chair
WiP Chair

Contents

Development of an Intelligent Compact Crawler Robot for House Foundation Inspection

Development of House Inspection Robot

Keiichiro Masuda
Graduate School of Science and Engineering
Hosei University
Tokyo, Japan
keiichiro.masuda.6g@stu.hosei.ac.jp

Goragod Pongthanisorn
Graduate School of Science and Engineering
Hosei University, Tokyo, Japan
po.goragod@gmail.com

Genci Capi
Faculty of Science and Engineering
Hosei University
Tokyo, Japan
capi@hosei.ac.jp

***Abstract*— In this work, we propose a mobile robot for house foundation inspection. The robot can operate in user-controlled mode and in autonomous mode. In user-controlled mode, the developed robot exchange information with the user through a GUI system. In addition, the operator can control the robot remotely. The robot moving trajectory is shown in the developed GUI. In autonomous mode the robot utilizes trained Deep Learning models running in the Raspberry Pi for concrete crack and water leaking detection. The results show that the proposed system is functioning well in the experimental environment and can be further expanded for other implementations.**

Keywords—Crawler robot, Deep Learning, House foundation investigation, damage recognition.

I. INTRODUCTION

In recent years, application of in human life environments is increasing, especially in rescue or human dangerous operations.

On the other hand, Japanese house safety is strongly related with the situations in their foundations. To cope with the hot and humid summer climate, it has an open structure with the characteristics of ventilation and heat shielding. Investigating the Japanese house foundation is difficult because the narrow space between the foundation and the floor. Therefore, the expert must move in a dark environment through narrow spaces (height around 60cm). However, since the foundation is generally not a place to be concerned about, the discovery of foundation troubles is discovered not in time. Therefore, regular and easy foundation inspections are really important for the safety of the house.

Two often foundation damages are crucks and water intrusions. Cracks in the foundation can appear for several reasons, including soil movement, water damage, or shoddy construction. Depending on how severe they are, foundation cracks can let water in, weaken the building, and cause additional damage if not repaired. Water collection around or beneath the foundation may be caused by improper drainage near the foundation or plumbing leaks. Long-term water exposure can undermine the foundation, cause erosion, and foster the spread of mold.

Several research works are focused on developing compact robots for investigation or rescue operations. For example, in [1] a crawler robot (MACbot) that has four track modules that can adapt the terrain changes is presented. It can run four tracks at once and crawl tracks as needed because there are four motors at each track. one crucial area for research and development in rescue robot systems is the study of robotic mechanism models with terrain adaptation and operational capabilities ([2]). However, there are some challenges in conquering high obstacles or steps ([4]).

Michael et al. proposed an integrated interface for robots [5]. They primarily use camera images in their system, which also displays the controlled robot's environment map, sensor data, and buttons that may be used to provide commands to the robot. Additionally, a lot of autonomous rescue robots are being created today [6-9].

In this paper, we introduce an intelligent compact crawler robot for house foundation inspection. By using the robots, house foundation inspection can be done more conveniently and efficiently. The developed robot can operate in user-controlled mode and in autonomous mode. We have developed a GUI for a remote control of the crawler robot. Furthermore, the robot can investigate the foundation situation autonomously. The robot utilized the trained deep learning algorithms [10-13], to identify cracks in the pillars and water leaks. In addition, the robot records its own movement route and creates a movement route diagram autonomously shown in the GUI monitor. The robot utilizes the generated route to return to its initial location.

II. HARDWARE DESIGN

A. Developed system

Fig.1 shows the entire crawler robot system. The operation of the crawler robot is performed by remote control by connecting the Raspberry Pi and a PC terminal via WI-FI communication. In user-controlled mode, the operator sends commands to the robot using the GUI. On the other hand, the robot captured camera image is transmitted to the user's PC.

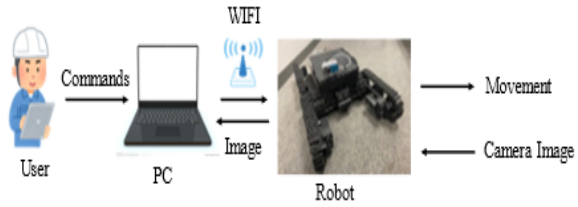
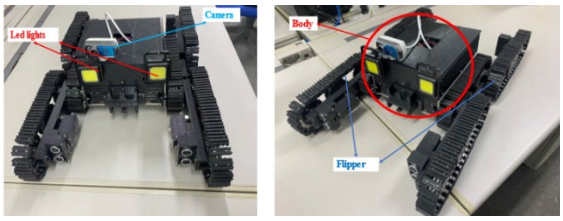


Fig.1. User-robot interface system.



(a) Front view (b) Side view

Fig.2 Developed robot.

Table 2. Robot specification

Length	520 [mm]
Width	320 [mm]
Height	130 [mm]
Weight	4.3 [kg]

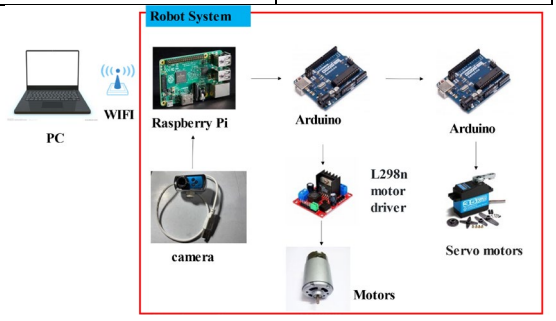


Fig. 3 Robot dataflow.

B. Robot

Based on the operation environments, we developed a compact crawler robot as shown in Fig. 2. The robot consists of the body and four movable flippers. The four flippers are utilized by the robot to climb stairs and steps or pipes that are present in-house foundations. The robot parameters are given in Table1. Main parts such as the flipper part of the robot were manufactured in 3D printer using a PLA to reduce the weight.

There are two main motors for driving the left and right crawlers of the body and four servo motors for driving the four movable flippers [3]. In addition, two LED lights are placed in the robot to lighten the environment.

Fig. 3 shows a schematic diagram of the control system of the robot. Two Arduino boards are used for servo motor control, and main motors. The Raspberry Pi sends the motor commands to the Arduino boards. To operate the robot in user mode, first we connect the Raspberry Pi and the PC via Wi-Fi communication wirelessly.

III. Developed GUI

There are two main parts in the user controlled mode: the operation interface and the image recognition system. To perform wireless remote control, we developed an GUI on the browser, as shown in Fig. 4. The GUI consists of three parts. The buttons in the upper half of the GUI (Fig.4 ①) are the operation buttons for controlling the robot flippers. The buttons on the right (Fig.4 ②) are the buttons for controlling the movement of the robot body. The operator presses buttons on the operation interface to issue commands to the Raspberry Pi. For operation interface we used Flask, which is an operation interface on a browser. Flask is a lightweight customizable framework, which is lighter than other isomorphic frameworks. Using Flask, the GUI pressed buttons are converted into commands and sent to the Raspberry Pi. Raspberry Pi converts commands from the operation interface into signals that Arduino can read. In addition, the robot captured image is shown in the user's monitor in a separate window.

Because it is a dark enviroment, after some forward, right and left robot motions it is very difficult to know the exact location of the robot. Therefore, we added a window in the GUI (Fig.4 ③) showing the robot motion trajectory. The Canvas element was used to create the movement path. You can draw figures in Html using the Canvas tag. When the operation button of the main motor is pressed, the operation interface autonomously records the operator's instructions and creates a moving path map of the robot based on the instructions. The starting point of the trajectory map is at the origin of the trajectory map. During foundation inspection using a robot, the robot may enter a position that the operator cannot see. It is possible to check the position of the robot even when the robot cannot be seen by moving path map. Also, using this path map, the robot can autonomously returns to its initial location, autonomously. By using the "return" button of the operation interface, the operation interface can change the robot trajectory information into a movement commands, which are sent to the robot. It is also possible to reset the route map by using the "reset Map" button.

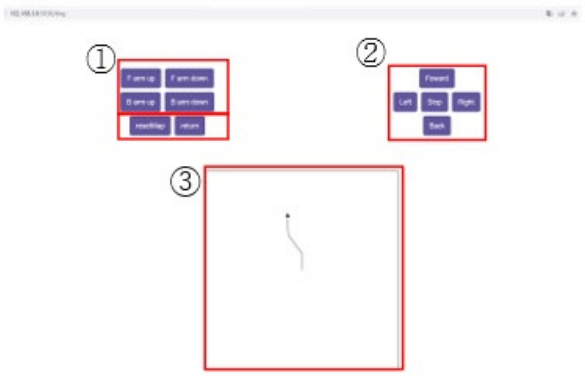
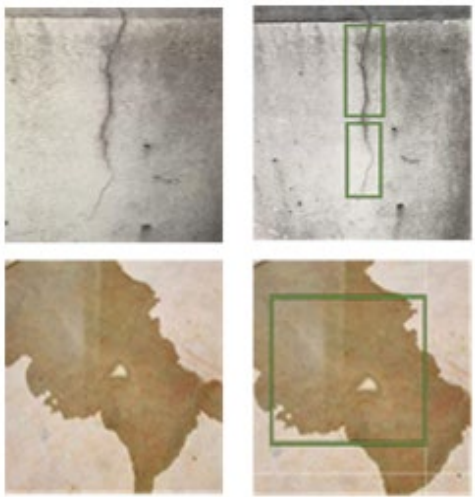


Fig. 4 Developed GUI.



(a) Original mage (b) Labeling image
Fig. 5. Sample images of crack and water database.

IV. Demage recognition

In this work, two demeges of the house oundations are considered: cracks and water leaks. For cracks and water leaks recognition we trained deep neural networks. DNNs are machine learning methods that uses multi-layer networks. In our implementation we used ensorFlow Lite which is an open source software library for machine learning in mobile devices such as Raspberry Pi.

For training the DNNs we labeled cracks and water leaks to create the dataset, as shown in Fig. 5. A total of 165 crack images and 151 water leak images were created. 252 are used as training images, 31 validation images, and 33 test images. The model performance is evaluated by the precision and the recall equations as follows:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

where TP is the true positive rate, FP is false positive rate, and FN is False Negative.

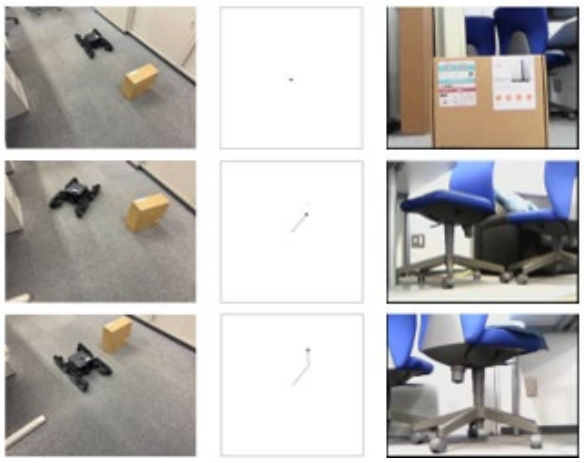
V. EXPERIMENTAL RESULTS

Initially we evaluated the developed GUI for different robot trajectories. Fig. 6 shows the robot motion controlled by the user relying only in the robots cuptured image.

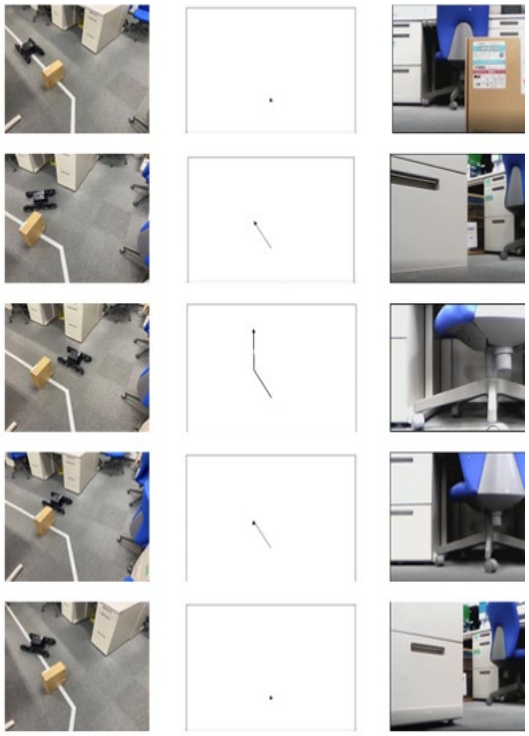
We also tested experimentally the ability of the robot to return to its initial location after moving autonomously in the enviroment. The robot automatically returned to the initial position using the "return" button (Fig. 7). Then, the robot's actual path, the map path and the initial and final positions were compared. The experiments were performed three times. The experimental results showed that the distance error between robot nitial and returned final locations were in a range of 33-37 mm.

Relation between precission and recall are shown in Fig. 8. Using the TensorFlow Lite's image recognition program, the DNNs achieved an accuracy of 76% and 63% for cracks and water leaks, respectively. In this experiment, we used 10 images of cracks and 10 images of water leaks to evaluate the actual performance of. The experiment is shown in Fig. 9. This performance evaluation is an experiment of recognition accuracy of the program. The evaluation criterion for recognition accuracy is the accuracy of program judgment.

As an experimental environment, we reproduced a dark house foundation situation in the Human Assistive Robot Laboratory on the Koganei Campus of Hosei University. We also added columns with cracks and ground with water leaks. The planned movement route is shown in Fig. 10(a). The experimental environment is shown in Fig. 10(b). Only the light of the robot was used in the experiments, as shown in Fig. 10(c).

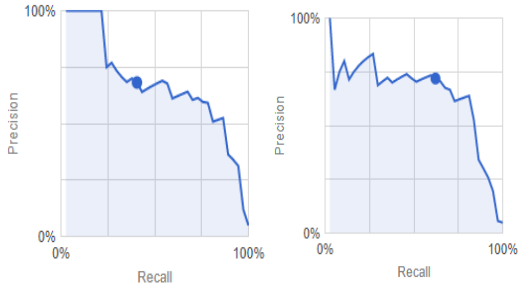


(a) Robot image (b) GUI image (c) Robot camera image
Fig. 6 Generated robot route in the GUI monitor.



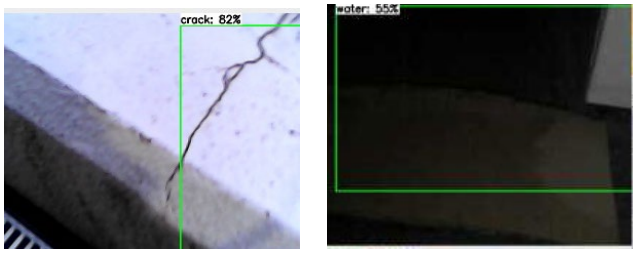
(a) Robot image b) GUI image (c) Robot camera image

Fig. 7. Robot autonomous motion.



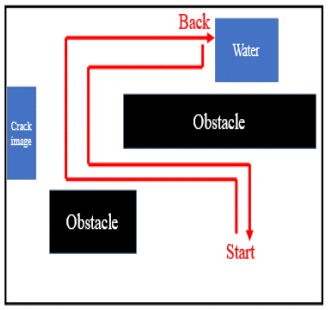
(a) Crack model (b) Water model

Fig. 8. Model evaluation results.

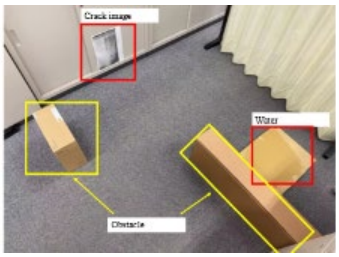


(a) Crack recognition by the robot (b) Water recognition by the robot

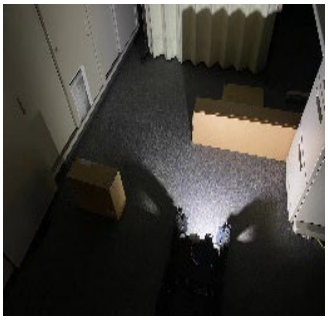
Fig. 9. Object detection result.



(a) Environment layout



(b) Real environment



(c) Dark environment

Fig. 10. Experimental environment.

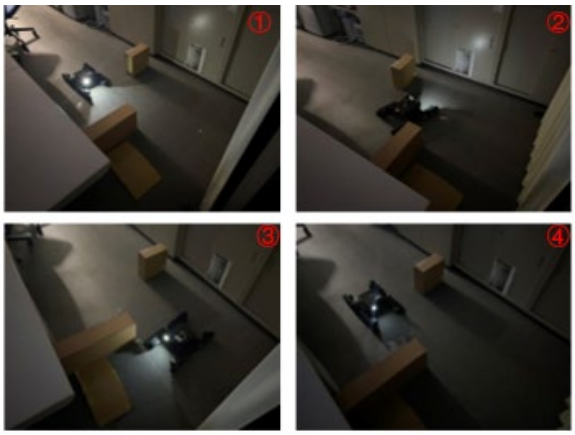
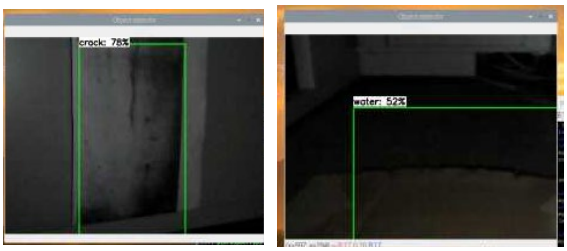


Fig. 11. User controlled robot using GUI and autonomous navigation.



(a) Crack (b) Water leak

Fig. 12. Damage detection.

In the experiments, the robot was controlled using the GUI by the user based on the image shown in the monitor. The robot detects cracks and water leaks in the experimental environment, and finally returns to its initial position autonomously.
Fig. 14 shows the video capture of the robot operation in the environment. After avoiding two obstacles and reaching the target point, the robot autonomously returns to the initial position. The error between the final position and initial position, even in this complicated environment, was 55 mm.

While the robot moves in the environment, a crack on the wall was detected with a recognition rate of 78%, as shown in Fig. 12. Finally, a water leak was detected with a recognition rate of 52% (Fig.15). In average, the average recognition accuracy for water leaks and cracks exceeded 60%.

VI. Conclusion

In this work, we presented an intelligent crawler robot for house foundation inspection. The operator can remotely control the robot. For remotely operation of the robot, the Raspberry Pi and a personal computer were connected via WIFI communication. The remote-control system operated on the browser using Flask. In addition, the movement path map was recorded by the robot, returning autonomously to its initial location even in a complicated route. For the damage recognition, such as cracks and water leaks, we used TensorFlow Lite using images streamed from the webcam. As a result, it was found that water leakage and cracks were recognized even in dark environments.

However, some issues must be considered in the future works, such as: 1) Improving the recognition accuracy of deep learning algorithms by collecting more data and testing other DL architectures; 2) Increasing the kind of damages recognized by the robot; 3) Improving the algorithm for robot localization in the environment.

References

[1] Quy-Hung Vu, Byeong-Sang Kim and Jae-Bok Song, "Autonomous stair climbing algorithm for a small four-tracked robot," 2008 International Conference on Control, Automation and Systems, Seoul, Korea (South), 2008, pp. 2356-2360.

[2] Zhenzhong Yu, Kaiti Cai, Wei Liu, Weicou Zheng, Review of Rescue Robot Technology, Journal of Jiangnan University (Natural Science Edition), 2015,14(04): 498-504.

[3] Hui Zhang, Xiangdong Cai, Dan Hai, Dengke Zhu, Shaoke Qian, Xun Li, Zhiqiang Zheng, NuBot rescue robot overall design , Robot Technique and Application, 2010(04): 17-19.

[4] Homma, K., Yamada, Y., Matsumoto, O., Ono, E., Lee, S., Horimoto, M., & Shiozawa, S. (2009, June). A proposal of a method to reduce burden of excretion care using robot technology. In *2009 IEEE International Conference on Rehabilitation Robotics* (pp. 621-625). IEEE.

[5] Michael Baker, Robert Casey et al. " Improved interfaces for human-robot interaction in urban search and rescue.", SMC (3) 2004: 2960-2965.

[6] Yugang liu, Goldie Nejat, "Robotic urban search and rescue: A survey from the control perspective", Journal of Intelligent & Robotics Systems, Vol. 72, Issue 2, pp. 147-165, 2013.

[7] David Erdos, Abraham Erdos, Steve E. Watkins, "An Experimental UAV System for Search and Rescue Challenge", IEEE Aerospace and Electronic Systems Magazine, Vol.28, Issue 5, pp.32-37, 2013.

[8] Teodor Tomic, Korbinian Schmid, Philipp Lutz, Darius Burschka,Toward a Fully Autonomous UAV: Research Platform for Indoor and Outdoor Urban Search and Rescue", IEEE Robotics & Automations Magazine, Vol.19, Issue.3, pp. 46-56, 2012.

[9] Ivan Vasilyev, Alla Kashourina, Maxim Krasheninnikov, Ekatherina Smirnova, "Use of Mobile Robots Groups for Rescue Missions in Extreme Climatic Conditions", 25th DAAAM International Symposium on Intelligent Manufacturing and Automation, pp. 1242 –1246, 2014.

[10] T. Duc Dung, and G. Capi, Application of Neural Networks for Robot 3d Mapping and Annotation Using Depth Image Camera, International Journal of Robotics and Automation, Vol. 37, Issue 6, 2022.

[11] 5. S. Nilwong, D. Hossain, S. Kaneko, G. Capi, Deep Learning-Based Landmark Detection for Mobile Robot Outdoor Localization. Machines 2019, 7, 25.

[12] D. Hossain, G. Capi, "Multiobjective Evolution of Deep Learning Parameters for Robot Manipulator Object Recognition and Grasping", Advanced Robotics, 32(20): 1090-1101.

[13] W. Qi, Y. Chun, Y. Sheng, G. Zhao and L. Wang, "Stability Analysis of Obstacle Avoidance Ability and Environment Adaptation Modeling of Snake Robot based on Deep Learning and Binocular Vision," 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2022, pp. 299-3024.

Recycling of Circuit Boards by Robot Manipulator Using Stereo Vision and Deep Learning

Intelligent robot for recycling of circuit boards

Yuho Takahashi
Graduate School of Science and Engineering
Hosei University
Tokyo, Japan
yuho.takahashi.6e@stu.hosei.ac.jp

Goragod Pongthanisorn
Graduate School of Science and Engineering
Hosei University
Tokyo, Japan
po.goragod@gmail.com

Genci Capi
Faculty of Science and Engineering
Hosei University
Tokyo, Japan
capi@hosei.ac.jp

Abstract— ***To make possible the recycling of printed circuit boards, automated systems for object classification and degree of overlapping are needed. In this thesis, a recycling system with a robot manipulator was developed by using Deep learning and a stereo vision approach. The system operates in the following order: object detection by deep learning, calculation of 3D point clouds by stereo vision, and grasping by the robot manipulator. Four experiments were conducted to evaluate the developed recycling system. The four experiments were: measurement of object detection accuracy, measurement of stereo vision responsiveness, measurement of vertical judgment accuracy and grasping accuracy, and operation test of the actual machine. Deep learning and stereo vision for the robot manipulator were found to be effective for the printed circuit board recycling system. The results also shed light on the challenges of automating the recycling process.***

Keywords-component; formatting; style; styling; insert (key words)

I. INTRODUCTION

Electronic circuit boards used in computers, smartphones, home appliances, and other devices are made of rare and valuable metals. Therefore, it would be advantage to have an in-depth recycling of the PCBs materials. Several recycling methods for PCBs have already been proposed. Most of the planned works rely on units to separate the components and heating technique to melt the solder ([1], [2]). In addition to manual disassembling, an automatic disassembly was proposed in [3], using the visual data. In [4] a disassembly machine using a robotic arm was proposed. Other works suggested separation methods based on each material's specific density [5], magnetic separation [6,] and integrating gravity and the force of a viscous liquid [7].

