IKON - IT Kompetenz Netzwerk zur Stärkung der Grenzregion Sachsen-Tschechien 2021

Prof. Dr. Dr. h. c. Wolfram Hardt (Hrsg.)

IBS Scientific Workshop Proceedings
Herausgegeben von Stiftung IBS, Wolfram Hardt
Band 13

TUDpress

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

ISBN 978-3-95908-237-2

THELEM Universitätsverlag
GmbH und Co. KG
http://www.tudpress.de

Gesetzt von den Herausgebern.
Printed in Germany.

IKON - IT Kompetenz Netzwerk zur Stärkung der Grenzregion Sachsen-Tschechien 2021

Prof. Dr. Dr. h. c. Wolfram Hardt (Hrsg.)

TUD*press*
2021

Preamble

Am 29. September 2020 startete die zweite Bildungsphase des internationalen und EFRE-geförderten IKON-Projektes unter der Leitung von Prof. Dr. Dr. h. c. Wolfram Hardt der Technischen Universität Chemnitz. Bei dem Projekt der handelt es sich um das „IT Kompetenz Netzwerk zur Stärkung der Grenzregion Sachsen –Tschechien“ kurz IKON, welches als Kooperation zwischen der TU Chemnitz und der Jan Evangelista Purkyně-Universität (UJEP) seit dem November 2019 erfolgreich verwirklicht wird.
Inhaltlich knüpfte die zweite Bildungsphase an die Themen der ersten Bildungsphase (November 2019 bis Juli 2020) an und baute indessen auf der bereits geschaffenen Wissensgrundlage der ersten Durchführung auf, sodass deren thematischer Hintergrund in der zweiten Bildungsphase vertieft werden konnte. Dabei standen wieder die drei großen Themen: Automotive, Digitale Prozesse und Industrieautomation im Vordergrund. In den mehrtägigen Seminaren und Workshops wurden komplexe und aktuelle Aufgabenstellungen bearbeitet. Die Ergebnisse können in dieser Zusammenfassung nachgelesen werden.

Dne 29. září 2020 odstartovala druhá realizační fáze projektu podpořeného v rámci Programu přeshraniční spolupráce Česká republika – Sasko v období 2014 – 2020 IKON pod vedením Prof. Dr. Dr.h. c. Wolframa Hardta z Technické univerzity v Chemnitz. Hlavní téma projektu je „IT síť kompetencí k posílení příhraničního regionu Česko – Sasko“, zkráceně IKON, které je v rámci spolupráce mezi Technickou univerzitou Chemnitz (TUC) a Univerzitou Jana Evangelisty Purkyně v Ústí nad Labem (UJEP) úspěšně realizováno od listopadu 2019.
Obsahově navazovala druhá realizační fáze na témata z první realizační fáze (listopad 2019 až červenec 2020), přičemž vycházela z již získaných výsledků první realizační fáze a mohla tak zpracovávána témata v rámci druhé fáze dále prohloubena a rozvíjena. Jednalo se opět o tři základní témata: Automotivní řízení, Digitální procesy a Průmyslová automatizace, která byla zpracovávána na základě aktuálních zadání studenty a vědeckými pracovníky v několikadenních seminářích a workshopech. Dosažené výsledky jsou obsaženy v tomto sborníku.

Inhaltsverzeichnis

Modeling of Complex System Fault-Tolerance

Abstract

System fault-tolerance can be achieved by performing such procedures as error detection and system recovery. Different methods can be applied to perform these procedures. There exists class of complex systems for which these procedures can be executed by using self-checking and self-diagnosis for error detection and gradual degradation for system recovery. Efficiency of these methods can be evaluated by performing modeling and simulation. Students have conducted research on the subject of "Modeling of complex system fault-tolerance". Particularly, they have determined the necessary changes that should be made in diagnosis algorithm (table algorithm) after detection of error, and how these changes influence the consequent system recovery (by using Petri Nets). Three variants of system recovery were modeled and assessed. The obtained results allow making the appropriate recommendations concerning system recovery and, consequently, how to provide system fault-tolerance.

Motivation

Student background

There are six students registered to complete the tasks of Block 6. They are all Bachelor students of University J.E. Purkyne (UJEP). They are all from faculty of Science. The gender ratio can be considered as balanced with 4 men and 2 women. All of the students are enrolled in the last semester of study (Table 1).

Student	Gender	Faculty	Field of study	Current semester
1	male	FS	IS	6
2	male	FS	IS	6
3	male	FS	IS	6
4	male	FS	IS	6
5	female	FS	IS	6
6	female	FS	IS	6

Six students registered for this block have chosen the following topics.

- Developing diagnosis algorithm. Making changes in the algorithm after detection