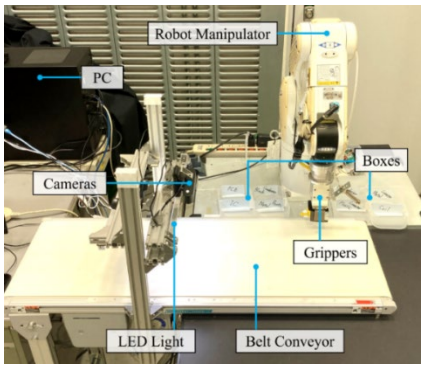
There are still a lot of problems to be resolved even if several ways for recycling PCBs have been suggested. This is as a result of the wide variation in PCBs and their intricate structure. In this paper, we provide a deep learning ([8]) based method for recycling PCB components. Recent Deep Belief Networks (DBN)-based techniques have proven to perform at the cutting edge in a range of tasks, including text databases, natural language processing, picture and speech processing, and many more. Only object classification and recognition are the focus of the majority of DBNs works [9–13]. Previous DBN applications have generally concentrated on niche uses, like robot semantic place recognition [14]. The authors employed a two-step cascade structure with two deep layers and an RGBD depth picture from a kinect sensor in [15].

In this paper, we propose a stereo vision-based object recognition and localization for robotic recycling tasks. In difference from our previous system [16], in our implementation, the robotic manipulator recognizes the PCB components which overlap each-other. The camera image is used as input of the CNN to recognize the recycled objects and separate them in specific boxes.

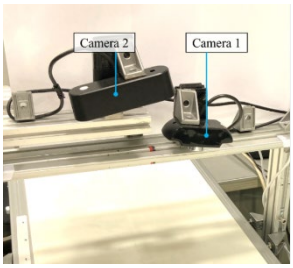
We evaluate the system in in terms of a) object recognition; b) relative vertical location of objects using the stereo vision; c) robotic manipulator grasping accuracy. In our work, the PCBs are cut into tiny objects and placed on a belt conveyor. The robot manipulator motion is developed based on camera data and conveyer speed, and the categorized pieces are then put in different boxes. The performance of two grippers because PCB board components have a variety of sizes, shapes, and are frequently quite flat.

II. SYSTEM CONFIGURATION

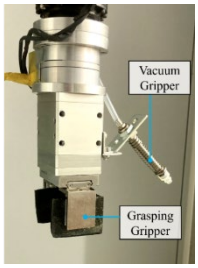
A robot manipulator and a belt conveyor are used to separate pieces of printed circuit boards of various types, sizes and shapes, as shown in Fig. 1. Fig. 1(a) shows the system for recycling of circuit boards, using a Denso VS-050 robot manipulator. The robot has a small base, and it can move in a wide range. Two cameras, as shown in (Figure 1(b)), are utilized for object recognition and stereo vision generation. An electric gripper is used to grasp the objects that are not flat (Fig. 1(c)). In addition, for flat objects we use a vacuum gripper (Fig. 1(c)).



（a）Recycling system of waste circuit boards.



（b）Cameras



（c）Grippers

Fig. 1. Recycling system.

III. DEEP LEARNING

In this work, Faster R-CNN is used to detect the recycled objects in the camera images [17]. Deep learning is a machine learning technique composed of multiple layers like the biological brains. Each layer consists of multiple neurons. During learning, the weights of the neuron connections are trained to generate the features for object recognition. The number of layers and the number of neurons in each layer are adjusted by trial and error. The input of Faster RCNN is the image. Faster R-CNN not only recognizes the object, but also generates the location of the object in the captured image. In addition, it was possible to detect multiple types of objects, simultaneously.

In Faster R-CNNs, the RPN (Region Proposal Network) is first trained, and then the CNN (Convolutional Neural Network) layers are trained. RPN detects if there is an object in the captured image, while CNN recognizes the object. For training, many images containing objects to be classified are prepared as training data. In addition, the position information and type data of the objects in those images are collected.

In this work, we investigate the effect of the number of pixels of the training data and the image representation on the detection accuracy. We change the number of pixels from 1920 x 1080 to 960 x 540. RGB and HSV image representations are utilized, and the results are compared. In HSV the image information is stored by three elements "hue, saturation, and brightness". In the RGB images each pixel is composed by red, green, and blue colors. Then, from the prepared training data, 4 patterns are trained with a network composed of 450 convolution layers and maximum pooling layers. The configuration of the network is shown in Figure 2. The learning conditions are 32 epochs, 1 mini-batch size, and 0.0001 learning rate.

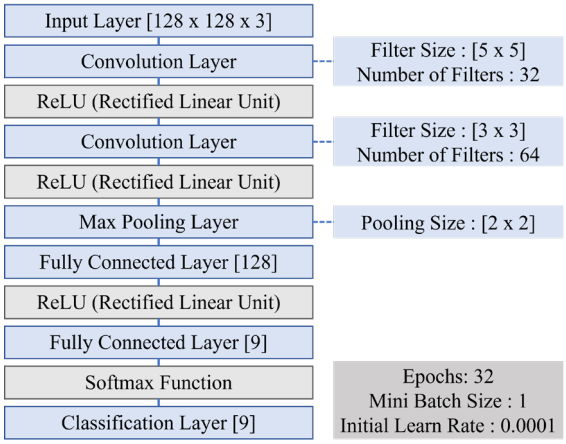


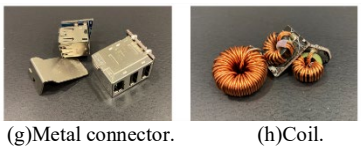
Fig. 2. CNN configuration.

In this work, 450 images are used for training and 150 images are used for evaluation. In addition, cross-validation was performed by exchanging the training data and the evaluation data and performing learning four times. The evaluation is done in terms of the recognition rates and the position of the object in the captured images. To judge the types and positions of objects on the belt conveyor, we created a network trained using Faster R-CNN [15]. We first created sample data to train Faster R-CNN. There are eight objects that must be recognized: The eight classes are PCB (the resin part of the board), IC (integrated circuit), Black Condenser, Silver Condenser, Plastic Connector, Metal Connector, Metal Parts (metal plates, USB connectors, etc.), and Coils, as shown in Fig. 3. The reason for choosing these eight classes is that they

are widely used in various electrical circuits.

(a)PCB. (b)IC. (c)Black condenser.

(d)Sliver connector. (e)Plastic connector. (f)Metal connector.



(g)Metal connector. (h)Coil.

Fig. 3. Recycle objects.

IV. STEREO VISION

In our implementation first the objects are recognized in the 2D captured image. When objects overlap each-other, 3D data are required. Stereo vision is a method to calculate 3D information (3D point cloud) of the object using images captured by two cameras. The purpose of the stereo data is to determine the relative position of recognized objects in the case they overlap each-other. This is important to grasp objects correctly and separate them in their respective boxes. The 3D information is obtained from the corresponding points in the two cameras captured images [18]. Referred to parameters shown in Fig. 4, the following equations are written:

$$l = a + b = z\tan\theta_a + z\tan\theta_b \quad (1)$$

$$\therefore z = \frac{l}{\tan\theta_a + \tan\theta_b} \quad (2)$$

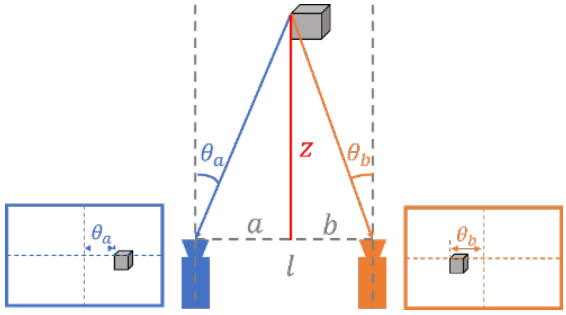


Fig. 4. Depth calculation using stereo vision.

In this work, a checkerboard is used to set the coordinate system for judging the position of the objects and the parameters for the position relationship between the two cameras (Fig. 5). Since the checkerboard is made up of black and white squares, it is easy to detect the corners with edge detection. Since these corners are equally spaced and coplanar, the positional relationship between the camera and the checkerboard is calculated. In other words, you can transform the coordinate system of the camera image and the coordinate system of the surface on which the checkerboard is located. Also, by using this principle, the positional relationship between the two cameras is calculated.

To evaluate the performance of the stereo vision, two pair of overlapping objects are placed on the belt conveyor simultaneously. The 3D information is calculated using the stereo vision. This was repeated 84 times, and the correct average correct results for each class are recorded. As the recycling objects overlap each other, in addition to recognition accuracy, the relative position in the vertical direction generated by the stereo vision is also important. This is strongly related to the grasping accuracy of the robot manipulator. In this work, it is assumed that two overlapping objects can be detected correctly, and 3D information can be calculated by stereo vision.

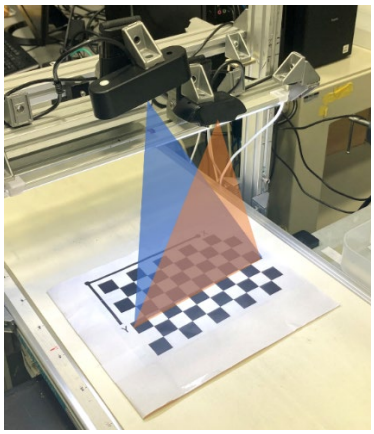


Fig. 5 Using a checkerboard with two cameras.

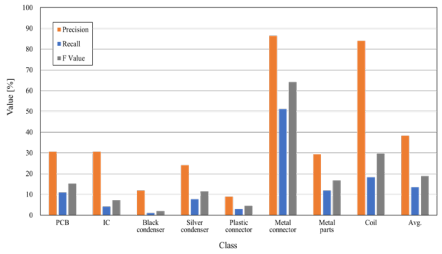
In our implementation, there are 28 combinations of paired recycle objects, making the recognition task difficult. We placed pair of objects on the belt conveyor overlapping each other. The robot grasping accuracy is calculated only in the case when the objects are correctly recognized. The PCBs are collected by the vacuum gripper, and other objects by the electric gripper. To calculate the gripper accuracy, 24 grasping experiments for each pair are conducted.

V. RESULTS

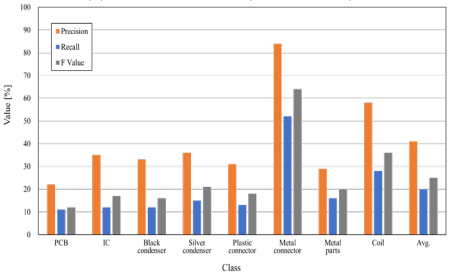
The results of RGB and HSV for different pixel number is shown in Fig. 6 and 7, respectively. During the experiments, the number of pixels of the camera and the training images are the same. The results show that the HSV gave better results compared with RGB. Results show that there is no much difference in computation accuracy of Faster RCNNs trained with pictures of different number of pixels. However, when the number of pixels is reduced by 1/4, the computational load and time is also reduced. Because the computation time is strongly related to the robot response, small number of pixels would be advantage for real time robot implementations. Therefore, the results of pattern 4 are the best among four learning patterns.

We measured the accuracy of 3D information of stereo vision for each class. The average accuracy of 3D information is 63%. Although it is slightly lower than the average recognition accuracy of 67%, it is good for real time implementation. We measured the accuracy of relative vertical position of objects using the visual image. The results for each combination and their average values are shown in Fig. 8. The average accuracy is 72%, and the gripping accuracy is 63%. By class, the values for Metal Connector and Metal Parts are slightly lower. This is because the objects are made of metal reflecting the light, making it difficult for the camera to see the outline of the object.

More than 100 operation experiments are conducted to evaluate the developed system. Fig. 8 shows the screen captured image in which objects are not only recognized but also their relative vertical position is generated. The red color objects are above blue color objects.

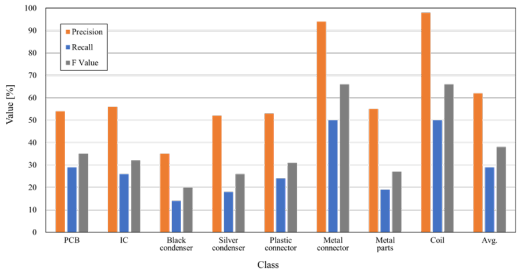


(a) Result of RGB (1920x1080).

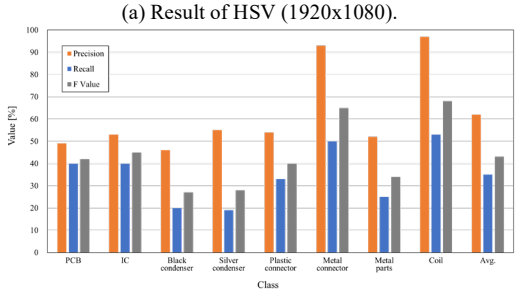


(b) Result of RGB (960x540).

Fig. 6. RGB image results.



(a) Result of HSV (1920x1080).



(b) Result of HSV (960x540).

Fig. 7. HSV image results.

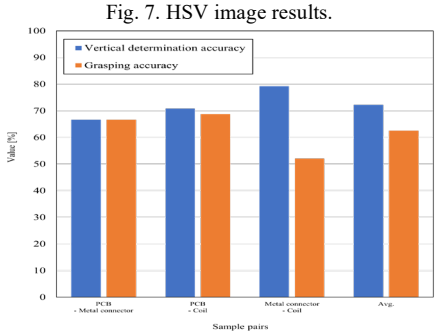


Fig. 8. Overlapping and grasping accuracy.

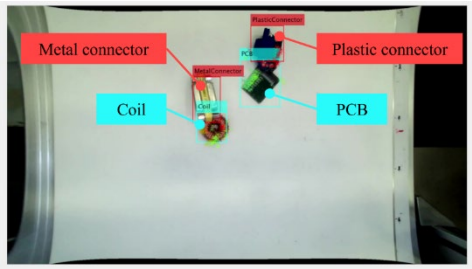


Fig. 9. Object recognition in overlapping situation.

VI. Conclusion

In this work, we proposed a recycling system for printed circuit boards using a robot manipulator, deep learning and stereo vision. RGB and HSV visual representations with different number of pixels are used for recognition and stereo information. The deep learning was trained to generate the object category and the location of the object in the captured image. The results showed that HSV image representation gave a better performance.

References

[1] I. Stobbe, K. Schrank, C.Weickardt, H. Griese, H. Reichl, and L. Stobbe, "Quality assured disassembly of electronic components for resue," in Proc. IEEE Int. Symp. Electron. Environ., San Francisco, CA, May 6–9, 2002, pp. 299–305.

[2] S. Yokoyama, K. Ikuta, and M. Iji, "Recycling system for printed wiring boards with mounted parts," in Proceedings of First Int. Symposium on Environmentally Conscious Design and Inverse Manufacturing, 1999, pp. 814–817.

[3] K. Feldmann and H. Scheller, "The printed circuit board-a challenge for automated disassembly and for the design of recyclable interconnect devices," in Conference on Clean Electronic Products and Technology Concept, no. 415, 1995, pp. 186–190.

[4] H. Zebedin, K. Daichendt, and P.Kopacek, "A newstrategy for a flexible semi-automatic disassembling cell of printed circuit boards," in Proc. IEEE Int. Symp. Industrial Electronics—ISIE 2001, Pusan, Korea, June 2001, pp. 12–16.

[5] J. B. Legarth, "Recycling of electronic scrap," Ph.D. dissertation, Tech. Univ. Denmark Lyngby, 2002.

[6] P. Galbraith and J. L. Devereux, Beneficiation of Printed Wiring Board With Gravity Concentration: Concurrent Technologies Corporation, International Association of Electronics Recyclers, 2002.

[7] K. Feldmann and H. Scheller, "The printed circuit board-a challenge for automated disassembly and for the design of recyclable interconnect devices," in Conference on Clean Electronic Products and Technology Concept, no. 415, 1995, pp. 186–190.

[8] D. Hossain, G. Capi, M. Jindai and S. Kaneko, "Pick-place of dynamic objects by robot manipulator based on deep learning and easy user interface teaching systems," Industrial Robot, vol. 44, issue 1, pp. 11-20, January 2017.

[9] MA Keyvanrad, and MM Homayounpour, "A brief survey on deep belief networks and introducing a new object oriented MATLAB toolbox (DeeBNet V2.0)", arXiv:14083264 (2015).

[10] G. E. Hinton, S. Osindero, and Y.-W. Teh : "A Fast Learning Algorithm for Deep Belief Nets", Neural Computation, Vol. 18, No. 7, pp. 1527–1554 (2006).

[11] A. Mohamed, G. Dahl, and G. Hinton, "Deep Belief Networks for phone recognition", NIPS Workshop on Deep Learning for Speech Recognition and Related Applications, Canada, pp. 1–9 (2009).

[12] V. Nair and G. Hinton : "3D Object Recognition with Deep Belief Nets", Advances in Neural Information Processing Systems 22, pp: 1339-1347 (NIPS 2009).

[13] B. Leng, X. Zhang, M. Yao, and Z. Xiong, "3D Object Classification Using Deep Belief Networks", MultiMedia Modeling, Vol. 8326, pp 128-139 (2014).

[14] A. Hasasneh, E. Frenoux, and P. Tarroux, "Semantic Place Recognition Based on Deep Belief Networks and Tiny Images", 9th International Conference on Informatics in Control, Automation and Robotics ICINCO, 2012, Rome, Italy.

[15] I. Lenz, H. Lee, and A. Saxena, "Deep Learning for Detecting Robotic Grasps", International Journal of Robotics Research, Vol. 34, No. 4-5, pp 705-724 (2015).

[16] K. Naito, et al. : Recycling of printed circuit boards by robot manipulator: A Deep Learning Approach, 2021 IEEE International Symposium on Robotic and Sensors Environments (ROSE), 2021

[17] S. Ren, et al. : Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, arXvi, Cornell University, https://arxiv.org/pdf/1506.01497.pdf, 6 Jan 2016

[18] R. Hartley, and A. Zisserman : Multiple View Geometry in Computer Vision, Cambridge University Press, 25 May 2004.

Towards Modelling 5G Communication in Software Architectures of Vehicular CPS

Zenepe Satka, Saad Mubeen, Mohammad Ashjaei
School of Innovation, Design, and Engineering
Mälardalen University
Västerås, Sweden
firstname.lastname@mdu.se

John Lundbäck
Mastering Software Complexity
Arcticus Systems
Järfälla, Sweden
john.lundback@arcticus-systems.com

Abstract—**Advanced vehicular Cyber Physical System (CPS) applications like autonomous quarries require high-bandwidth and low-latency communication among the vehicles and their control centers. 5G offers a promising solution to meet these communication requirements. One of the core challenges is how to model and timing analyze distributed software architectures of these large and complex systems that utilize 5G communication. To the best of our knowledge, there is no existing modelling language or a component model that models distributed software architectures of CPS that use 5G communication. In this paper, we take the first step in addressing this gap by proposing extensions to an existing industrial component model for vehicular systems, namely RCM, to support modelling of 5G communication in the distributed software architectures.**

I. Introduction

Vehicular software systems have been drastically increasing in size and complexity for the past few years [1], [2]. This can be attributed to the increasing demand for new software-based features in modern vehicles. Furthermore, these systems are often constrained by real-time requirements that are specified on their software architectures. Hence, the developers of these systems have to deal with the challenges of managing the size and software complexity as well as verifying the specified timing requirements at the design time. This makes the development of these systems a daunting task. Model-based Engineering (MBE) and Component-based Software Engineering (CBSE) [3], [4], complemented by timing analysis techniques [5], [6], are proving effective in dealing with these challenges during the development of these systems.

Many industrial CPS use cases in the vehicular domain require the vehicles to communicate with each other to perform a joint functionality, e.g., collaborating vehicles in an autonomous quarry. These vehicles are often equipped with high data-rate sensors (e.g. lidars can transmit data at 100 Mbit/s and more). The large amount of acquired data needs to be processed and then communicated wirelessly to other vehicles with low latencies. 5G offers a promising solution to meet the high-bandwidth and low-latency wireless communication requirement. With the intensive development of 5G [7], it is foreseen that CPS will eventually depend on the availability offered by 5G and beyond to interact with the physical world as well as process and communicate information in an end-to-end manner [8]. Modelling of 5G at the software architecture abstraction is challenging because of its higher communication and protocol complexity compared to wired networks like Switched Ethernet. **One of the core challenges is how to model and timing analyze distributed software architectures of these large and complex systems that utilize 5G communication.** The existing modelling languages and component models in the vehicular domain, such as EAST-ADL [9], [10], AUTOSAR standard [11], [12], Fraunhofer ESK [13], Rubus Component Model (RCM) [14], to name a few, support modelling and timing analysis of distributed software systems that use only wired (onboard) real-time communication. This communication is often based on low-bandwidth and low-latency networks like CAN standard [15] and low-latency and high-bandwidth networks like Ethernet TSN standards [16]. **To the best of our knowledge, there is no existing modelling language or a component model that supports software architecture modelling of distributed software systems that use 5G communication.**

This paper takes the first step in bridging the gap by extending an existing industrial component model for vehicular systems, namely RCM [14], to support modelling of 5G communication. The proposed extensions are comprehensive enough to model the detailed timing information about 5G on the software architectures. In the future, the modelled timing information will be utilized to perform end-to-end timing analysis [6], [17] of the software architectures of vehicular software systems that use 5G communication. The concrete contributions in this paper include:

- We present the system model of 5G communication that is used to introduce an abstract model of 5G that can be incorporated into the abstraction of software architectures.
- We propose extensions to an existing industrial component model, RCM, to support modelling of 5G. The extensions are expressive enough to model all the timing information that is required to perform end-to-end timing analysis of distributed software architectures using 5G.

II. Background

A. 5G Communication

5G is the fifth generation of mobile networks designed to provide higher network speed, higher bandwidth, ultra-low latency, and high reliability. It introduces a new feature called Ultra-Reliable Low-Latency Communication (URLLC) to support real-time applications with latencies as low as 1 ms and reliability levels up to 99.999% [18]. URLLC together with the QoS mechanisms [19] is a promising candidate for wireless communication in time-critical CPS. Furthermore, 5G provides support for enhanced mobile broadband (eMBB), and massive machine-type communications (mMTC) [20]. The eMBB provides high data rates, higher user mobility, and high density to support a wide range of high-bandwidth applications. Fig. 1 depicts the features and services supported by 5G [21].

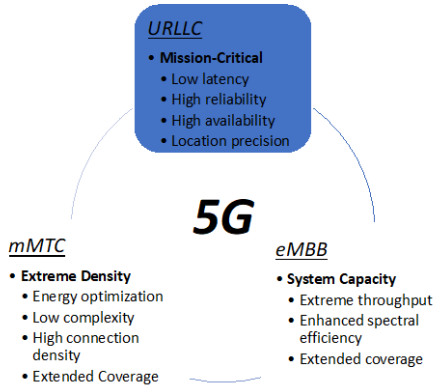


Fig. 1: Features and services provided by 5G.

B. Rubus Component Model (RCM)

Rubus, developed by Arcticus Systems[1], is a collection of models and tools for model-driven development of time-critical software systems [14]. The modelling language and tool suite supported by Rubus are called RCM and Rubus-ICE respectively. Rubus also provides a real-time operating system that is certified according to ISO 26262 safety standard. Rubus has been used in the vehicle industry for over 30 years by several OEMs and Tier-1 companies, e.g., Volvo Construction Equipment, BAE Systems Hägglunds, Hoerbiger, Knorr Bremse, to mention a few.

In RCM, the highest-level hierarchical element is called the *system*, which contains one or more models of nodes or Electronic Control Units (ECUs) and zero or more models of networks as shown in Fig. 2. The *network* object consists of two parts: one is independent of the underlying communication protocol, while the other is protocol dependent. The protocol-independent part defines messages, their properties, and data/signals that are mapped to them. The protocol-dependent part of the network is defined for each individual communication protocol separately.

The lowest-level hierarchical element in RCM is called the *Software Circuit (SWC)*, which encapsulates a software function. The interaction between the SWCs is clearly separated in terms of data (via data ports) and control flows (via control ports) as shown in Fig. 2. One important principle in RCM is to separate functional code from the infrastructure that implements the execution model. This, in turn, facilitates visualisation of explicit synchronisation and data access at the abstraction of software architectures.

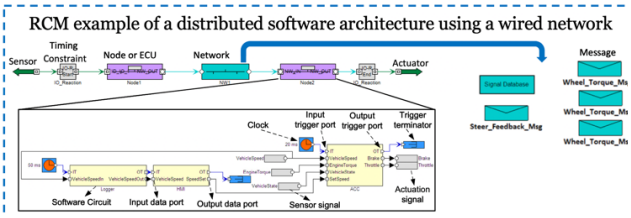


Fig. 2: RCM model of a distributed software architecture.

Today, Rubus supports modelling and end-to-end timing analysis of distributed software architectures that employ onboard (wired) networks like CAN [22] and TSN [23]. Similar to other modelling languages in the vehicular domain like AUTOSAR, EAST-ADL, and AMALTHEA, RCM does not support modelling and end-to-end timing analysis of 5G communication in distributed software architectures.

[1] https://www.arcticus-systems.com

III. Model of End-to-End 5G Communication

This section presents an abstract model of end-to-end communication in 5G. This model will provide input for extensions to RCM. We consider the 5G system to consist of a single-antenna or general Node B (gNB) using mid-band frequencies and a channel bandwidth of 20MHz. The end-to-end connection in the 5G network is shown in Fig. 3, where user equipment 1 (UE 1) is the transmitter, while UE 2 is the receiver. UE 1 uses the uplink (UL) transmission to send data to the 5G network via the gNB, while UE 2 uses the downlink (DL) to receive the data via the gNB.

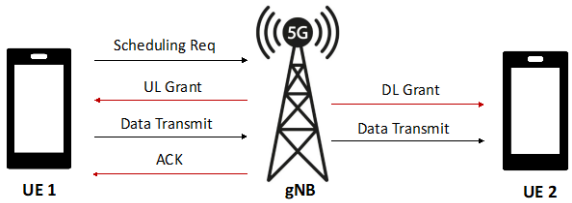


Fig. 3: End-to-end connection in 5G network.

To establish an uplink transmission, the 5G network follows some procedures which are simplified below:

1) *Scheduling Request:* The user sends a scheduling request via the control channel to the gNB asking for a specific amount of data to be sent with some level of priority. In this request, the user also sends the Channel Quality Indicator (CQI) value which is an indicator of the channel quality considering the signal strength and different interference components existing on the radio side.
2) *Uplink (UL) Grant:* The gNB evaluates the request and sends an uplink grant to the user considering its CQI value. This grant holds information related to the specific time and frequency slot for the uplink transmission, the appropriate Modulation Coding Scheme (MCS), and the coding rate used by the user when transmitting the data.
3) *5G Data Transmission:* The user takes the grant from the gNB and starts the uplink transmission using the allocated time and frequency slot.
4) *Acknowledgment (ACK):* The gNB sends a positive ACK to the user in cases where the transmission is done correctly and no errors were encountered, otherwise it will send a negative ACK.

Once the data is transmitted successfully to the network, the gNB identifies the receiver and starts establishing a downlink transmission as follows:

1) *Downlink (DL) Grant:* The gNB allocates the required radio resources and sends a downlink grant to the receiver with information about the allocated time and frequency as well as the MCS and coding rate to be used for the downlink transmission.
2) *5G Data Transmission:* After the needed resources are allocated, the gNB starts the downlink data transmission.

Higher CQI value results in the assignment of higher modulation schemes in the uplink or downlink grant, which means a higher amount of data can be sent/received within the allocated time slot as also shown in Fig. 4.

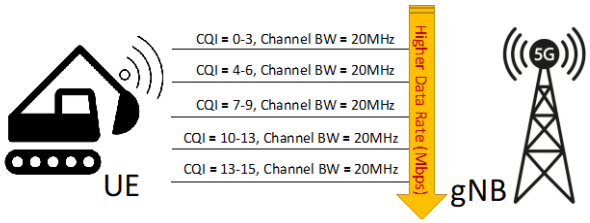


Fig. 4: Data rate values depending on the CQI value.

IV. PROPOSED EXTENSIONS TO RCM

5G communication in RCM is modelled by extending the protocol-dependent part of the existing network object (see Fig. 2). The extensions include new properties of the protocol-dependent part corresponding to 5G communication as well as interfaces for the device side for uplink and downlink transmission. Fig. 5 shows the abstract model of end-to-end 5G connection in RCM.

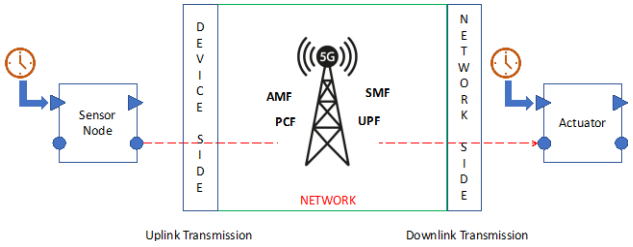


Fig. 5: Abstract model of end-to-end 5G connection in RCM.

User Equipment (UE) Model in RCM: The UE can be viewed as the sensor node in Fig. 5. In RCM, the UE is modelled with the existing node element (see Fig. 2) that can send or receive messages over the 5G network object.

gNB Model in RCM: The gNB is modelled as a special type of switch element in RCM. This element contains a set of communication channels. Depending on the user-requested QoS-es and the received CQI value, the gNB assigns one of the channel IDs which fulfills the user's requirements. The properties of the gNB model in RCM are shown in Fig. 6.

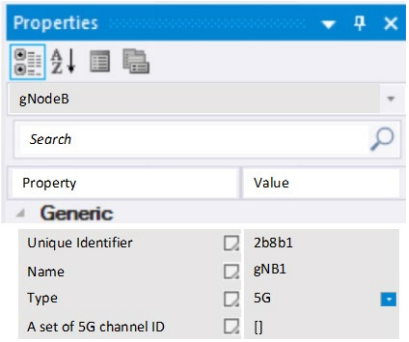


Fig. 6: gNB properties in RCM.

Scheduling Request Model in RCM: The scheduling request in RCM is modelled as a special type of message (see Message element in Fig.2). This special message has additional properties that are depicted in Fig. 7. This message is sent by the user to notify the gNB about the requested Quality of Services (QoS) and the size of the data payload in bytes. The properties associated to this message include "Transmission Type" (Periodic or Sporadic), "Size of Data Payload" in bytes, a "CQI" value, "Priority" and "Deadline".

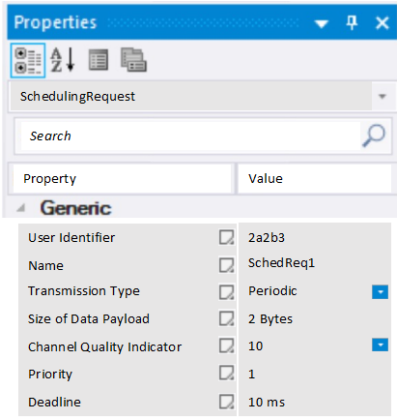


Fig. 7: Scheduling Request properties in RCM.

5G Channel Model in RCM: The properties of the 5G channel model in RCM are presented in Fig. 8. There are two types of channels in 5G: control channels and data channels. The control channels are used to transmit control information such as the Scheduling Request or the Configured Grant. On the other hand, the data channels are used to transmit the data payload on the uplink side or downlink side. A channel is assigned to each connected user with a specific Modulation Coding Scheme (MCS) which is related to the number of bytes that a device can transmit using this channel. Depending on the channel quality information, modulation scheme and coding rate, there is a calculated data rate in Mbit/s for each channel.

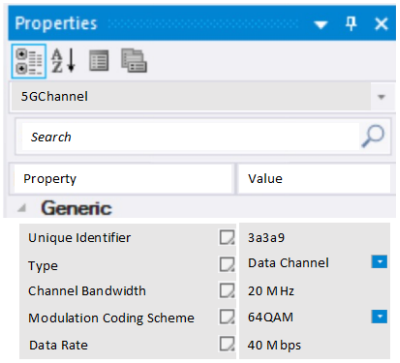


Fig. 8: 5G channel properties in RCM.

Configured UL/DL Grant Model in RCM: The configured grant contains control information notifying a specific user about the channel ID which can be used for data transmission. The channel ID itself contains a set of characteristics to ensure the requested QoS for the uplink/downlink transmission. The properties of the configured grant in RCM are presented in Fig. 9.

5G Data Transmission Model in RCM: The data transmission contains all the 5G message properties: (i) the transmission type which can be periodic or sporadic; (ii) the data rate in Mbit/s also known as the network speed for the channel being used; (iii) a priority level of the message; (iv) a deadline constraint which needs to be met by the 5G network, and (v) the size of the payload in bytes. These properties are modelled in RCM as depicted in Fig. 10.

Acknowledgment (ACK) Model in RCM: In Rubus, the acknowledgment is a simple control message, sending a positive integer in cases where the transmission is done correctly and no errors were encountered, otherwise, it sends a zero to notify the user that an error was encountered.

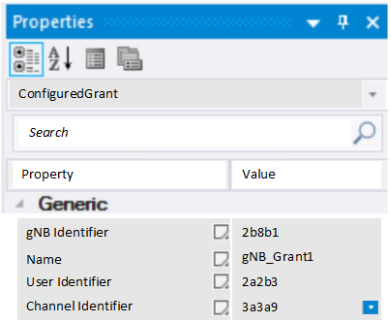


Fig. 9: Configured Grant properties in RCM.

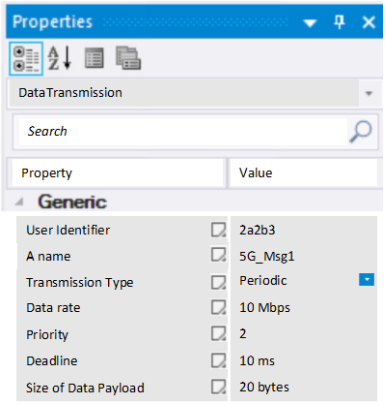


Fig. 10: Data Transmission properties in RCM.

V. SUMMARY AND ONGOING WORK

The high-bandwidth and ultra-low-latency communication supported by 5G has the potential to support real-time communication in many time-critical CPS. The work in this paper presents the first step to address the core challenge of modelling 5G communication in distributed software architectures of complex CPS in the vehicular domain. In this regard, we proposed extensions to an industrial component model for vehicular systems, RCM, to support modelling of 5G communication. The extensions are designed to be comprehensive enough to model the detailed timing information about end-to-end 5G connection on the software architectures. The modelled timing information will be used in the future to perform end-to-end timing analysis of the distributed software architectures.

Currently, we are in the process of validating the proposed extensions to RCM by modelling distributed software architecture of a large industrial use case of an autonomous quarry. The outcomes of modelling the use case, corresponding discussions, and usability feedback from the vehicle manufacturer will be used to refine the proposed extensions to RCM. In the future, we plan to develop an end-to-end timing analysis of 5G-based distributed software systems and support the application of the analysis at the software architecture abstraction.

ACKNOWLEDGMENT

This work is supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA) via the DESTINE, PROVIDENT & INTERCONNECT projects.

REFERENCES

[1] J. Schroeder, C. Berger, A. Knauss, H. Preenja, M. Ali, M. Staron, and T. Herpel, "Predicting and evaluating software model growth in the automotive industry," in *IEEE International Conference on Software Maintenance and Evolution*, Sep. 2017, pp. 584–593.

[2] L. Lo Bello, R. Mariani, S. Mubeen, S. Saponara, "Recent advances and trends in on-board embedded and networked automotive systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, 2019.

[3] T. Vale, I. Crnkovic, E. S. de Almeida, P. A. da Mota Silveira Neto, Y. C. Cavalcanti, and S. R. de Lemos Meira, "Twenty-eight years of component-based software engineering," *Journal of Systems and Software*, vol. 111, pp. 128 – 148, 2016.

[4] T. A. Henzinger, J. Sifakis, "The Embedded Systems Design Challenge," in *14th International Symposium on Formal Methods*, 2006.

[5] B. Akesson, M. Nasri, G. Nelissen, S. Altmeyer, and R. Davis, "A comprehensive survey of industry practice in real-time systems," *Real-Time Systems*, vol. 58, 09 2022.

[6] N. Feiertag, K. Richter, J. Nordlander, J. Jonsson, "A Compositional Framework for End-to-end Path Delay Calculation of Automotive Systems under Different path Semantics," in *Workshop on Compositional Theory and Technology for Real-time Embedded Systems*, 2008.

[7] Z. Satka, M. Ashjaei, H. Fotouhi, M. Daneshtalab, M. Sjödin, and S. Mubeen, "A comprehensive systematic review of integration of time-sensitive networking and 5g communication," *Journal of Systems Architecture*, vol. 138, 2023.

[8] C. Tranoris, S. Denazis, L. Guardalben, J. Pereira, and S. Sargento, "Enabling Cyber-Physical Systems for 5G Networking: A Case Study on the Automotive Vertical Domain," in *4th International Workshop on Software Engineering for Smart Cyber-Physical Systems*, 2018.

[9] "EAST-ADL Domain Model Specification, V2.1.12,," http://www.east-adl.info/Specification/V2.1.12/EAST-ADL-Specification_V2.1.12.pdf.

[10] R. T. Kolagari, D. Chen, A. Lanusse, R. Librino, H. Lönn, N. Mahmud, C. Mraidha, M.-O. Reiser, S. Torchiaro, S. Tucci-Piergiovanni, T. Wägemann, and N. Yakymets, "Model-based analysis and engineering of automotive architectures with east-adl: Revisited," *Int. J. Concept. Struct. Smart Appl.*, vol. 3, no. 2, pp. 25–70, 2015.

[11] The AUTOSAR Consortium, "Autosar technical overview," in *Version 4.3.*, May 2016, http://autosar.org.

[12] S. Fürst and M. Bechter, "Autosar for connected and autonomous vehicles: The autosar adaptive platform," in *46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W)*, Jun. 2016, pp. 215–217.

[13] Fraunhofer ESK, Safe Adaptive Software for Fully Electric Vehicles, https://www.iks.fraunhofer.de/en/projects/safeadapt.html.

[14] S. Mubeen, H. Lawson, J. Lundbäck, M. Gålnander, and K.-L. Lundbäck, "Provisioning of predictable embedded software in the vehicle industry: The rubus approach," in *4th IEEE/ACM International Workshop on Software Engineering Research and Industrial Practice*, 2017.

[15] ISO 11898-1, "Road Vehicles–interchange of digital information–controller area network (CAN) for high-speed communication", ISO Standard-11898, Nov 1993.

[16] Time-Sensitive Networking (TSN) Task Group, TSN standards, https://1.ieee802.org/tsn.

[17] M. Becker, D. Dasari, S. Mubeen, M. Behnam, and T. Nolte, "End-to-end timing analysis of cause-effect chains in automotive embedded systems," *Journal of Systems Architecture*, vol. 80, pp. 104 – 113, 2017.

[18] M. Bennis, M. Debbah, and H. V. Poor, "Ultra-Reliable and Low-Latency Wireless Communication: Tail, Risk, and Scale," *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834–1853, 2018.

[19] Z. Satka, M. Ashjaei, H. Fotouhi, M. Daneshtalab, M. Sjödin, and S. Mubeen, "QoS-MAN: A Novel QoS Mapping Algorithm for TSN-5G Flows," in *28th IEEE RTCSA Conference*, 2022.

[20] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges," *IEEE Access*, vol. 6, 2018.

[21] Mallinson, K. 3GPP – The Path to 5G: As much Evolution as Revolution. The Mobile Broadband Standard. [Online]. Available http://www. 3gpp.org/news- events/3gpp- news/1774- 5g_wiseharbour.

[22] S. Mubeen, J. Mäki-Turja, and M. Sjödin, "Communications-oriented development of component-based vehicular distributed real-time embedded systems," *Journal of Systems Architecture, Vol. 60 (2)*, 2014.

[23] S. Mubeen, M. Ashjaei, and M. Sjödin,"Holistic modeling of time sensitive networking in component-based vehicular embedded systems," in *45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2019, pp. 131–139.

A Brief of Distributed Data Processing

Clemente Izurieta
Montana State University
Gianforte School of Computing
Bozeman, MT, USA
clemente.izurieta@montana.edu

Nate Woods
Montana State University
Gianforte School of Computing
Bozeman, MT, USA
me@bign8.info

Ann Marie Reinhold
Montana State University
Gianforte School of Computing
Bozeman, MT, USA
reinhold@montana.edu

Abstract—**Distributed data processing is a cornerstone in modern cloud and edge computing environments because of its ability to handle large amounts of information that can overwhelm a single computer. However, the ontogeny of research in the field of distributed data processing remains poorly characterized. Therefore, we reviewed 70 publications discussing distributed data processing. Distributed processing systems is an active area of research with publications increasing in numbers since the early 2000s. The most salient topics in distributed processing systems were affiliated with system architecture and programming paradigms. However, researchers lack standard metrics for reporting throughput, hampering the comparison of existing studies. This study is a first step towards characterizing this field of research and identifying important areas of opportunity.**

Keywords—mapping study, data processing, distributed processing systems.

I. Introduction

The volume and velocity of big data pose significant challenges to the software engineering community. The rates at which data are being collected, stored, and processed in data-center and cloud-computing environments have grossly outpaced the rate at which hardware can be developed [5]. The current wave of data production has software designers playing catch-up with the volume of data needing to be processed.

Because distributed systems are capable of handling large amounts of data, software engineers have proposed several techniques to advance distributed data processing. However, these techniques have been developed without a thorough review of needs and trends in the field of software engineering. Here, we take a first step towards this review by addressing the following three goals:

- Goal 1 (G1): Assess and review areas of research in distributed data processing.
- Goal 2 (G2): Aggregate and report the throughput of distributed processing systems researched.
- Goal 3 (G3): Investigate trends in the primary literature to explore how the published research has documented research attention and changes.

A. Terminology

Throughout this manuscript, we reference foundational terms used in the distributed processing domain.

1) Data: A *datum* is a single piece of information or a fact that can also be referred to as a record or a sample. The pluralization of a datum is data, and a set of facts is also known as a set of data or a data set.

2) Worker: A *worker* is a single running process that is capable of handling a given task. Workers can also be called processors or nodes depending on the context, but for the purposes of this discussion, these terms can be considered equivalent. Here, we use the term "node" for consistency.

3) Batch Processing: Batch processing breaks data processing into separate gathering and processing phases. This enables software designers to isolate expensive processing from the critical path of data acquisition.

4) Stream Processing: Stream processing handles samples in real time, providing more frequent and incremental results than batch processing. This is achieved by allowing workers to maintain state from previous inputs and treating additional inputs as incremental changes to the current state.

B. Prior Work

No existing studies of distributed data processing cover both stream and batch processing in detail. Stream processing has been researched more extensively than batch processing. For instance, a systematic literature review of distributed data processing empirically characterized stream programs and contributed valuable details of stream processing programs to guide the design of stream processing frameworks [7]. However, batch processing techniques were not characterized in this work. No other studies provided ample treatment of batch processing, thereby leaving a gap in the literature on distributed data processing. Other mapping studies focused on distributed data processing within particular subdisciplines of computing (e.g., Big Data [1] and Internet of Things [2]). For instance, Akoka et al. [1] discuss Big Data, where distributed data processing was used to help process data. Similarly, Alkhabbas et al. [2] discuss the characterization of the Internet of Things (IoT) where distributed data processing focuses on *things* that have physical representation in the real world. However, we were unable to find a systematic investigation of distributed data processing across computing subdisciplines.

Critical gaps in the existing body of research remain. Our work fills these gaps by (1) covering a breadth that is not found in prior work including both batch and stream processing techniques; (2) ensuring our study is agnostic across computing subdisciplines (e.g., Big Data); (3) using *Natural Language Processing* (NLP) to impart objectivity; and (4) normalizing

performance metrics across studies to enable comparisons across works that no other study provides.

II. Approach

We reviewed the primary literature on the field of distributed data processing and the ontogeny of this field from 2002-2019 following the guidelines provided by Budgen et al. [4].

A. Literature Search Criteria and Strategy

We searched the IEEE Xplore®[1], Association for Computing Machinery Digital Library[2] and Google Scholar[3] for publications on the topic of distributed data processing. These sites provided indexed results on topical publications and ensured our search corpus covered a breadth of publications in distributed data processing.

To focus our review on the most salient and topical areas, we defined our search string as follows: ("distributed data" OR "data processing") AND ("processing pattern" OR "algorithm design"). These terms characterize the key concepts of distributed data processing, design patterns, and algorithmic design better than the blanket term "stream processing."

B. Selection Criteria

The pruning of the corpus was done using stringent inclusion and exclusion criteria. We included papers that discuss distributed data processing, were published between 2002-2019, were written in English, and contained experimental research. We excluded publications that used distributed data processing but did not research it directly, and eliminated publications that were not peer-reviewed. Using these criteria, we selected a total of 70 publications.

C. Identifying Contribution Areas and Common Terms

We performed a comprehensive review of all 70 papers by first reading them from cover-to-cover. We then determined the contribution areas in each publication to determine the scope of topics in the corpus using a manual coding approach. We coded the contribution areas of the publications by manually documenting the themes and topics described in the *Abstract, Introduction, Results, and Conclusion* sections of each paper. Using these documents, we then grouped related publications based on the topics each publication covered (Table I).

We next analyzed the abstracts from every paper using NLP. We strategically selected abstracts for NLP analysis because abstracts contain the most salient topics and important keywords that capture and characterize the most important topics in a study. Using WEKA-3.8.4 [6], we sanitized all words in each abstract by removing stop words using the English Dictionary from Natural Language Tool Kit [3]. We manually stemmed remaining words because automated stemming algorithms performed poorly on the technically written abstracts. We defined the set of common terms as the list of unique words present in 20% of abstracts. We calculated the frequency of each of these terms in each abstract in the corpus.

TABLE I
Publication Topics Identified via Manual Coding. Article Id corresponds to the unique article identifier in the corpus

Topic	Articl Id
System Design	
Network	25, 44
Dynamic	6, 8, 24, 41
Architecture	9, 29, 48
Performance	45, 67
Edge Computing	14, 30, 57
Single Processor	33
Extend Architectures	15, 19
via generalization	1, 21, 40, 64
via specialization	22, 60, 65
Performance	5, 11, 50
GPU / FPGA	16, 36
Data at Rest	26, 42
Data Model	52, 53
Data in Motion	13, 47
Applications	
Query Optimization	23, 66
Graph Queries	27, 28, 35, 62
Visualization	12, 59
Security	34, 70
Accuracy	51
Classification	18, 37
Decision Trees	7, 38, 68
Classifier Chains	20, 39
Nearest Neighbor	32, 56, 61
Ensemble Techniques	3, 31, 43
Decision Makers	2, 46
Clustering	58, 63
Social Media	49, 55
Stock Market	4
High Dimensional Data	10, 17, 54, 69

D. Data Extraction

We extracted the following throughput metrics:

- *Rate* R (bytes/second). The amount of data a system can process over the time necessary to process that data.
- *Data Size* D (bytes). The total amount of data processed.
- *Duration* t (seconds). The time it took to process data.
- *Sample Size* $\overline{S_i}$ (bytes). The average size of each sample.
- *Number of Samples* $|S|$. The number of samples.
- *Nodes* N (positive integer). The number of workers partaking in each operation.

Few studies directly reported the metric of primary interest to us: the rate at which data was processed (Table II). However, with the exception of N, these metrics can be defined in terms of one another (equation 1).

$$D = R \cdot t \qquad D = \overline{Si} \cdot |S| \tag{1}$$

Therefore, we calculated values for unreported metrics for as many publications as possible. For all remaining publications, we used data imputation to infer values for metrics that were not reported directly and could not be calculated algebraically.

[1] https://ieeexplore.ieee.org
[2] https://dl.acm.org/
[3] https://scholar.google.com/

TABLE II
PUBLICATIONS REPORTING THROUGHPUT METRICS BY CATEGORY

Category	Reported (or Calculated)	Inferred	Insufficient Info
Nodes	55	0	15
Samples	45	5	20
Duration	31	5	34
Data Size	9	30	31
Rate	6	23	41

TABLE III
SIZE APPROXIMATIONS USED IN IMPUTATION

Classification	Attribute Type	Size (bytes)
Numeric	dimension, attribute, sample	4
(Numeric, Numeric)	point, location, graph edge.	8
String	tweet, word, log, trace, request.	140

E. Data Imputation

We employed statistical imputation to infer values for missing metrics. We assigned values to attributes based on their size and type (Table III). For numerical types, we assumed that the 32-bit representation (either a float or an integer) would be sufficient to propagate the necessary information through a distributed system. For types consisting of two points, we chose 8 bytes or the sum of two 4-byte attributes. For text types, we chose 140 bytes to mirror "tweet" [8].

We inferred data size from 30 publications and processing rate from 23 publications (Table II). However, imputation also introduced a threat to validity. Any imprecision in our selection of size approximations (Table III) propagated to the results. To mitigate these threats to the conclusion validity of this study, we present imputed results metrics as *Inferred*. Therefore, the reader can clearly identify which results are based on data reported directly versus results inferred via imputation.

III. RESULTS & DISCUSSION

A. G1 Findings

Prior research has focused on two main research topics: the design and architecture of distributed systems, and the applications that run on distributed systems. These two topics were clearly discernible from both the manually coded (Table I) and NLP (Fig. 1) results. These topics represent the two highest-level topics identified from the manual coding; the most common words identified via semi-automatic coding confirmed the importance of these two topics. The terms "System(s)" and "Application(s)" fell within the top 10 words and are the two most informative words in the top 10. Other words in the top 10 (e.g., "Stream(...)" and "Time") were topical but did not assist in discerning research topics whereas other words reflected terms commonly used in research publications and were neither topical nor discerning (e.g., "Based", "Two", and "Results").

Both manual and semi-automatic coding agreed on other frequently used terms. Examples of such terms include "Performance", "Network", "Architecture", and "Graph". The agreement in these terms demonstrated that both coding approaches detected similar patterns in the field of study. However, the results of the coding approaches were also complimentary. For instance, "Parallel" and "Mining" were only identified by the semi-automated approach whereas "Dynamic" and "Query" was only identified by manual coding.

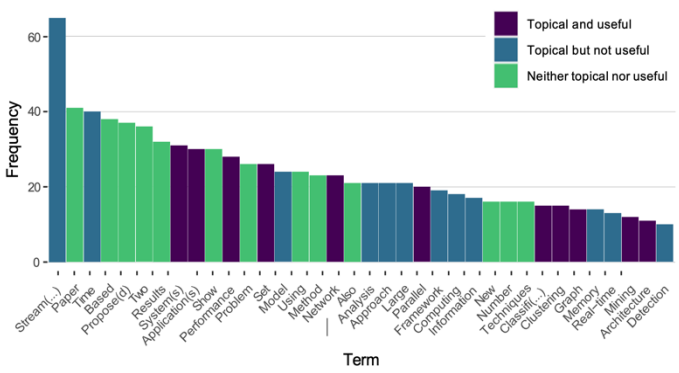


Figure 1. Frequency of Abstracts Containing the Terms Indicated on the X-axis. Bar color indicates whether the term was topical and useful, topical but not useful, or neither topical nor useful. Stemmed terms indicated by "(...)"; terms where singular and plural instances were combined indicated with "(s)".

B. G2 Findings

The field lacks standard and consistent metrics for reporting findings. Note the number of metrics in Table II that were neither reported nor could be calculated algebraically. A lack of comparable metrics between studies precluded efficient inter-study comparison and meta-analysis. It also hampered benchmarking, as imputation can introduce inconsistencies in results. Nevertheless, we characterized the throughput of distributed processing systems to the extent possible.

With respect to nodes (Fig. 2A), a large percentage (44%) of distributed system research was not conducted on physically distributed systems (i.e., research was conducted on one node on 31 of the 70 papers in our corpus). Further, only 6 papers reported researching systems larger than 16 nodes, with only 1 paper investigating a 144-node system.

With respect to the number of samples being processed (Fig. 2B), most publications reported numbers between one thousand and ten million. Only 14 publications processed data sets with 10 million samples or more. Three publications reported handling data sets on the order of billions of samples. Without additional context, it is difficult to interpret these results because samples can vary in size.

With respect to size (Fig. 2C), the paucity of data was surprising because knowledge of how much data was being processed is foundational for comparing experiments with one another. The largest data set was nine terabytes. Given that the average size of nodes was approximately 16 gigabytes, a distributed system is a promising avenue for data processing.

With respect to runtime (Fig. 2D), most publications reported experiment durations between 1 sec. and 16 min. Given that stream processing systems can be run continuously, these short experiments offered little inference for long-running experiments or the maintenance of distributed systems. These limited durations are concerning for professionals looking into using a distributed solution for their data processing needs because the paucity of information about the long-term utility

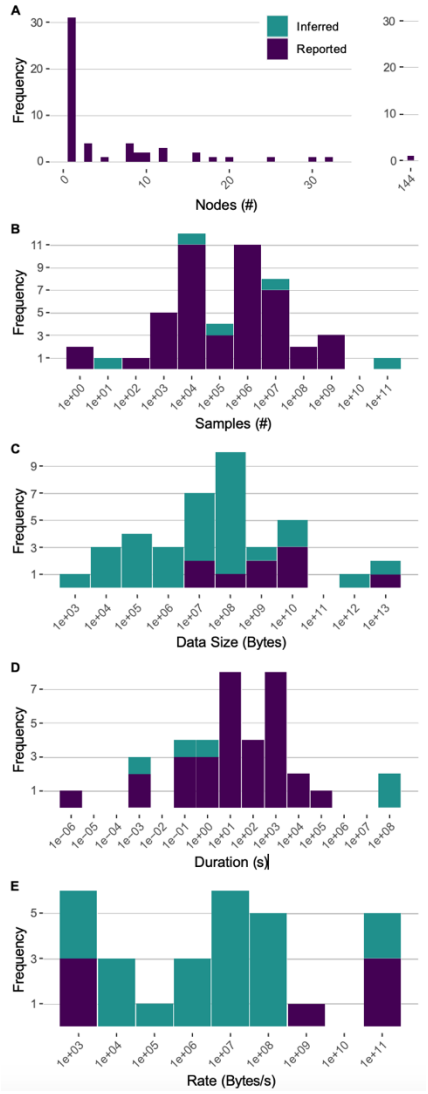


Figure 2. Histograms of Key Metrics. "Frequency" (y-axis) is the count of publications in the corresponding bars on the corresponding x-axis. Results directly reported or algebraically calculated are shown in dark purple; results inferred via imputation are shown in blue-green color.

and maintainability of distributed systems makes it difficult to form a basis for judgment.

With respect to rate (Fig. 2E), most were inferred via imputation due to insufficient reporting. Recall, only 7 of 70 publications reported the rate at which data could be processed by their distributed system (Table II). However, the rate is a critical metric and perhaps the most relevant metric for consumers of distributed processing systems.

C. G3 Findings

From the early 2000s to 2016, research in the area of distributed data processing increased (Fig. 3). Thereafter, research publication numbers fluctuated. Note that we formed

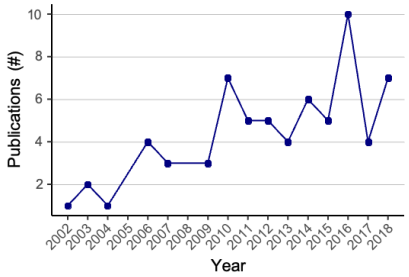


Figure 3. Count of Distributed Data Processing Publications in Corpus

our corpus of publications during 2019 and therefore did not have a complete count of publications for that year.

IV. Conclusion and Future Directions

Since the early 2000s, the field has grown, and he most active areas of research included algorithm design and system design. Many researchers are contributing to ever-increasingly specialized optimizations to be performed over the general pattern of distributed processing. Despite the growth, a standardized set of metrics for reporting results is lacking. Without such standards, the comparison of experimental results from one study to the next is hampered. A core set of metrics should be reported, including those listed in Table (II).

Importantly, the scope of academic research is limited in comparison to commercial applications. The scale that industry leaders such as Facebook, Amazon, Netflix, and Google operate is significantly greater than any study reviewed here. Future academic research on this topic would benefit from industry partnerships, simulation experiments, and exploring the use cases under which distributed data processing is—and is not—the most advantageous solution available. The corpus of new papers is also growing, and we intend to investigate publications through 2023.

References

[1] Akoka J., Comyn-Wattiau I., and Laoufi N. Research on big data–a systematic mapping study. *Comp. Stand & Interfaces*, 54:105–115, 2017.

[2] Fahed Alkhabbas, Romina Spalazzese, and Paul Davidsson. Characterizing internet of things systems through taxonomies: A systematic mapping study. *Internet of Things*, 7:100084, 2019.

[3] Sean Bleier. Natural language took kit's list of english stopwords. https://gist.github.com/sebleier/554280, 2021. Accessed: 2021-01-24.

[4] D. Budgen, M. Turner, P. Brereton, and B. A. Kitchenham. Using mapping studies in soft. engineering. In *PPIG*, V8, pg. 195–204, 2008.

[5] Omri M.N., Helali L. A survey of data center consolidation in cloud computing systems. doi.org/10.1016/j.cosrev.2021.100366, Feb 2021.

[6] Machine Learning Group University of WAIKATO. Weka 3 - data mining with open source machine learning software in java. https://www.cs.waikato.ac.nz/ml/weka/, 2021. Accessed: 2021-01-24.

[7] Thies W., and Amarasinghe S. An empirical characterization of stream programs and its implications for language and compiler design. In *2010 19th Int. Conf. on Parallel Architectures and Compilation Techniques (PACT)*, pages 365–376. IEEE, 2010.

[8] Twitter. Decahose api. https://developer.twitter.com/en/docs/twitter-api/enterprise/decahose-api/overview/decahose, 2021.

How Can Simulation-based Safety Testing Help Understand the Real-World Safety of Autonomous Driving Systems?

Fauzia Khan
Dept. of Computer Science
University of Tartu,
Tartu, Estonia
fauzia.khan@ut.ee

Laima Dalbina
Dept. of Computer Science
University of Tartu
Tartu, Estonia
laima.anna.dalbina@ut.ee

Hina Anwar
Dept. of Computer Science
University of Tartu
Tartu, Estonia
hina.anwar@ut.ee

Dietmar Pfahl
Dept. of Computer Science
University of Tartu
Tartu, Estonia
dietmar.pfahl@ut.ee

Abstract—**An Automated Driving System (ADS) requires exhaustive safety testing before receiving a road permit. Moreover, it is not clear what exactly constitutes sufficient safety for an ADS. One would assume that an ADS is safe enough if it is at least as safe as a Human Driven Vehicle (HDV). However, evaluating the safety of an ADS by comparing its behavior with that of a typical HDV in the real world is costly and risky. In this paper, we give an overview of our approach to compare the performance of ADS with HDV. While the overall approach is still in progress and ongoing, we provide a detailed approach utilizing established guidelines to systematically generate test scenarios specifically aimed at safety testing. Using our approach, various scenarios could be generated and tested, contributing to autonomous vehicles' trustworthiness.**

Keywords - Autonomous Driving System (ADS); Human Driven Vehicle (HDV); Safety Testing; Scenario Generation; CARLA Simulator.

I. INTRODUCTION

The automotive industry and research community are working hard to deploy autonomous cars on roads in the future. However, it requires exhaustive safety testing before it is safe at an acceptable level. Simulation-based testing is a cost-effective way to evaluate ADS safety. We aim to compare the behavior of an ADS to that of a Human Driven Vehicle (HDV) via simulation. However, the conclusions drawn from simulation-based testing are not always clear. There are some open challenges and questions, which are as follows:

Challenge 1 - How can we transfer the real world into a simulation model, and what aspects should be considered? To transfer the real world into a simulator, many challenges are posed. For example, how to model the behavior of an ADS and a human driver, how to set up the environment, and on top of that, which scenarios must be tested? Vehicles encounter a wide range of scenarios based on the combination of scenery, traffic and road objects, environment, road geometry, and maneuvers [1]. Additionally, the complexity of driving tasks and the uncertainty of the driving environment grows exponentially, translating into infinite scenarios that ADS could encounter. It is practically impossible to test every scenario using a simulator. Thus, it leads to a few interesting sub-questions: **Q1)** How are scenarios derived systematically? **Q2)** How to select the critical scenarios for safety testing? **Q3)** How to generate exemplary test scenarios showing that ADS behaves better than HDV. There are also challenges to modeling an HDV in the simulator - as part of the context in which the ADS moves. The model of an HDV depends on the behavior of a human driver. A human driver is diverse in behavior as different drivers make different decisions under the same surroundings and driving situation, mainly due to aggression, attention, and experience. Furthermore, a human driver also makes mistakes, violates rules, and takes evasive maneuvers to prevent accidents that could cause by the mistakes of other drivers. All these aspects create behavior complexity in a simulation environment and make comparing an ADS with an average HDV difficult.

Challenge 2 - How can we build trust in simulations? That is, how to guarantee that simulation results correctly represent the real-world behavior of an ADS. This could be achieved by running scenarios in the simulator where the real-world reference behavior is known. We could consider scenarios where we know the human driver has more information and is better than ADS. If the ADS in the simulator reproduces the reference behavior, it supports the assumption that the ADS is modeled correctly in the simulator. Typical reference behavior could be the braking of the ADS on a straight road with dry streets and good visibility in daylight. Based on the physical characteristics of the car and its speed, one can calculate at what distance the car should be able to stop without hitting the obstacle.

Challenge 3 - How can we quantify the advantage and disadvantages of an ADS compared to an HDV with an average driver? In certain aspects, ADS performs better than an HDV. ADS can potentially reduce the mistakes that human drivers make while driving. For example, ADS are never distracted (e.g., using cell phones, drunk, or tired); it has better perception (e.g., no blind spot), faster response time, and more precise brakes, acceleration, and steering control. However, ADS might not perform well when it comes to certain situations. For example, corner cases (rare situations or not expected to happen in the real-world). Also, ADS lacks intuition and instinct compared to human drivers, which affects the response of ADS

in a particular situation. For example, a pedestrian is standing near the edge of a sidewalk, seemingly distracted and showing signs of potential jaywalking. A human driver might intuitively recognize the pedestrian's behavior and anticipate the possibility of them stepping into the road unexpectedly. However, an ADS may struggle to accurately interpret the pedestrian's behavior and intentions.

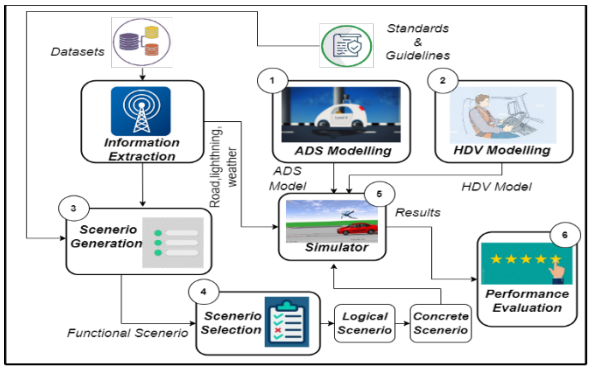


Figure 1. The proposed approach for comparing the behavior of ADS and HDV via Simulation-based Safety Testing.

In principle, we are still trying to resolve some of the above-mentioned challenges for translating the real world into the simulator. Figure 1 gives an overview of the proposed approach. As it is a work in progress, in this paper, we mainly explain steps in our approach for generating the test scenario systematically using public datasets by following the guidelines provided by the Center for Connect and Autonomous Vehicles [2] and existing literature [3], [4]. The work presented in this paper is related to the above-mentioned challenge # 1. More specifically, we tried to answer the sub-question Q1 by showing how a wide range of driving conditions can be effectively covered by systematically generating test scenarios, facilitating a better evaluation of ADS performance in diverse scenarios. It could also help identify potential risks and vulnerabilities of the ADS system, contributing to autonomous vehicles' overall development and trustworthiness.

II. Background

This section explains the key term "scenario" and abstraction levels of scenario representation.

Scenario: A specific situation or context that captures the essential elements of a particular driving experience or event. It is a quantitative description of the *ego vehicle*, its *activities*, *static environment*, and *dynamic environment* [5], [6].

Functional scenario: is the highest abstraction level that depicts possible situations on a real road as a brief text [7]. For example, in a highway merge scenario, the ego car merges from an on-ramp into highway traffic.

Logical Scenario: Functional scenarios are converted to logical scenarios by adding variables and parameter ranges [7]. To extend the previous example, the highway merge scenario incorporates different merge lane geometries, a range of surrounding vehicle speeds, varying traffic speed ranges, merging gaps, and timing. The combinations of all these attributes result in many logical scenarios.

Concrete Scenario: A concrete scenario refers to a specific instance of the logical scenario where precise values and conditions are defined for the various parameters of the scenario [7]. For example, specific road geometry, weather condition, speed, interaction time, etc.

III. Proposed Approach

Our proposed approach (see Figure 1) consists of six steps: (i) modeling an ADS in the simulator, (ii) modeling an HDV in the simulator, (iii) generating test scenarios, (iv) selecting test scenarios, (v) simulations, and (vi) performance evaluation. We give an overview of the first two steps in our approach. We explain step (iii), generating test scenarios, in more detail. Steps (iv) - (vi) are work in progress.

A. Modeling an ADS in Simulator

The choice of the simulator is important because it could significantly impact the reliability, accuracy, and validity of the results obtained from the performance evaluation of an ADS and HDV. The simulator must be capable of accurately replicating real-world conditions. Additionally, the ability to collect and analyze data on performance metrics such as travel time, fuel consumption, safety, and compliance with traffic rules is another important consideration when selecting a simulator. We opted for CARLA[1] because it possesses both capabilities [8]. To model an ADS in the simulator, the first step is to create a vehicle model by defining its physical properties, such as mass, dimensions, and color. The blueprint library *get blueprint_library* () function provided by the Carla Python API can be used to create the vehicle model. In the next step, sensors are integrated into the selected vehicle to enable perception of the surrounding environment. The desired parameters and sensors could be set in CARLA using *vehicle_blueprint.set _attribute* (). Using the above-mentioned steps, we model an ADS similar to a mini Cooper S with sensors such as a camera, Lidar, and radar. Additionally, the modeled ADS has all the built-in supporting systems that a regular HDV possesses.

B. Modeling an HDV in Simulator

To model an HDV in a simulator is challenging due to the diversity in human driver behavior, as individuals make different decisions when facing similar surroundings and driving scenarios. This is primarily attributed to variations in factors such as aggression, attention, and experience. We opt for CARLA's human-like driving behavior model. These models are based on real-world data distribution and include different driver behavior, e.g., aggressive, distracted, naturalistic, and low aggressive drivers. We select the human driver model by configuring the Non-Player Character (NPC) agent parameters in the CARLA configuration files. In the future, we plan to make different persona's according to each human-driver model, run a simulation, and compare the HDV behavior with an ADS.

[1] https://carla.org/

C. Generating Test Scenarios

To generate test scenarios, we perform the following steps: **1) Dataset Collection:** The first step is to collect a suitable dataset from public repositories based on the testing objectives. Choose a trustworthy source of high-quality datasets to ensure the dataset's reliability and quality. Formulate and run a search query using key terms to find relevant datasets. From the results, select a dataset that aligns with test objectives and contains the necessary information to generate functional scenarios. Keeping safety testing of ADS as objective, the necessary information a dataset should contain includes target object, provoking event, maneuver, etc.

The target object refers to the entity responsible for causing the accident, while the provoking event represents the specific action that triggered the accident. The accident directly affects the ego vehicle, and its corresponding driving situation is called the maneuver [9].

We formulated a search query based on our test objective to assess the advantages and disadvantages of ADS compared to HDV. We focused on key terms such as "road accident datasets" and "road accidents caused by human error" to find relevant datasets from the Kaggle repository. The chosen dataset[2] includes reports of traffic collisions in Addis Ababa, Ethiopia, spanning the period from 2017 to 2020.

2) Dataset Preprocessing: The next step is to preprocess and clean the dataset using the standard data preprocessing techniques to filter unnecessary, missing, and inconsistent data. In our case, the selected dataset is already preprocessed and clean. However, we refined the data by filtering the irrelevant attributes such as location, driver age, and experience. These attributes are unnecessary for generating functional scenarios. Additionally, we adjusted the column labels to align with our specific requirements using Python commands.

3) Feature Extraction and Categorization: Once data preprocessing is complete, elements for the functional scenario are extracted and categorized as the target object, provoking event, ego vehicle, and maneuver. Let us consider an example scenario where these elements are briefly explained to provide a clearer understanding.

Vehicle A abruptly changed the lane and hit Vehicle B, which was overtaking.

Here, Vehicle A is the target object, abruptly changing the lane is a provoking event, Vehicle B is the ego vehicle, and overtaking is a maneuver. We categorize the values of main elements (target object, provoking event, maneuver, road junction types) into subtypes based on their distinctiveness or uniqueness. We also merged similar categories to streamline the classification process. For instance, the categories of "Driving under the influence of drugs" and "Drunk driving" were considered similar, and thus, we merged them into a single category called "Drunk driver."

4) Generation of Functional Scenarios: In this step, the extracted main elements are combined to create a description known as a functional scenario. For example, we extract the Target object = animal, Maneuver = going straight, and Provoking Event = overspeeding. Combining these key elements, the functional scenario is "Vehicle A is going straight and overspeeding, hitting the animal." This step is repeated until all combinations of the defined category values are achieved.

The subsequent steps in the approach are in progress. We plan to develop a scenario selection method for safety testing to prioritize generated functional scenarios. The selected functional scenarios will be transformed into logical scenarios and refined into concrete scenarios by assigning specific values to the parameters. For example, information related to road junction type, weather conditions, lighting conditions, speed limits, and other relevant factors could be extracted from the dataset. Simulation results would be used i) to determine the reliability of the simulator and ii) to quantify ADS's advantages and disadvantages over the typical HDV.

IV. Preliminary Results

The results of steps (i) and (ii) of our approach will be published once the simulation is complete. We only show the preliminary result of step (iii) of our proposed approach for generating test scenarios from the chosen dataset (cf. Section III.C). Table I shows the extracted subtypes for each element of the functional scenario. We classify the target objects into seven categories, the provoking events into 18 subtypes, and the maneuvers into 11 subtypes. The total number of generated combinations without repetition is 1386. For each combination, it is possible to generate scenarios, However, not all scenarios are meaningful or relevant. We generated a list of functional scenarios using these combinations. The next step involves manually removing irrelevant or meaningless scenarios from the list. Due to space limitation, we only present a small initial set of generated functional scenarios in Table II. The last column in Table II shows the functional scenario after combining all the main elements captured in columns two to five.

TABLE I. Extraction of Functional Scenarios Elements and their Subtypes

Elements	Subtypes
Target object	Roadside-parked vehicles, Vehicles, Roadside objects, Animals, Rollovers, Pedestrians, and Train
Provoking event	Moving Backward, Overtaking, Changing lanes to the left, Changing lanes to the right, Overloading, No priority to a vehicle, No priority to pedestrian, No distancing, Getting off the vehicle improperly, Improper parking, Driving carelessly, Driving at high speed, Driving to the left, Over-speeding, Unknown, Overturning, Turnover, Drunk driving.
Maneuvers	Going straight, U-Turn, Moving Backward, Turnover, Waiting to go, Getting off, Reversing, Parked, Stopping, Overtaking, and Entering a junction.

[2] https://www.kaggle.com/datasets/saurabhshahane/road-traffic-accidents

V. FUTURE WORK

In the future, the generated test scenario will be prioritized and converted into logical and concrete scenarios. The concrete scenarios would be loaded in the simulation environment and executed. The results from the simulation would be used to quantify the performance of both ADS and HDV based on performance criteria. The possible metrics to evaluate the performance criteria could be the number of accidents, the severity of the accident, or parts of a vehicle damaged in simulated scenarios. We also plan to answer the many open questions and challenges (cf. section I) and run simulations that help identify the advantages or disadvantages of ADS behavior over that of HDVs in simulated safety-critical traffic situations and to translate these findings into corresponding real-world behavior.

TABLE II. AN INITIAL SET OF GENERATED TEST SCENARIOS AFTER APPLYING STEP (III) OF OUR PROPOSED APPROACH

Maneuvers	Target Object	Provoking Event	Functional Scenario
Driving Straight	Vehicle	Change lane to the left	The ego vehicle is driving straight. The target object (vehicle) is lane changing into ego's vehicle driving ahead.
Driving Straight	Vehicle	Overtaking	The ego vehicle is driving straight, while the target object (vehicle) is overtaking.
Reversing	Roadside objects	-	The ego vehicle is reversing and colliding with a roadside object.
U-Turn	Animal	-	The ego vehicle takes a U-turn and hits the animal.
Driving Straight	Pedestrian	Drive carelessly	The ego vehicle is driving straight carelessly while the target object (pedestrian) is crossing.

VI. RELATED WORK

Due to space limitations, we present only a few recent relevant studies related to simulation-based safety testing of ADS. Matthew et al. [10] presented a simulation framework based on an adaptive sampling method to test an entire ADS. Jha et al. [11] proposed a fault injection tool that systematically injects faults into the hardware and software of an ADS to evaluate safety and reliability. Ben et al. [12] presented an approach to test ADS in a simulation environment (Simulink). They used multi-objective search and surrogate models based on a neural network to identify critical test cases regarding an ADS behavior. Wicker et al. [13] performed black box testing to evaluate the robustness of neural networks against adversarial attacks in traffic sign recognition in self-driving cars. Our approach is different from existing work in simulation-based safety testing. We have a black-box point of view on the ADS when observing its behavior and focus on identifying ADS behavior that differs from that of an HDV, both positive and negative.

ACKNOWLEDGMENT

This work is supported by Estonian Research Council grant PRG1226, the Bolt Technology OU grant, and the Estonian state¨ stipend for doctoral studies.

REFERENCES

[1] F. Duarte and C. Ratti, "The impact of autonomous vehicles on cities: A review," *Journal of Urban Technology*, vol. 25, no. 4, 2018.

[2] "Uk standards: Natural language description for abstract scenarios for automated driving systems," *Specification-BSI Flex 1889v1.0:202207*,

[3] Y. Zhu, J. Wang, F. Meng, and T. Liu, "Review on functional testing scenario library generation for connected and automated vehicles," *Sensors*, vol. 22, no. 20, p. 7735, 2022.

[4] W. Ding, C. Xu, M. Arief, H. Lin, B. Li, and D. Zhao, "A survey on safety-critical driving scenario generation-a methodological perspective," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[5] H. Elrofai, D. Worm, and O. Op den Camp, "Scenario identification for validation of automated driving functions," in *Advanced Microsystems for Automotive Applications 2016: Smart Systems for the Automobile of the Future*, Springer, 2016, pp. 153–163.

[6] S. Geyer, M. Baltzer, B. Franz, S. Hakuli, M. Kauer, M. Kienle, S. Meier, T. Weißgerber, K. Bengler, R. Bruder, *et al.*, "Concept and development of a unified ontology for generating test and usecase catalogues for assisted and automated vehicle guidance," *IET Intelligent Transport Systems*, vol. 8, no. 3, pp. 183–189, 2014.

[7] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018, pp. 1821–1827.

[8] P. Kaur, S. Taghavi, Z. Tian, and W. Shi, "A survey on simulators for testing self-driving cars," in *Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*, IEEE, 2021.

[9] S. Park, S. Park, H. Jeong, I. Yun, and J. So, "Scenario-mining for level 4 automated vehicle safety assessment from real accident situations in urban areas using a natural language process," *Sensors*, vol. 21, no. 20, p. 6929, 2021.

[10] M. O'Kelly, A. Sinha, H. Namkoong, R. Tedrake, and J. C. Duchi, "Scalable end-to-end autonomous vehicle testing via rare-event simulation," *Advances in neural information processing systems*, 2018.

[11] S. Jha, T. Tsai, S. Hari, M. Sullivan, Z. Kalbarczyk, S. W. Keckler, and R. K. Iyer, "Kayotee: A fault injection-based system to assess the safety and reliability of autonomous vehicles to faults and errors," *arXiv preprint arXiv:1907.01024*, 2019.

[12] R. Ben Abdessalem, S. Nejati, L. C. Briand, and T. Stifter, "Testing advanced driver assistance systems using multi-objective search and neural networks," in *IEEE/ACM international conference on automated software engineering*, 2016.

[13] M. Wicker, X. Huang, and M. Kwiatkowska, "Feature-guided black box safety testing of deep neural networks," in *24th International Conference, Held as Part of the European Joint Conferences on Theory and Practice of Software, Greece*, Springer, 2018.

Applying Software Quality in Use Standards to Improve Scientific Software Selection

Yvette D. Hastings, Ann Marie Reinhold
Gianforte School of Computing
Montana State University
Bozeman, MT, USA
yvettehastings@msu.montana.edu, reinhold@montana.edu

***Abstract*—Across scientific domains, researchers are challenged by the process of selecting suitable modeling software. These challenges are particularly numerous in the earth sciences, arising from selecting software based on a few of the myriad complex earth system processes and wide availability of modeling software. Earth scientists lack a framework to guide scientific software selection. In this paper, we operationalize a framework based on the Quality in Use Model, as codified by the International Organization for Standardization (ISO) 25010 standard, to identify and create metrics to assess software that is used to simulate a subset of earth science processes known as soil processes. We applied this framework to assess software for three highly cited soil process models: Community Land Model (CLM), Decision Support System for Agrotechnology Transfer (DSSAT), and HYDRUS-1D. DSSAT scored the highest for the Quality in Use Model metrics, followed by HYDRUS-1D and CLM. This study is the first of its kind to apply the ISO 25010 Product Quality Model to a class of modeling software in the earth sciences, and its application shows promise for streamlining software selection.**

Keywords —ISO 25010: Quality in Use Model; software quality; soil process models; CLM; DSSAT; HYDRUS-1D

I. INTRODUCTION

End users in the earth sciences (here, earth-science end users (ESEUs)) are challenged by the process of selecting modeling software to meet their research needs. This is especially true for soil process models (SPMs), as they are foundational in many earth science research projects that investigate a wide array of processes in earth systems, including the effects of climate change and agriculture on the health and sustainability of soil and water systems [1]–[3]. While SPM software is widely available, SPMs have become computationally complex and mathematically complicated, as scientific software designers have attempted to keep step with advances in empirical research [2], [4]–[6].

These complexities present researchers with the challenge of determining which scientific modeling software best enables them to address their objectives efficiently and effectively. When selecting software, ESEUs must consider whether the SPM has the suitable spatio-temporal scales, whether the underlying mathematical algorithms are appropriate for their analysis, and whether the SPM is regularly maintained and updated with advancing scientific knowledge [1], [3], [4]. Further questions arise as to whether the ESEU has enough knowledge of how to use the modeling software (e.g., if the software is in a programming shell (e.g., Python or Java) or has a graphical user interface (GUI)) [7]. This knowledge impacts ESEU's experience with the software and can impact research results. For example, poor GUI design can result in ESEUs incorrectly setting up models or interpreting results incorrectly; in both cases, the validity and verification of simulations are seriously hampered.

ESEUs have an active forum on ResearchGate.net to assist one another with software selection [8], underscoring the multifaceted challenges associated with selecting and running SPM software. While literature review and peer commentary (e.g., forums) can provide useful information, ESEUs may still deliberate whether their chosen SPM software best addresses their research objectives or if another modeling software would have been a superior choice. Hence, we assert that ESEUs would benefit from considering a wide range of software-quality aspects when selecting modeling software.

The purpose of this paper is to operationalize a framework to evaluate the quality in use of SPM software to aid ESEUs in software selection. We identify aspects of the Quality in Use (QIU) Model, as codified and described, in the International Organization for Standardization (ISO) 25010 as a starting point for ESEUs to improve the efficiency of selecting, parameterizing, executing, and validating SPMs [9]. The Quality in Use Model identifies and defines five main components, with nine subcomponents, that software products aim to meet to allow users to meet their project goals [9]. Within each of these components and subcomponents, we developed metrics that can be applied to SPM software. We then used these criteria and metrics to evaluate three of the most utilized SPMs in Vereecken *et al*. [3]. Keeping the end user in mind, we also evaluate the software based on common considerations soil process modelers have.

II. METHODS

We conducted a systematic search for studies that use software to generate SPMs; the suite of potential software was constrained to those that generate one of the SPMs listed in Table 2 in [3] within the categories of water cycling, nutrient cycling, biological activity, salinization, buffering and filtering, recycling of waters, and biomass production for food, fiber and energy. We searched Google Scholar to determine which of

these SPM software had the greatest number of citations [19]–[21]. We evaluated the top three: Community Land Model (CLM; 25,000 citations), Decision Support System for Agrotechnology Transfer (DSSAT; 17,900 citations), and HYDRUS-1D (11,400 citations).

Below, we use the term "basic simulation" to describe the simple models that we executed in CLM, DSSAT, and HYDRUS-1D. Although each software creates SPMs, we could not identify a use case where we could simulate the same environmental system with the same parameters across all software. The reason for this is that differences among the intended use, scope, and parameters of the scientific modeling software prevented a direct comparison of resulting models. All simulations were run to completion without warning or error.

- CLM is a component of the Community Earth System Model (CESM) v2.1.3. Required software that is listed in the CESM2.1.z Quickstart Guide was installed on an Ubuntu virtual machine and the CESM v2.1.3 GitHub repo was cloned [10], [11]. Data stored on the GLADE drive of the ARC NCAR/UCAR Cheyenne server[1] was used for a basic simulation. This simulation was performed following the instructions in the CESM Quickstart Guide and from an online CESM tutorial [10], [12]. The only modification from the tutorial was to change the project identifier between building and submitting the model. This was not a step in the tutorial but was required to run the model.
- DSSAT v4.8.0.0 and associated GUI were downloaded and installed in accordance with the developer's instructions [13]–[16]. The 'Introductory Simulation', found in the 'Accessories' tab of the user interface, was followed to perform a basic simulation. We provided values for the required input fields for the test simulation.
- HYDRUS-1D v4.17.0140 and associated GUI were downloaded and installed from the developer's website [17]. Tutorials 1 and 3, available at [18], were used to create a basic simulation. Instructions and data for the tutorials, where applicable, were included on their respective webpages.

A. ISO 25010: Quality in Use

CLM, DSSAT, and HYDRUS-1D were evaluated using the five components, nine subcomponents, and provided definitions as codified in the Quality in Use Model of [9]. In accordance with implementing the QIU Model, our research group—composed of software engineers and ESEUs—held brainstorming sessions over multiple days to critically evaluate and identify measurable metrics to quantify each component and subcomponent based on an ESEU perspective. Our scoring scale was set from 1-10, with 1 being the lowest score and 10 being the highest score possible. We used this scale to keep score ranges universal for all components/subcomponents.

Components/subcomponents that assess a range of values (e.g., costs) were broken up into equal intervals bounded between 1 and 10 using one of the following equations:

$$y = \frac{x}{n} * 10, \quad (1)$$

$$y = 1 + \left(\frac{n-x}{n}\right) * 10, \quad (2)$$

where y is the metric score, n is the highest possible boundary value, and x is a value between $\frac{n}{10}$ and n. Components/subcomponents measured with a Boolean received a score of 10 for "true" and 1 for "false"; where appropriate, components/subcomponents not meeting a true or false answer received intermediate scores. Higher scores indicate higher QIU.

We created metrics for each component/subcomponent of the ISO 25010 QIU model (Fig. 1) as follows.

- **Effectiveness**: SPMs need to incorporate an array of processes interacting in the environment. We have listed three critical processes ESEUs must simulate. Using Eqn.1, the highest possible boundary value (n) is 3, where an x of 3 received a score of 10 and an x of 1 received a score of 3.33.
- **Efficiency**: **Costs** – Many government agencies and academic institutions use SPMs to study the environment. Often, funds to support software licensing are limited. SPM software we evaluated are free, but many others may require licensing fees. Using Eqn. 2, where n is $2,000 USD, licenses costing ≥ $2,000 USD received a score of 1 on our scoring scale, and licenses costing ≤ $200 USD received a score of 10. **Time** – High memory and CPU usage hinder the productivity of ESEUs. Using Eqn. 2, where n is 100%, CPU usage > 90% received a score of 1 and CPU usage ≤ 10% received a score of 10. CPU and memory usage are measured on the same scale. **Materials** – Software that requires multiple programs and dependencies to run reduces user experience, as these are often cumbersome and tedious to install. Using Eqn. 2, where n is 10, software requiring ESEUs to manually install ≥ 10 dependencies to run an SPM received a score of 1, and software requiring ≤ 1 dependency to be manually installed received a score of 10.
- **Satisfaction**: **Usefulness** – Not all ESEUs are familiar with a terminal window. Having a GUI can often improve user experience. **Trust** – Software that has been validated and verified improves user trust. **Pleasure** – Having available tutorials that follow model flow improves user experience and learnability of the software. **Comfort** – We identified two metrics to test this subcomponent. Metric 1 assesses if the software requires constant user input during a long model run, thus potentially reducing user experience. Metric 2 looks at how many commands/steps a user is required to go through from start to finish (i.e., parameterization to analysis). Metrics for usefulness, trust, pleasure, and the first comfort subcomponents were measured with a Boolean. For the second comfort metric, using Eqn. 1, where n is 5, software requiring ≥ 5 steps received a score of 2 and software requiring ≤ 1 step received a score of 10.
- **Freedom From Risk**: **Economic Risk Mitigation** – Cyber-attacks are costly. Having secure servers to run models and store data reduces cyber threats. **Health and Safety Risk Mitigation** – No obvious test metric exists to assess this subcomponent. Therefore, this is not included in our evaluation and is denoted as N/A in Fig. 1. **Environmental Risk Mitigation** – High energy use in the home/office is a risk to the environment. Here, we did not assess economic and environmental risk mitigation subcomponents, but they merit follow-up in future work.

[1]https://arc.ucar.edu/knowledge_base/70549542

This material is based upon work supported by the National Science Foundation under the following Grant Numbers: SitS CBET-2034430 and EPSCoR Cooperative Agreement OIA-1757351. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

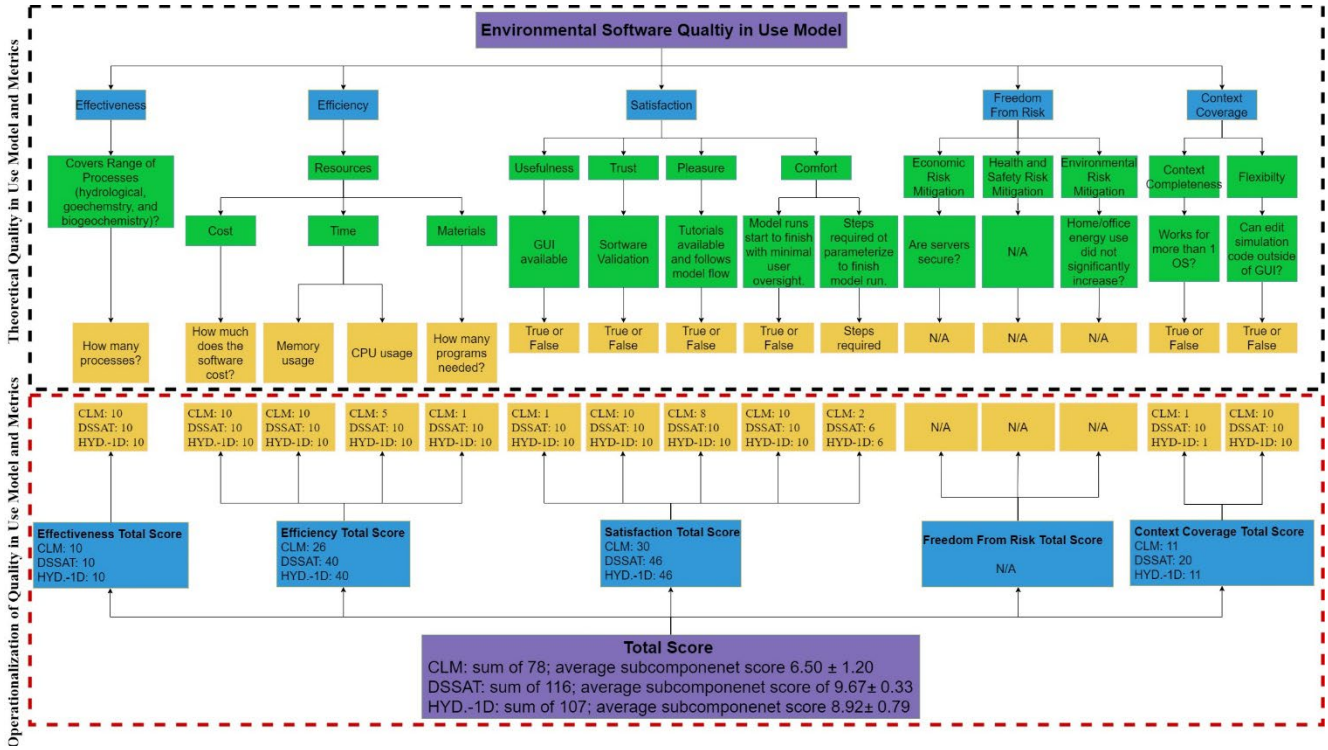


Figure 1. Quality in Use (QIU) Framework developed from the ISO 25010 Software Quality in Use Model. QIU Framework was used to evaluate the three SPM software: CLM, DSSAT, and HYDRUS-1D (abbreviated as HYD.-1D). Black-dashed box: the theoretical QIU Framework and metrics; dashed red box: operationalization of the theoretical QIU model and metrics; purple boxes: environmental software QIU model and operationalization scores; blue boxes: QIU components; green boxes: QIU subcomponents; yellow boxes: QIU metrics. Arrows indicate the flow of information.

- **Context Coverage**: **Context Completeness** – Software that runs on multiple operating systems (OSs) allows users the freedom to use computer environments (i.e., "contexts") that are familiar to them. **Flexibility** – ESEUs perform research covering a wide array of scientific theories. Code that is modifiable allows ESEUs to add/update algorithms that meet their research goals. Context completeness and flexibility are measured with a Boolean.

B. ESEU Considerations

CLM, DSSAT, and HYDRUS-1D were assessed to determine which ESEU primary considerations were met by each scientific modeling software. These common considerations overlap with certain components of the QIU Model; e.g., W*hich operating system is required to run the software? Is the software easy to learn and use, with ample tutorials? To what extent is run time a consideration? Can end users modify the simulation parameters using code (rather than a GUI) if needed?* In addition, ESEUs also must consider how models created by the software represent the earth system of interest; e.g., *How are hydrological processes simulated? Which chemical processes can be simulated and how so (kinetics, thermodynamics, etc.)? Is this software appropriately verified and validated for an environmental system of interest?*

III. Results & Discussion

Software popularity was not directly related to QIU scores. Although CLM had the most citations, DSSAT had the highest QIU score, with a total score of 116 points and an average subcomponent score of 9.67 ± 0.33 points (standard error). HYDRUS-1D received the second highest score in our QIU scoring index, with a total score of 107 points and an average subcomponent score of 8.92 ± 0.79 points. CLM received the lowest score of the three software, with a total score of 78 points and an average subcomponent score of 6.50 ± 1.20 points. These findings indicate DSSAT provides ESEUs with greater user experience related to ease of use and flexibility of the modeling software.

Comparison of common considerations of ESEUs across CLM, DSSAT, and HYDRUS-1D indicated that DSSAT had the greatest coverage (Table 1). DSSAT had 15 of the 18 assessed considerations listed, while HYDRUS-1D and CLM each had 13 of the 18 assessed considerations. What set DSSAT apart was that DSSAT can run on multiple OSs and platforms. Further, DSSAT is not limited to its GUI. ESEUs can run DSSAT in R and Python platforms [22], both of which can be operated in Windows and UNIX-like operating systems; neither CLM nor HYDRUS-1D had this option. Although DSSAT had the highest QIU scores (Fig. 1) and addresses the greatest number of ESEU considerations (Table 1), it is intended primarily for simulating agricultural systems. Consequently, it requires crop systems to be identified and associated parameter values to be inputted. Conversely, CLM and HYDRUS-1D are not limited to crop systems. This highlights the importance of identifying the intended purpose and scope of modeling software. Although intended purpose and scope are not explicitly part of the QIU framework presented here, they are related to the component of Effectiveness. While this could be viewed as a limitation of our operationalization of the QIU

Table 1. Common considerations for ESEUs. Software evaluated were CLM, DSSAT, and HYDRUS-1D (HYD-1D). Information is sourced from user manuals and software websites.

ESEU Consideration	CLM	DSSAT	HYD-1D
Operating Systems			
UNIX-like	x	x	-
Windows	-	x	x
Hypothesis Testing	x	x	x
Hydrological Mathematics			
Richard's Equation	x	-	x
Fickian Advection/Dispersion	-	-	x
Tipping Bucket	-	x	-
Reactive Soil Processes			
Biogeochemistry	x	x	x
Geochemistry	x	x	x
Modifiable Simulation Code	x	x	-
Time to Run Model	*	*	*
Spatial Scale			
Fine (Soil Column)	-	x	x
Moderate (Plot to Farm)	x	x	x
Coarse (Watershed to Global)	x	-	-
Parameterization Data			
Manual Entry	-	x	x
Data Upload	x	x	-
Learnability			
Tutorials	x	x	x
User Forums	x	x	x
Developer Contact	x	x	x
Validation	x	x	x
Correct Equations/Theories	*	*	*

*Not assessed in current study.

framework, this qualitative information was not readily quantifiable but was available on software websites, user manuals, and with minimal literature review.

The strength of our approach is in bridging between the computer sciences and earth sciences to operationalize a QIU framework to improve the software selection process for ESEUs. Our future work will incorporate the Product Quality Model codified by [9] into a similar framework, as well as comparing SPMs with common intended use and purposes through the lens of both QIU and Software Product Quality. Additionally, while the current investigation pertains to scientific modeling software in the earth sciences, the framework holds promise for assessing QIU in scientific software in other scientific disciplines and domains.

ACKNOWLEDGMENTS

We thank Drs. S.A. Ewing, C. Izurieta, R.A. Payn, D. Reimanis, and S. Warnat for their collaboration, and the NCAR/UCAR support team for account support and server access.

REFERENCES

[1] L. Li *et al.*, "Expanding the role of reactive transport models in critical zone processes," *Earth-Science Reviews*, vol. 165. Elsevier B.V., pp. 280–301, Feb. 01, 2017. doi: 10.1016/j.earscirev.2016.09.001.

[2] C. I. Steefel, D. J. DePaolo, and P. C. Lichtner, "Reactive transport modeling: An essential tool and a new research approach for the Earth sciences," *Earth Planet Sci Lett*, vol. 240, no. 3–4, pp. 539–558, Dec. 2005, doi: 10.1016/j.epsl.2005.09.017.

[3] H. Vereecken *et al.*, "Modeling Soil Processes: Review, Key Challenges, and New Perspectives," *Vadose Zone Journal*, vol. 15, no. 5, p. vzj2015.09.0131, May 2016, doi: 10.2136/vzj2015.09.0131.

[4] J. Carrera, M. W. Saaltink, J. Soler-Sagarra, W. Jingjing, and C. Valhondo, "Reactive Transport: A Review of Basic Concepts with Emphasis on Biochemical Processes," *Energies (Basel)*, vol. 15, no. 3, Feb. 2022, doi: 10.3390/en15030925.

[5] C. I. Steefel, S. B. Yabusaki, and K. U. Mayer, "Reactive transport benchmarks for subsurface environmental simulation," *Computational Geosciences*, vol. 19, no. 3. Kluwer Academic Publishers, pp. 439–443, Jun. 27, 2015. doi: 10.1007/s10596-015-9499-2.

[6] D. Wang *et al.*, "A scientific function test framework for modular environmental model development: Application to the community land model," in *Proceedings - 2015 International Workshop on Software Engineering for High Performance Computing in Science, SE4HPCS 2015*, Institute of Electrical and Electronics Engineers Inc., Jul. 2015, pp. 16–23. doi: 10.1109/SE4HPCS.2015.10.

[7] F. J. R. Meysman, J. J. Middelburg, P. M. J. Herman, and C. H. R. Heip, "Reactive transport in surface sediments. I. Model complexity and software quality," *Comput Geosci*, vol. 29, no. 3, pp. 291–300, 2003, doi: 10.1016/S0098-3004(03)00006-2.

[8] "How to choose reactive transport modeling software." ResearchGate.net. https://www.researchgate.net/post/How_to_choose_Reactive_Transport_Modeling_software (accessed May 05, 2023).

[9] *Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)-System and software quality models*, ISO/IEC FDIS 25010:2010(E), International Organization for Standardization, Geneva, CH, 2010.

[10] National Center for Atmospheric Research. "CESM Quickstart Guide (CESM2.1)." CESM2. https://escomp.github.io/CESM/versions/cesm2.1/html/ (accessed Jun. 04, 2023).

[11] *CESM*. (v2.1.3), National Center for Atmospheric Research. Accessed: June 8, 2023. [Online]. Available: https://www.cesm.ucar.edu/models

[12] National Center for Atmospheric Research. "Welcome to the CESM Tutorial." https://ncar.github.io/CESM-Tutorial/README.html (accessed Jun. 25, 2023).

[13] J. W. Jones *et al.*, "DSSAT Cropping System Model," *European Journal of Agronomy*, vol. 18, pp. 235–265, 2003, doi: 10.1016/S1161-0301(02)00107-7.

[14] G. Hoogenboom *et al.*, "Advances in crop modeling for a sustainable agriculture," *The DSSAT crop modeling ecosystem*, pp. 173–216, 2019, doi: 10.19103/AS.2019.0061.10.

[15] G. Hoogenboom *et al.*, "Decision Support System for Agrotechnology Transfer (DSSAT) Version 4.8," 2021. DSSAT.net

[16] DSSAT. (v4.8.0.0), DSSAT Foundation, Inc. Accessed: June 8, 2023. [Online]. Available: dssat.net

[17] *HYDRUS-1D*. (v4.17.0140), PC-Progress s.r.o. Accessed: May 31, 2023. [Online]. Available: https://www.pc-progress.com/en/Default.aspx?H1d-downloads

[18] "Hydrus-1D Tutorial Book." PC-Progress. https://www.pc-progress.com/en/Default.aspx?h1d-tutorials (accessed May 31, 2023).

[19] "CLM Google Scholar Results." Internet Archive WaybackMachine. https://web.archive.org/web/20230523152409/https://scholar.google.com/scholar?hl=en&as_sdt=0%2C27&q=Community+Land+Model+%28CLM%29&btnG= (accessed May 23, 2023).

[20] "DSSAT Google Scholar Results." Internet Archive WaybackMachine. https://web.archive.org/web/20230523151936/https://scholar.google.com/scholar?hl=en&as_sdt=0%2C27&q=%22DSSAT%22&btnG= (accessed May 23, 2023).

[21] "HYDRUS 1D Google Scholar Results." Internet Archive WaybackMachine. . https://web.archive.org/web/20230523153823/https://scholar.google.com/scholar?hl=en&as_sdt=0%2C27&q=%22HYDRUS+1D%22&btnG= (accessed May 23, 2023).

[22] P. D. Alderman, "A comprehensive R interface for the DSSAT Cropping Systems Model," *Comput Electron Agric*, vol. 172, May 2020, doi: 10.1016/j.compag.2020.105325.

An Evaluation of Word Embeddings on Vulnerability Prediction with Software Metrics

Sousuke Amasaki, Tomoyuki Yokogawa
Department of Systems Engineering
Okayama Prefectural University
Soja, Japan
{amasaki,t-yokoga}@cse.oka-pu.ac.jp

Hirohisa Aman
Center for Information Technology
Ehime University
Matsuyama, Japan
aman@cse.oka-pu.ac.jp

Abstract—**CONTEXT: Software vulnerability is a crucial risk for a digital world. Developers dedicate enormous effort to removing vulnerable code from their software products. Vulnerability prediction aims to spot which modules are more vulnerable using software metrics. Recent studies conducted empirical experiments using textual information and software metrics. The result showed that the textual information did not help improve the predictive performance. However, their evaluations only considered Bag-of-Words (BoW) as textual information, and semantic relations among words have never been examined. OBJECTIVE: To examine the performance of vulnerability prediction with textual information considering semantic relations. Word2Vec was employed for capturing semantic relations. METHOD: A comparative study among BoW and two Word2Vec embeddings was conducted. For easy evaluation, we replicated a recent study that employed BoW. The Word2Vec embeddings were obtained from pre-trained models based on Google News and Stack Overflow. The former used large but non-SE-related texts, while the latter used small but SE-related texts. RESULTS: The non-SE Word2Vec improved vulnerability prediction in term of prediction stability. The SE-specific Word2Vec was less effective. CONCLUSION: Practitioners should consider textual information with non-SE Word2Vec for better vulnerability prediction.**

Keywords-component; vulnerability prediction; word embeddings; empirical study

I. INTRODUCTION

Software vulnerability prediction is a sort of defect prediction that uses software metrics such as complexity measures without analyzing program syntax and semantics. Contrastingly, vulnerable code pattern recognition analyzes program syntax and semantics to obtain textual and/or structural information from source code and construct prediction (recognition) models. These two categories use supervised/unsupervised learning algorithms, and the information of those categories can be combined for better prediction and recognition.

A recent study [1] conducted an empirical experiment that combined software metrics and textual information and used them for vulnerability prediction. The result showed that their textual information was seldom helpful in improving the predictive performance. However, they used classical Bag-of-Words (BoW) for textual information, and modern techniques that can consider semantic relations among words have not been employed for evaluation.

In this study, we investigated the effectiveness of a modern technique for textual information for vulnerability prediction. Our empirical study employed a replication kit of [1] that combined process and product metrics and BoW for vulnerability prediction. We used Word2Vec [2] as a modern technique for textual information instead of BoW. Word2Vec is a method to learn quality distributed representations for words. Words are represented as multi-dimensional vectors, and similar words are placed close together in that dimensional space. Our study used two pre-trained models from SE and non-SE texts to obtain Word2Vec vectors for evaluation.

II. METHODOLOGY

We investigated three research questions. The questions and their motivations are described in the following subsections.

A. Datasets

This study used the datasets provided through a replication package of a past study [1]. The datasets were collected from 9 Java projects. We used the same sample of 8,991 commits among 56,286 for evaluation. For each commit, 24 product metrics and 9 process metrics were collected. Their details such as definitions are in [1]. Also, our study collected BoW and Word2Vec vectors for evaluation.

B. Word2Vec-based features

This study defined two types of Word2Vec-based vectors as well as BoW-based vectors in [1]. One vector is the sum of Word2Vec vectors of the files in a commit. Each file modified in a commit was parsed to extract identifiers. Here, camel cases and snake cases were divided into individual words. These words were lowered and converted to Word2Vec vectors. The sum of the vectors was used as the vector of the file. The same procedure was used to make Word2Vec vectors of added lines and deleted lines of a commit. The other vector is the absolute difference between the Word2Vec vectors of added lines and deleted lines. Finally, the two types of vectors were concatenated to make Word2Vec-based features.

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors).*

In the above procedure, we used pre-trained Word2Vec models to convert words to vectors. Pre-trained models are based on a large amount of training data and are expected to learn many words and their relations that would not be able to realize from a small amount of data of 8,991 commits and the modified files in those commits. For that purpose, we prepared two pre-trained Word2Vec models.

The first pre-trained mode was trained with Google New dataset. The training data was obtained from non-SE documents, and the similarity among words differs from that of software engineering documents. The vector size is 300, and the vocabulary size is approximately 3 million words. The second model [3] was trained with 15GB of Stack Overflow posts (i.e., SE documents) collected from August 2008 to December 2017. All code snippets were removed before training, and natural language terms only appeared in the textual data. Although some symbols such as `+' are often used in a software engineering context (e.g., programming), we did not consider those terms as identifiers in Java code does not include them. The pre-trained model was trained with 1.7 million keywords. The vector size is 200, smaller than the first model.

C. Experimental Design

As noted in [1], the number of vulnerabilities is too small in some projects to conduct neither cross-validation nor a time-sensitive approach. Therefore, cross-project training was adopted. The cross-project training of N projects uses N-1 project data for training and predicts the remaining project. This experiment design finally made N predictive results.

D. Prediction Models

We used the same prediction models as [1], namely, SVM, KNN, Decision Tree, Random Forest, Extremely Randomized Trees, AdaBoost, Gradient Boosting, and XGBoost with hyper-parameter tuning.

E. Performance Measure

AUC (Area Under the Receiver Operating Characteristic Curve) was adopted for evaluation as well as [1]. AUC measure is not sensitive to thresholds that determine faultiness or not. It is also robust to class imbalance, which happens in vulnerability prediction. A classifier is perfect if an AUC value is 1.0. A meaningful classifier results in an AUC value of more than 0.5. If equal or less, the classifier is random guessing at best.

III. Research Questions

We investigated the following research questions. The questions and their motivations are described in the following subsections.

A. Effectiveness of Non-SE Word2Vec against BoW

Although the merits of word embeddings were confirmed in some software engineering tasks, past studies such as [1] on vulnerability prediction have adopted BoW only. The pre-trained model based on the Google News dataset was the most popular one and was suited to a baseline for evaluation. This research question aimed to answer whether using such a general-purpose pre-trained model is effective for vulnerability prediction compared to using BoW.

B. Effectiveness of SE-Specific Word2Vec against BoW

Efstathiou et al. [3] examined the effectiveness of domain-specific word2vec models compared to non-SE models. They demonstrated that the similarity of technical words was often different from general words. For example, “cookie” was like “session” in software engineering while it was to “cupcake” in general. This difference in specialty might also affect the predictive performance of vulnerability prediction. Therefore, this research question aimed to answer whether using a domain-specific model is effective for vulnerability prediction.

C. Contribution of Word2Vec Features on Vulnerability Prediction with Software Metrics

Two studies [1][4] introduced BoW into conventional vulnerability prediction based on software metrics. One of their interests was the effectiveness of textual information though they did not investigate other representations of textual information such as Word2Vec. Therefore, this research question aimed to answer whether word embeddings could add value for vulnerability prediction with software metrics.

IV. Results

This section answers the three research questions posed in Section III. The results were also discussed.

A. Effectiveness of Non-SE Word2Vec against BoW

Fig. 1 represents the result of cross-project validation with the prediction methods we employed. There are three groups corresponding to the types of textual information, namely, BoW, non-SE Word2Vec, and SE-specific Word2Vec. Each boxplot in each group represents the predictive performance of vulnerability prediction using a prediction method we employed. To answer RQ1, this subsection discussed the differences in predictive performance between BoW (the leftmost group) and the non-SE Word2Vec (the middle group).

The leftmost group in the figure showed the AUC of the prediction methods using BoW. It was observed that the prediction methods affected the predictive performance. Decision Tree, Random Forest, Extra Trees, and XGBoost were lesser than random guessing with AUC values of almost less than 0.5. SVM and Gradient Boost were better than random guessing with mean and median values of more than 0.6. KNN showed higher performance with AUC values around 0.7. AdaBoost was the best performer in terms of mean and median values.

The middle group in Fig. 1 showed the AUC of the prediction methods using the non-SE Word2Vec. The predictive performance of the prediction methods in this group was also diverse but showed slightly different trends from the leftmost group. While Decision Tree and XGBoost were still lesser than random guessing, Random Forest and Extra Trees performed better than those with BoW. The boxplot of SVM was not so different from that of SVM in the leftmost group. However, their mean and median values were improved with non-SE word embeddings. Gradient Boost and KNN were also improved with

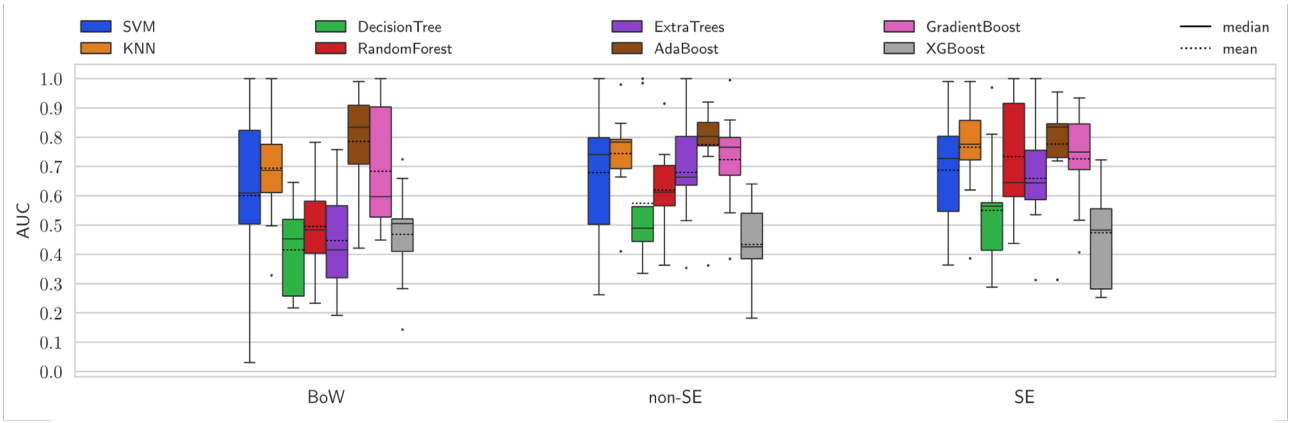


Figure 1. The predictive performance of vulnerability prediction methods with textual information

the use of the non-SE Word2Vec. Their boxplots got shorter than those with BoW. Although AdaBoost was still the best performer in terms of mean and median values, using the non-SE Wor2Vec worsened the median value compared to BoW. These observations implied that using the non-SE word embeddings was supported in terms of AUC. While the best performer in terms of median was AdaBoost with BoW, AdaBoost with non-SE Word2Vec showed good and stable performance.

In summary, using the non-SE Word2Vec was better than using BoW for Random Forest, Extra Trees, Gradient Boost, and KNN. The best performer in terms of median is AdaBoost with BoW while the statistical test did not reject the difference against AdaBoost with non-SE Word2Vec. The advantage of non-SE Word2Vec was in the stability represented by the smaller boxplot.

B. *Identify the Headings Effectiveness of SE-Specific Word2Vec against BoW*

To answer RQ2, the differences in predictive performance between the non-SE Word2Vec and the SE-specific Word2Vec were discussed in this subsection. The rightmost group in Fig. 1 showed the AUC of the prediction methods using the SE-specific Word2Vec. The predictive performance of the prediction methods in this group was also diverse but showed different trends from the middle group. The boxplots of Decision Tree and XGBoost crossed the line of random guessing. However, the mean and median of Decision Tree were improved and got beyond 0.5. Random Forest also showed better results regarding the upper side of the boxplot and the mean value. Extra Trees with the SE-specific Word2Vec showed slightly lower performance than that with the non-SE Word2Vec. The performance of SVM, KNN, AdaBoost, and Gradient Boost was stable. No clear effect of using SE-specific Word2Vec was observed. AdaBoost with the SE-specific Word2Vec was competitive with that with the non-SE Word2Vec.

In summary, using SE-specific word embedding was also a better option than using non-SE word embeddings for Decision Tree, Random Forest, and AdaBoost. Although the best performer was AdaBoost using the SE-specific Word2Vec, it was not clearly supported.

C. *Contribution of Word2Vec Features on Vulnerability Prediction with Software Metrics*

Fig. 2 showed boxplots of the AUC of the prediction models. The leftmost group in Fig. 2 showed the AUC of the prediction methods using BoW. In comparison to Fig. 1, adding software metrics improved the predictive performance of some prediction methods. The mean and median values of Decision Tree, Random Forest, Extra Trees, Gradient Boost, and XGBoost were improved. The boxplots of them also moved to higher places. The middle group in Fig. 2 showed the AUC of the prediction methods using the non-SE Word2Vec. In comparison to Fig. 1, adding software metrics improved the predictive performance of some prediction methods. The mean and median values of Decision Tree, Random Forest, Extra Trees, AdaBoost, Gradient Boost, and XGBoost were improved. The boxplots of them also moved to higher places. The rightmost group in Fig. 2 showed the AUC of the prediction methods using the SE-specific Word2Vec. In comparison to Fig. 1, adding software metrics improved the predictive performance of some prediction methods. The mean and median values of Decision Tree, Random Forest, Extra Trees, AdaBoost, Gradient Boost, and XGBoost were improved. The boxplots of them also moved to higher places. These observations implied that using word embeddings was supported in terms of AUC for some prediction methods while the statistical test did not reject the difference. Using non-SE Word2vec contributed to stable prediction with competitive performance. AdaBoost with non-SE Word2Vec and software metrics was considered a practically best choice for its stability.

V. THREATS TO VALIDITY

We investigated the following research questions. The questions and their motivations are described in the following subsections.

A. *Internal Validity*

The main threat is in the implementation of the experiment we conducted. We used the same experimental design as the replication package of [1] to minimize that threat. We only modified some code to use Word2Vec vectors for evaluation. We executed the original experiment with that code, and even

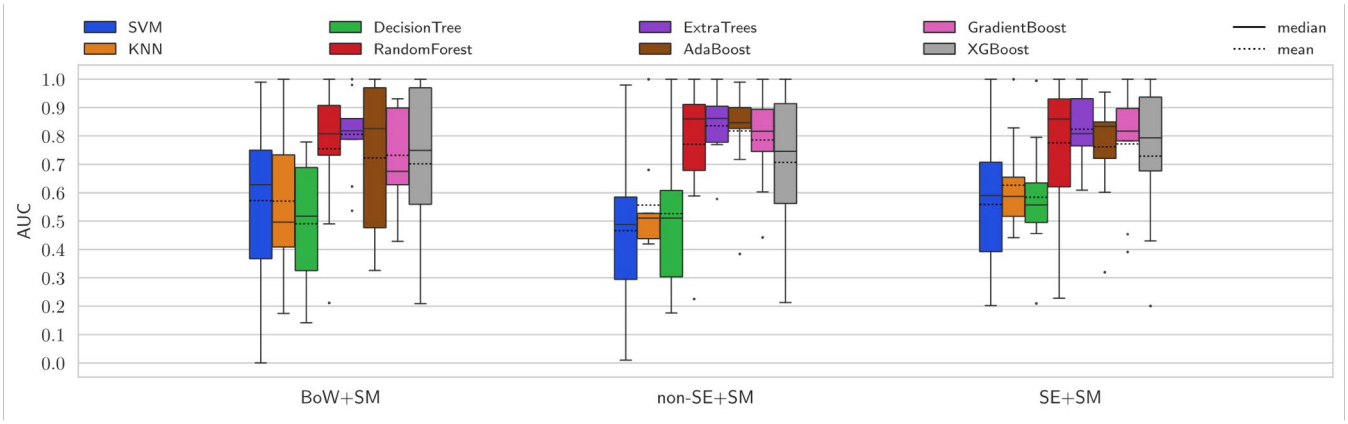


Figure 2 The predictive performance of vulnerability prediction methods with textual information and software metrics

if any bugs existed, we could evaluate the effectiveness of Word2Vec under the same condition as BoW.

B. *Construct Validity*

The vulnerable contributing commits (VCCs) of the datasets were labeled automatically with the SZZ algorithm. The SZZ algorithm was widely used for defect prediction but was not perfect. Although some mislabeling might thus exist, manual labeling was unrealistic for a large number of commits. For word embeddings, we used two Word2Vec pre-trained models only. The effectiveness of other word embeddings, such as Glove, was still unknown.

C. *External Validity*

This study used 9 Java datasets only. Although their domain and the statistics, such as commit counts, are diverse, the small number of projects still threats external validity. However, we also think that threats to external validity were not more serious than those of past studies using fewer datasets.

VI. CONCLUSIONS

This study conducted an evaluation of word embeddings on vulnerability prediction with software metrics. Using 9 Java projects with vulnerabilities registered in NVD, two Word2Vec pre-trained models were compared with the classical bag-of-words. The experiment results suggested the use of non-SE Word2Vec model was better than the use of BoW in term of stability. No clear difference in terms of mean was observed. Note that the effect was not found on all the prediction methods.

Our future work includes the use of additional information. Vulnerable code pattern recognition studies have used N-gram of source text in addition to BoW. Furthermore, recent studies have focused on deep learning-based features other than word embeddings. For instance, Chakraborty et al. [5] examined a deep learning-based technique based on graph embeddings and triplet loss with a dataset they collected from C/C++ code. Seeking better features is one of the top priority issues for better vulnerability prediction. An implication for practitioners is that trying to use word embeddings is a good idea for vulnerability prediction.

ACKNOWLEDGMENT

This work was partially supported by JSPS KAKENHI Grant #21K11831, #21K11833, and #23K11382.

REFERENCES

[1] F. Lomio, E. Iannone, A. De Lucia, F. Palomba, and V. Lenarduzzi, “Just-in-time software vulnerability detection: Are we there yet?” The Journal of Systems & Software, vol. 188, p. 111283, 2022.

[2] T. Mikolov, K. Chen, G. Cornado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” in Proc. of Workshop at the International Conference on Learning Representations, 2013.

[3] V. Efstathiou, C. Chatzilenas, and D. Spinellis, “Word embeddings for the software engineering domain,” in Proc. of Working Conference on Mining Software Repositories, ser. MSR ’18. ACM, 2018, p. 38–41.

[4] H. Perl, S. Dechand, M. Smith, D. Arp, F. Yamaguchi, K. Rieck, S. Fahl, and Y. Acar, “Vccfinder: Finding potential vulnerabilities in open-source projects to assist code audits,” in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: ACM, 2015, p. 426–437.

[5] S. Chakraborty, R. Krishna, Y. Ding, and B. Ray, “Deep learning based vulnerability detection: Are we there yet?” IEEE Transactions on Software Engineering, vol. 48, no. 09, pp. 3280–3296, 2022

A simple embedded system for solar tracking

Danko Milić
University of Montenegro
Faculty for Electrical Engineering
Podgorica, Montenegro
danko.milic@t-com.me

Radovan Stojanović
University of Montenegro
Faculty for Electrical Engineering
Podgorica, Montenegro
stox@ucg.ac.me

Abstract— **This paper presents a low-cost hardware-software alternative for tracking the sun position, for purposes of improving the performances of solar voltaic modules. The system ensures the optimal coordinate position of the photo voltaic modules to achieve the reception of maximum solar radiation. As a result, the solar panel can produce more electricity. The system is based on ATmega328P microcontroller, which control the movement of two servo motors along two axes. The rotation position and related actions are performed by off-the-shelf microcontroller, taking in account the inputs from the four light-dependent resistors (LDRs). The prototype is developed and tested in real conditions. Some of the results are presented and discussed. The main advantage of the system is its low-cost, reliability and logical scalable design.**

Keywords: Dual Axis Solar Tracker, Solar Power, Photovoltaic Solar Panel, Arduino Uno Development Board, ATMEGA 328 AVR Micro Controller, LDR.

I. INTRODUCTION

With the inevitable shortage of fossil fuel sources in the future, renewable energy sources have become a topic of interest for researchers, technicians, investors and decision makers worldwide. Solar energy, wind energy, geothermal energy, tidal energy, hydropower and bioenergy, due to their renewable nature, are considered a favorable substitute for fossil fuels.

In 2015, when the Paris Climate Agreement was signed, 5% of the world's electricity was produced from solar and wind energy, growing to 10% in 2021, exceeding the nuclear electricity production for the first time.

By the date 2022, global electricity production is still fossil fuels dominated, 61%. Coal accounted for 36% (10,186TWh), fossil gas 22% (6,336TWh) and other fossil fuels 3% (850TWh) of global production. Hydro remained the largest clean source of electricity with 15% (4,311TWh), and nuclear the second largest clean source with slightly more than 9% (2,611TWh). Wind and solar energy together reached a share of 12% of global electricity (3,444TWh), wind with 7.6% (2,160TWh) and solar energy with 4.5% (1,284TWh).

According to the scenario of net zero CO2 emissions of the International Energy Agency (IEA), to limit global warming to 1.5C, solar electricity production should increase to 7,552 TWh per year by 2030. This requires the growth of solar energy production by 25% per year, and the share of solar in global electricity production would reach 20% in 2030, compared to the current 4.5% achieved in 2022.

To achieve this growth, countries must continually increase their annual solar production targets. For example, in 2023 this would require 318TWh of additional solar electricity generation, while in 2030 the increase would be at least 1,500TWh. IEA Net zero scenario, Figure 1, predicts that by 2040 the energy sector should be at the level of net zero CO2 emissions and the share of solar electricity production will amount to around 40% [1]

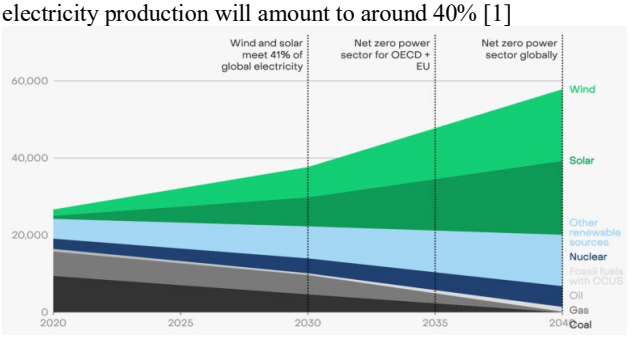


Figure 1. IEA Net zero scenario (EMBER, Global Electricity Review 2023), from [1]

Although there is continuous improvement in photovoltaic materials and current technology provides the cells with an efficiency level of up to 23.5%. Some projects have reached efficiencies higher than 40%, but they are not yet commercially available. Therefore, in order to reduce the price per produced KWh of energy, the efforts rely on the dimensions of the panels and/or the radiation intensity. Increasing the area of solar panels is not a sustainable solution. This increases investment costs and requires larger areas of land. However, the alternative is to maximize the energy extraction from the panels by using the array of cells to their full potential. Of course, all under the condition that it is profitable to invest in an automatic tracking system, bearing in mind that these systems are more expensive to build and maintain. In this case, the exposing the panel surface perpendicular to the sun's rays should be enabled. It is sun tracker strategy, based on Lambert's law, and one or dual axes movement/tracking. There are a number of systems for these purposes, with different features and prices as well as publications presenting related research [2],[3],[4],[5].

This paper proposes a solution that is suitable for companies that base their business on mixed system integration. Individual components are developed and

installed as needed, normally if this does not exceed the cost of supply and training.

II. HARDWARE DESIGN

A. General description

The working principle of the system is as follows: the solar panel is positioned temporally and spatially so that its surface is maximally exposed to solar radiation. As a result, it becomes more efficient and produces more electricity.

The system architecture is given in Figure 2. The position of the panel in X-Y plane is achieved by two servo motors, which rotate the solar panel over two axes, based on the inputs received from the four photo sensors located in plane of panel, one in each quadrant of the coordinate system Each LDR sends an equivalent signal of its corresponding resistance value to the microcontroller. The difference between the LDR voltages is used to drive the servo motor, by implementing adequate control algorithm.

One of the two DC servo motors is mechanically connected to the drive shaft of the other so that the first will move with the rotation of the other's shaft. In controlling DC servo motors numerous strategies can be done as example PWM, generated by microcontroller or FPGA [6], [7].

Servo motor shafts are used to drive the solar panel mounted on the solar tracking set [8]. The servo motors are arranged in such a way that the solar panel can move along the X-axis (east-west, left-right) as well as the Y-axis (north-south, up and down).

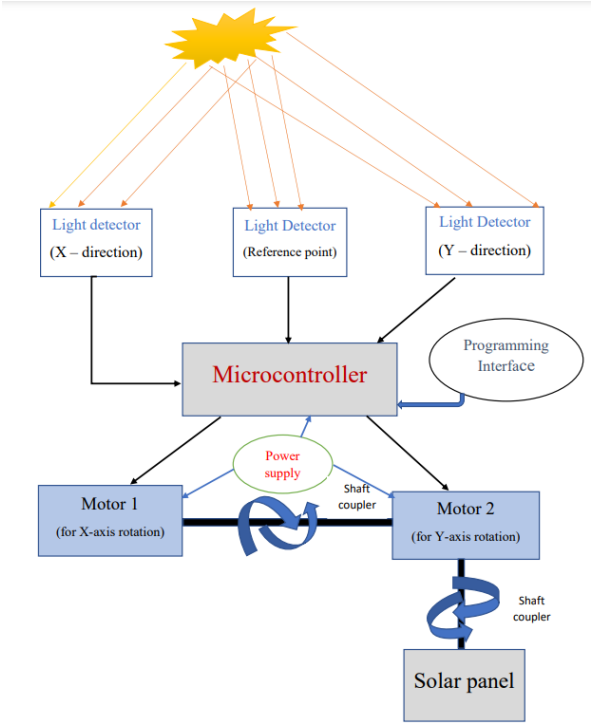


Figure 2. Hardware model block diagram

B. Sensing

The sensitivity curve of LDR is not linear, more exponentials. Figure 3.

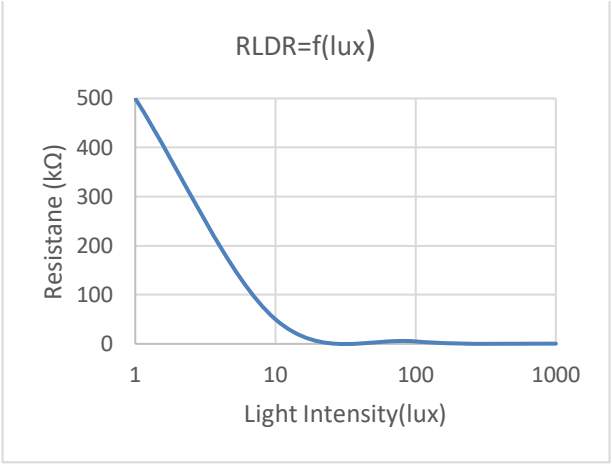


Figure 3. LDR resistance dependence of the illumination

When exposed to the light, the resistance ranges from 5-10 KΩ. For the dark its resistance is of MΩ value. In the simplest applications the LDR is connected in voltage follower circuit, Figure 4, and Vo is changes, increases or decreases, depending on how the LDR is connected. For case on Figure 4, Vo=VCC(5V)*LDR1/(R1+LDR1) that means for maximum illumination, Vo will weigh above 5V and for minimum above 0V, i.e. in digital equivalent, 1024 and 0, encoded as 10-bits.

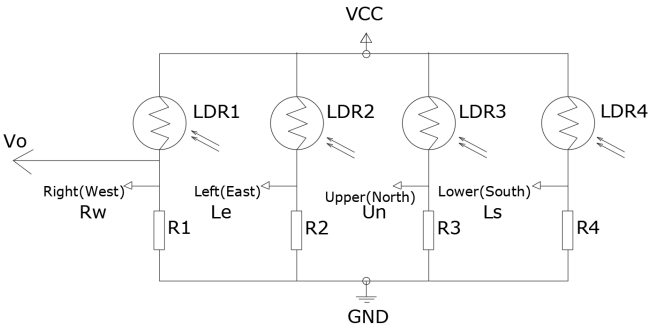


Figure 4. Quadrant of LDR photo sensors.

Rw, Le, Un and Ls are equivalent radiation information from Right(West), Left(East), Upper(North) and Lower(South) sensors.

Other sensors are also used in the system such as temperature sensor, humidity sensor, and light intensity sensor. They provide additional information, the values of which are displayed on LCD or used as needed.

C. System integration diagram

Figure 5 shows basic system diagram for proposed a two-axis solar tracker by using Arduino UNO development board powered by 5V, and connected to the LDR photo sensors. Servo X, rotates the solar panel round X axis. Servo Y: rotates the solar panel round Y axis.

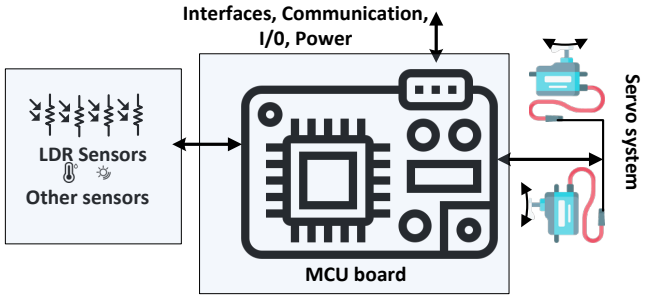


Figure 5. Basic circuit diagram

III. SOFTWARE DESIGN

The software implementation consists of coding the tracking system algorithm in the Arduino IDE environment and loading the code into the microcontroller. A software flowchart is given in Figure 6. The algorithm is based on comparing analog voltage values provided by photo sensors (left, right, upper and lower LDR). For azimuth tracking, the voltage values from the left and right LDR are compared. If the left (east) LDR receives more light, the voltage on the left LDR will be greater than the voltage on the right LDR and the horizontal motor will rotate to the left (east). If the right (west) LDR receives more light, the voltage on the right LDR will be higher than on the left and the horizontal servo motor will move in that direction, i.e. to the right (towards the west). To monitor the tilt angle, the voltage value from the upper and lower LDR is compared. If the upper (north) LDR receives more light, the voltage on the upper LDR will be greater than the voltage on the lower LDR and the vertical motor will rotate upwards (north). If the lower (south) LDR receives more light, the voltage on the lower LDR will be higher than on the upper LDR and the vertical servo motor will move down (south). In order to avoid the constant movement or vibration of the motor, a minimum voltage difference ("error") was introduced, which should be smaller than the voltage difference between the two sensors in order for the motor to start.

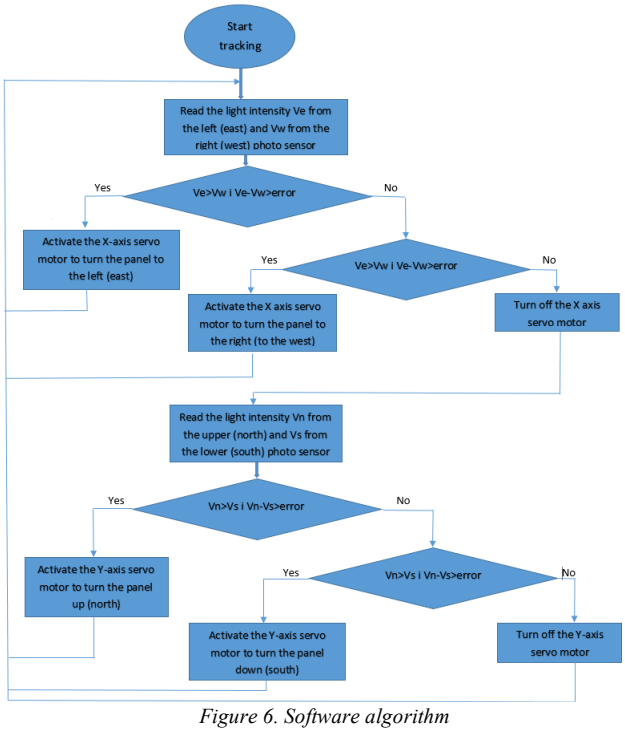


Figure 6. Software algorithm

IV. SYSTEM INTEGRATION

A. Testing the components

The components are connected to prototype, which back and front view are given in Figure 7 and Figure 8. which are connected to the development board, and the board to the computer for purpose of system initialization and testing. After testing all the components, assembly of the solar tracking was started by mounting the components on perforated acrylic plates. Keyestudio solar tracking kit is a good platform for time and performance efficient design.

The development board, LDR photo sensors (East, West, North), temperature and humidity sensor, energy module, servo motor Y, servo motor X, and Solar panel (rear view) are presented on the figure 7.

LCD screen, power module, Light intensity sensor, LDR photo sensors (South), and Solar panel (front view) are presented on the figure 8.

The motors are mounted on a moving mechanism on which a mini solar photovoltaic panel is mounted to enable the movement of the panel along two axes, X along azimuth (east-west, i.e. left-right) and Y along the angle of inclination of the sun's path (north-south, i.e. up- below), to allow the panel to always be at right angles to the sun's rays, to ensure greater efficiency in electricity production.

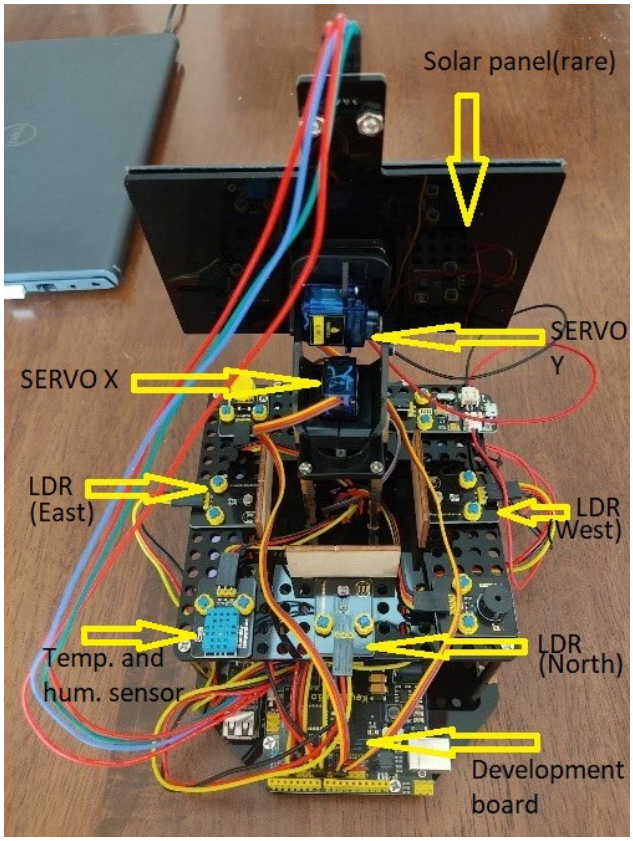


Figure 7. Prototype-back view

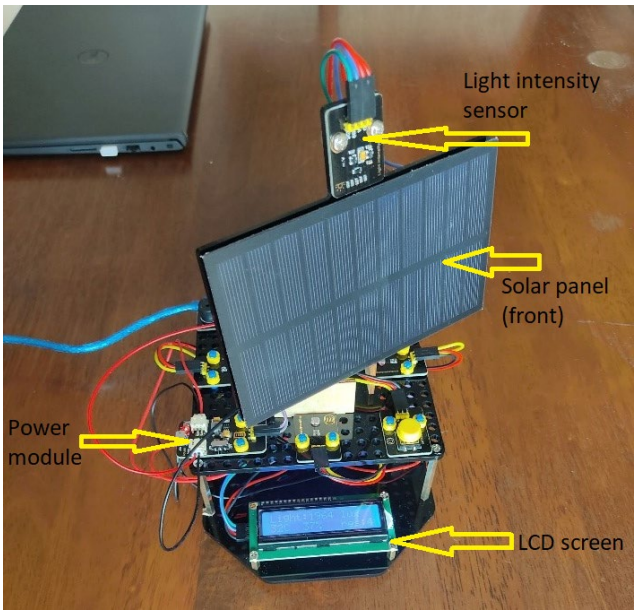


Figure 8. Prototype: front view

V. TESTING AND PRELIMINARY RESULTS

A. *Testing against functionality*

B. In order to verify the functionality of the system, the preliminary testing has been done in laboratory and real environments. In laboratory, a lamp test was successfully performed to determine whether the panel rotates towards the light source, Figure 9. A short video showing the panel tracing the lamp illumination,

https://youtube.com/shorts/wTpoHE-4QKo

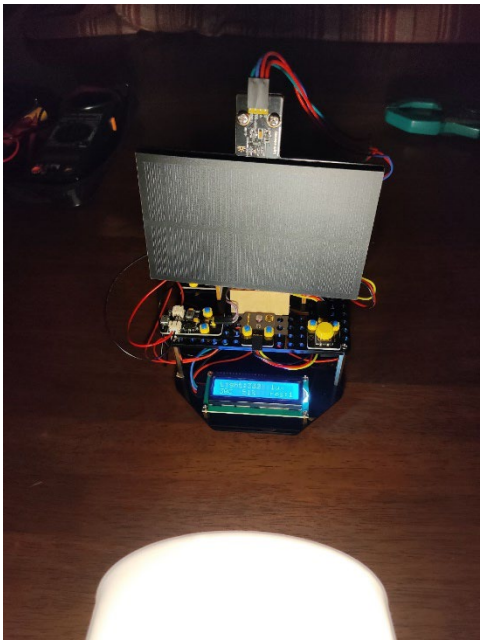


Figure 9. Testing with artificial light

In order to check the practical functionality of the system, testing was successfully carried out in a natural environment, in order to determine whether the panel turns towards the sun, Figure 10. The video shows the panel automatically pointing towards an artificial light source,
https://www.youtube.com/shorts/Qpte4JErmA8

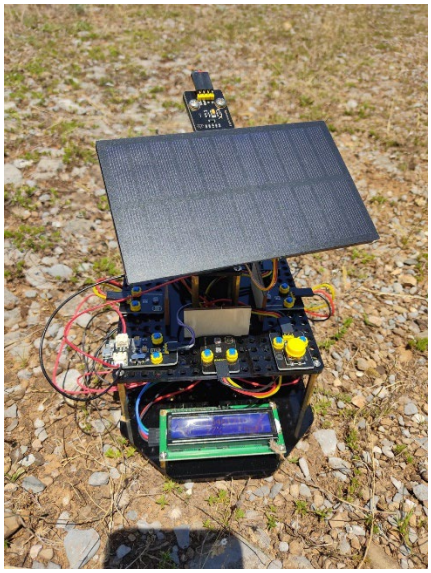


Figure 10. Testing with sunlight

C. *Testing to increased energy generation*

The measurement of the charge current (energy) produced by the solar panel system has been performed in a natural environment near Podgorica, on Ćemovsko Polje, on 07/08/2023. The generated current is measured with digital clamp meter, "ProsKit, MT-3109". The measurement was performed in the period from 06:00 to 20:15, in intervals of 15 minutes. The produced energy is calculated by using the equation: E= I*U*0.25h, as a product of the measured charging current produced by panel I, the voltage U provided by the panel (3.7V) and the measurement time interval (0.25h=15min). The measurement results are given in the graphic, Figure 11.

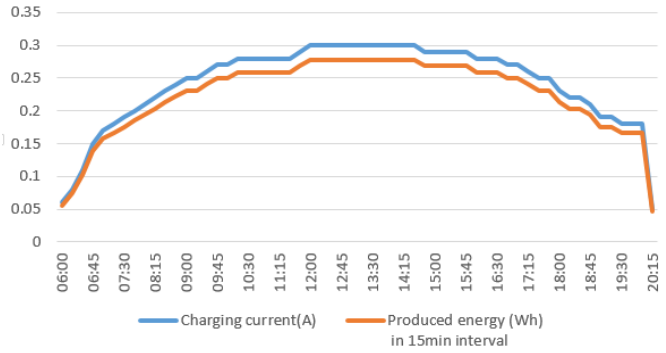


Figure 11 – measurement results with active tracking

Result of the measurement with inactive tracking function are given in Figure 11.

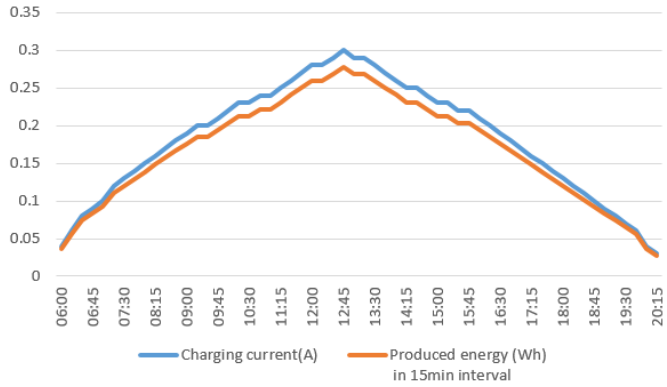


Figure 12 – measurement results with active tracking

The graphic in Figure 13 shows a comparison of the produced electricity with and without tracking.

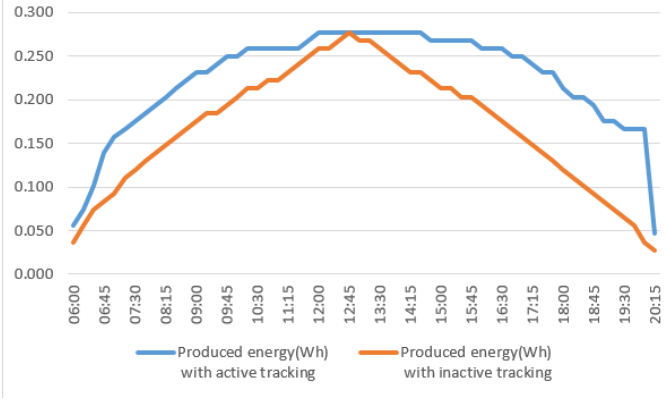


Figure 17 - comparison of the produced electricity with and without tracking

The total energy produced in the case when the tracking is active is 32.9% higher than the energy produced when the regulation was not active, which is a very significant difference that favors the application of such systems in practice.

VI. CONCLUSIONS

In this paper, a system for solar tracking is presented. The key elements of hardware and software are considered. The system is low cost and easy to implement by energy vendors and integrators. The system is able to improve the performance of photovoltaic modules for about 30% that can be of interest and benefit in practical applications.

ACKNOWLEDGMENT

Some of the results published in this paper are funded through the project "Self-sustained customized cyberphysical system experiments for capacity building among European stakeholders (SMART4ALL)", H2020, Grant Agreement No. 872614. We are thankful for support.

REFERENCES

[1] Ember: 2023 Global Electricity Review, available at: https://ember-climate.org/app/uploads/2023/04/Global-Electricity-Review-2023.pdf

[2] Muhammad E. H. Chowdhury, Amith Khandakar, Belayat Hossain and Rayaan Abouhasera "A Low-Cost Closed-Loop Solar Tracking System Based on the Sun Position Algorithm" Hindawi Journal of Sensors Volume 2019, Article ID 3681031, https://doi.org/10.1155/2019/3681031

[3] Mustafa Hamid Al-Jumaili1, Hussein M. Haglan, Mohammed Kareem Mohammed, Qusay Hatem Eesee "An automatic multi-axis solar tracking system in Ramadi city: design and implementation" Indonesian Journal of Electrical Engineering and Computer Science Vol. 19, No. 3, September 2020, pp. 1126~1234 ISSN: 2502-4752, DOI: 10.11591/ijeecs.v19.i3.pp1126-1234 https://www.researchgate.net/publication/341344753_An_Automatic_MultiAxis_Solar_Tracking_System_in_Ramadi_City_Design_and_Implementation

[4] Abdulmajid Murad, „Design and Implementation of Solar Tracking System for Photovoltaic Cells", Norwegian University of Science and Technology https://www.researchgate.net/publication/322333171_Design_and_Implementation_of_Solar_Tracking_System_for_Photovoltaic_Cells

[5] Aurélio Gouvêa Melo, Delly Oliveira Filho, Maury Martins de Oliveira Júnior, Sérgio Zolnier, Aristides Ribeiro, "Development of a closed and open loop solar tracker technology", Acta Scientiarum. Technology, vol. 39, no. 2, pp. 177-183, 2017, Universidade Estadual de Maringá https://www.redalyc.org/journal/3032/303250905007/html/

[6] Development set: https://ks0530-keyestudio-solar-tracking-kit.readthedocs.io/en/latest/KS0530.html

[7] Danko Milić, Nedeljko Lekić, Nebojša Šolaja, „FPGA bazirani PWM kontroler"(eng. FPGA based PWM Controller), Proc. 50th ETRAN Conference, Belgrade, June 6-8, 2006, Vol. I. https://www.etran.rs/common/archive/ETRAN_1955-2006/ET(R)AN_1955-2006/eTRAN/50.ETRAN.2006.1/EL/Milic%20Lekic%20Solaja.ETRAN2006.pdf

[8] Radovan Stojanovic, Tina Golubovic, Ivan Vojinovic and Jovan Djurkovic, DC Motor Control Using FPGA Generated PWM, Technical report, 2020, DOI: 10.13140/RG.2.2.25672.47364, https://www.researchgate.net/publication/339274946_DC_Motor_Control_Using_FPGA_Generated_PWM

An Experimental Platform for Fall Detection Using Beacon, Node MCU and MATLAB

Jovan Djurković[1], Radovan Stojanović[2] and Betim Cico[3]

[1]MECOnet LTD, Montenegro, [1]University of Montenegro, Faculty of Electrical Engineering, Montenegro, [3]Epoka University, Albania

[1]jovan@meconet.me

Abstract— **Healthcare wearables have become very powerful and useful devices capable to detect and monitor numerous health parameters and physical conditions. Fall Detection Systems (FDS) are a part of them, with function to detect the falls, mainly in adults. Real-time falls detection may reduce the risk of major problems and enable protective and medical personnel to act immediately. FDS varies in terms of embedded hardware and software (algorithms). Here, we present a system that can be used as an experimental platform to explore the fall detection algorithms based on inertial methods. Battery powered Bluetooth Low Energy (BLE) Beacon with built in accelerometer is used as a sensor device and data transmitter. BLE ESP based gateway receive the data and forward them to MATLAB host, on which experiments are conducted to find the most suitable algorithms. Later, the algorithms are embedded into suitable platforms, which are incorporated into sensor housings, receiver-indicators or home/hospital care systems. The architecture of overall experimentations system as well as preliminary testing results for accelerometer-based algorithms are presented, but without changing the hardware configuration, other algorithms can also be tested.**

Keywords-wearables; fall detection; accelerometer; Beacon; MATLAB; ESP Node

I. INTRODUCTION

Healthcare wearables have become very powerful devices that can detect and monitor numerous health parameters and physical conditions [1]. Although prevention efforts are underway, falls are still a major problem among the elderly population. They often lead to serious injuries, hospitalization, and even death. Not to mention the costs and efforts of the personnel who have to monitor such patients continuously. If a fall happens, then the protective or medical personnel have to react as soon as possible. Therefore, the need to develop a fall detection and alarm systems naturally arose.

A variety of different methods were developed over the last decade to automatically detect falls [2]. These have been based on video-cameras, acoustic sensors, inertial sensors, microwave radars and others. The detection device can be stationary, as cameras and scanners or mobile, most often in the form of a wearable devices, dressed on a body, as watches, bracelets, case on pillows or miniature pocket devices. Nowadays, the wearable sensors became superior because of cost, size, weight, battery powering, ease of use and, most importantly, portability [3]. With high integration and communication bum, the altering and data collection is no longer issue that provided ubiquitous health monitoring, including the remote aspects.

One of the pioneering and to date simplest fall detection systems are based on the inertial technique, where different approaches have been explored by using accelerometer or inertial measurement sensors (accelerometers and gyroscopes). The analysis of accelerometer and/or gyroscope outputs allows for detecting specific events, such as voluntary (e.g., walking, sitting, laying) or involuntary (e.g., fall) activities of daily living, based on statistical, threshold-based or deep learning algorithms. Fussed inertial method is today possible by combining a 3-axis gyroscope and a 3-axis accelerometer as separate chips, or even in the same silicon die [4]. Sometimes, the designers misunderstand the differences between an accelerometer and a gyroscope. Accelerometers measure linear acceleration (specified in mV/g) along one or several axes. A gyroscope measures angular velocity (specified in mV/deg/s). If someone takes accelerometer and imposes a rotation its outputs will not respond to change, as well as vice versa in case of gyroscope in case of linear motion.

In most cases, the fall can be expressed as a combination of linear motions, so the minimum required component for fall detection is an accelerometer. As stated in [5] the algorithms provide a high accuracy. However, one should be reserved when it comes to the accuracy of the fall, because there are completely conflicting conclusions, especially for real-world data, where the accuracy is significantly lower. The occurrence of falls is quite unpredictable.

To explore algorithms, especially for education purposes, the designers and researchers need feasible low-cost platform that meets the following requirements: wearable battery powered sensor, easily attached to body of patient, wireless low-power connection between host and sensor, the host machine that accepts the data and runs powerful mathematical and signal processing software like MATLAB and more. Later, the algorithms are embedded into suitable platforms, which are incorporated into sensor housings (wearables), home/hospital

receiver-indicators, home/hospital care systems or represent an integral part of other systems, as it is [6]. This paper presents such system and methodology, which is elaborated through proposed accelerometric algorithms.

II. METHOD

The system consists of hardware and software, Fig. 1. The sensor is in fact battery-powered Beacon, a small radio transmitter that transmits data according to adequate protocol to a compatible reader. It is possible to use any BLE Beacon. In our experiments it is DearBeacon E9, manufactured by MINEW, which is equipped with an accelerometer. The receiver is a MCU node that can work in standalone, gateway or IoT node; as example ESP32 or similar

The principal architecture of the system is shown in Fig. 1.

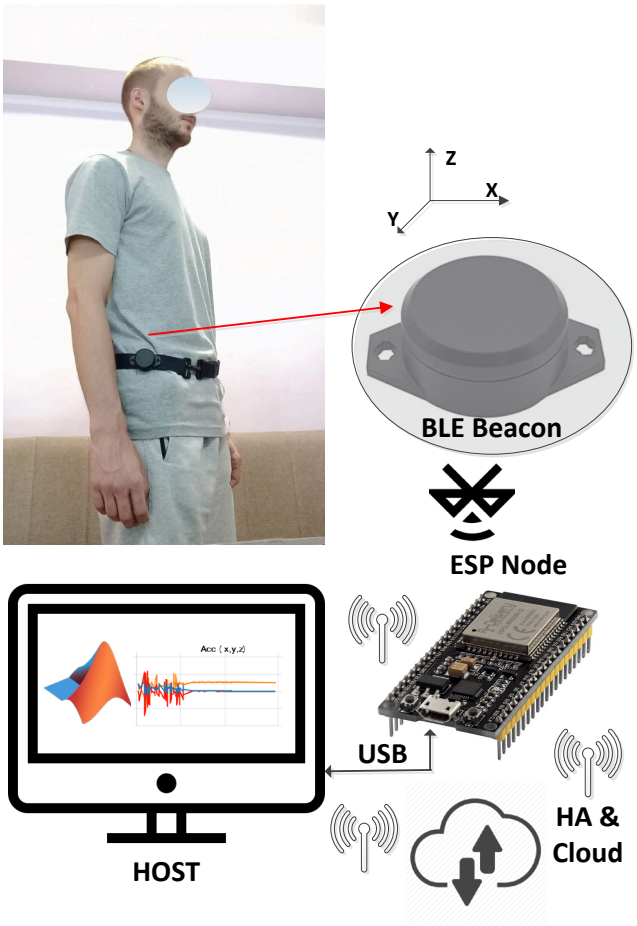


Figure 1. The principal architecture of FDS, based on Beacon, ESP MCU node and Matlab

A. Fall detection from accelerometric data

In proposed algorithm the fall is detected from accelerometric by using the decision rules over two parameters: the sum of the slope and mean of the max angle. Accelerometer data is firstly acquired in sets of 50 by 10Hz sampling frequency (5 seconds duration) and then processed.

$$A = \sqrt{A_x^2 + A_y^2 + A_z^2}, \quad (1)$$

where A_x, A_y, A_z are accelerometer data of three amplitudes in a range (±1 g).

The sum of the slopes in the window W of 50 samples size is calculated as (2):

$$slope_{sum} = \sum_{i=1}^{Wi}(A_{i+1} - A_i); if\, A_{i+1} > A_i \quad (2)$$

and $slope_{max}$ (3):

$$slope_{max} = \max\left(slope_{sum}(W1) \ldots slope_{sum}(Wi)\right) \quad (3)$$

The decision is taken as:

1. $slope_{max} < 0.1$ – for steady position (no movement)
2. $0.1 \leq slope_{max} \leq 0.6$ – for moving, doing everyday activities
3. $slope_{max} > 0.6$ – for the impacts, indicated a fall, jump or running

Only when the third condition is met, there is a chance that a fall has happened and this condition is later reinforced by analyzing the changes of angle.

Angle changes relative to upright position are observed as

$$\phi_{xr} = \phi_x - \phi_{x0};\ \phi_{yr} = \phi_y - \phi_{y0};\ \phi_{zr} = \phi_z - \phi_{z0}, \quad (4)$$

where ϕ_{x0}, ϕ_{y0}, ϕ_{z0} are tilt angles in upright position, measured experimentally ϕ_{x0} = -64.2°; ϕ_{y0} = 25.74°; ϕ_{z0} = 0°, while individual tilt angles are calculated as below:

$$\phi_x = \tan^{-1}\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right) \quad (5)$$

$$\phi_y = \tan^{-1}\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right) \quad (6)$$

$$\phi_z = \tan^{-1}\left(\frac{A_z}{\sqrt{A_x^2 + A_y^2}}\right) \quad (7)$$

When fall happens at least one of these angles will rise to about 90°, so for simplicity only maximal values of these three angles is observed: $\phi_{max} = \max(\phi_x, \phi_y, \phi_z)$. Next, the average value is calculated around peak position to exclude too short bursts which can't indicate a fall because a person needs some time to get up. A few seconds period of observation is enough for such short frame of 5 seconds refreshment rate, $\phi_{avg} = \text{mean}(\phi_{max}[max_{in}, max_{in}+30])$, max_{in} – index of an angle peak and 30 additional samples which correspond to 3 seconds

The peak can happen to be near the end of 50 samples frame so it is less than 30 samples from peak to the end. To make sure that average value is calculated on sufficient number of samples, two consecutive frames are joined to make 100 samples. Thus, when a peak is positioned nearer the end of new frame the average value is calculated upon gathering new samples, which consequently makes additional delay of 5 seconds.

From the angle side of view, an average angle change that meets right conditions to indicate a fall is $\phi_{avg} > 75$. When the

both axis and angle conditions are met, we can conclude that there was a fall:

$$\begin{cases} slope_{max} > 0.6 \\ \phi \text{avg} > 75 \end{cases} \quad (8)$$

III. EXPERIMENT AND RESULTS

We have tested the proposed methodology in real conditions. For this purpose, MATLAB GUI has been designed that communicates with ESP Node by virtual RS232 over USB. ESP Node scans the presence of nearby Beacons that is indicated by GUI, P1 presents Beacon (patient) 1 etc. and forwards data to MATLAB code. Figure 1 to Figure 7 present different states of the observed person's activity. Red graph background indicates that the fall condition is met (max slope and average phi graphs). It is illustrated how "Jumping", especially several consecutive jumps can be understood as a fall from slope aspect of view, that is logical.

A case of a real fall is shown in Figure 7 where both slope and phi variables surpassed the conditional values, and alarm for patient 1 is turned on.

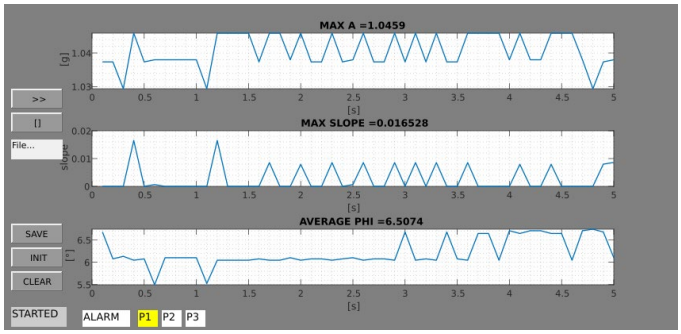


Figure 2. The state of "CALM"

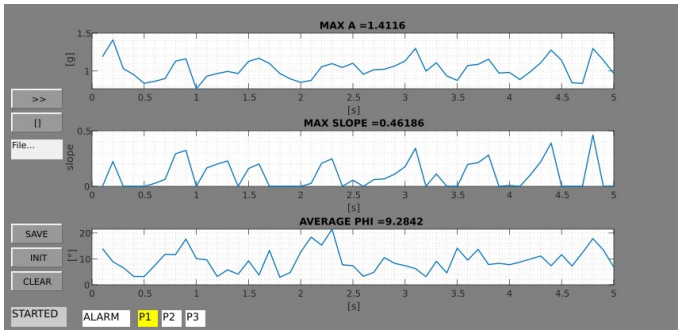


Figure 3. The state of "MOVING"

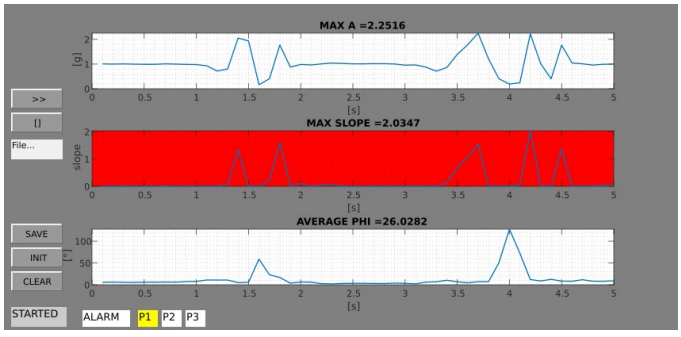


Figure 4. The state of "JUMPING". Several consecutive jumps

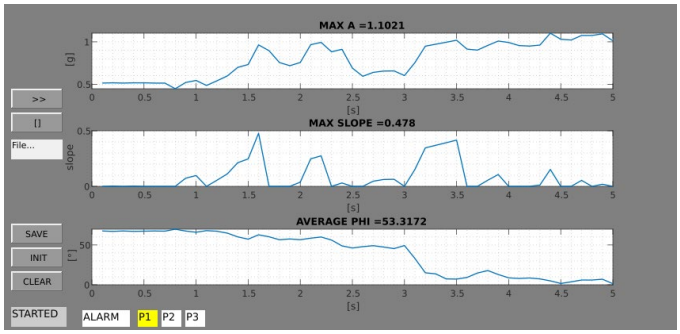


Figure 5. The state of "LAYING DOWN"

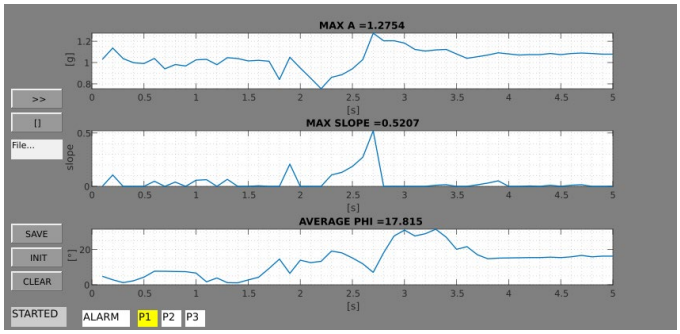


Figure 6. The state of "SITING". Several consecutive jumps

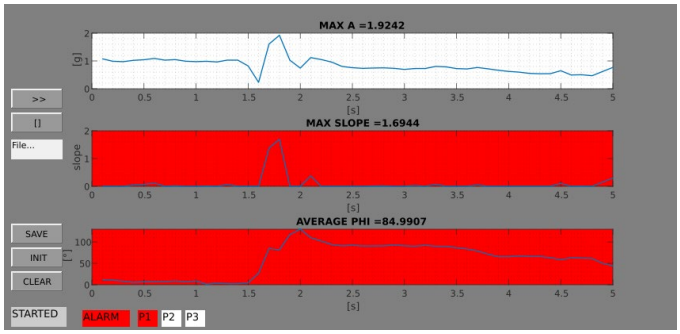


Figure 7. The state of "FALLING DOWN"

TABLE I. FALL DETECTION ACCURACY FOR DIFFERENT STATES (ACTIVITIES)

State/ Action	Correct detection rate (fall or not fall) in %			
	max slope > 0.6 avgphi > 75	max slope > 0.5 avgphi > 75	max slope > 0.6 avgphi > 70	max slope > 0.5 avgphi > 70
Fall	64.14	66.76	76.67	79.3
Moving	100	100	100	100
Laying down/standing up	98.04	94.12	96.48	91.77
Standing or sitting steadily	100	100	100	100
Sitting down/standing up	100	100	100	100
Jumping	99.17	99.17	99.17	99.17

Statistical testing has been performed over the sets of the data that contains different scenario from different persons, Table I. As seen, the fall accuracy is about 64% for real-world data. Other cases produce very low false positives.

CONCLUSIONS

This paper presents the experimental methodology for developing and testing algorithms for fall detection. It is low cost and feasible for any laboratory or designer group (company). We showed an example of employing the methodology on proposed accelerometer-based algorithms. The experiments showed that fall detection is still not easy task in case of using only one type of data (accelerometric). Obtained detection accuracy on real-world data is about 67%. Detection of other states has very good accuracy. The experiments on other algorithms, such as the neural networks, etc. are ongoing.

ACKNOWLEDGMENT

Some of the results published in this paper are funded through the PAE Experiment of Project "Self-sustained customized cyberphysical system experiments for capacity building among European stakeholders (SMART4ALL)", H2020, Grant Agreement No. 872614. We are thankful for support.

REFERENCES

[1] Stojanović, R., Djurković, J., Mijušković, S., Lutovac, B., & Škraba, A. (2023, June). SYNTROFOS: A Wearable Device for Vital Sign Monitoring, Hardware and Signal Processing Aspects. In 2023 12th Mediterranean Conference on Embedded Computing (MECO) (pp. 1-6). IEEE.

[2] Wang, X., Ellul, J., & Azzopardi, G. (2020). Elderly fall detection systems: A literature survey. Frontiers in Robotics and AI, 7, 71.

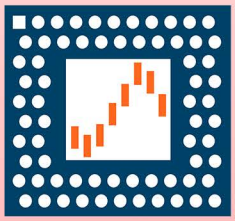
[3] Kubov, V. I., Dymytrov, Y. Y., Stojanović, R., Kubova, R. M., & Škraba, A. (2020, June). A Feasible IoT System for Monitoring PPG and ECG Signals by using Low-cost Systems-on-chips and HTML Interface. In 2020 9th Mediterranean Conference on Embedded Computing (MECO) (pp. 1-4). IEEE.

[4] https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/

[5] Bagala, Fabio, et al. "Evaluation of accelerometer-based fall detection algorithms on real-world falls." PloS one 7.5 (2012): e37062.

[6] R. Stojanović, A. Škraba and B. Lutovac, "A Headset Like Wearable Device to Track COVID-19 Symptoms," 2020 9th Mediterranean Conference on Embedded Computing (MECO), 2020, pp. 1-4, doi: 10.1109/MECO49872.2020.9134211.

Supported by:



ISSN 2980-7298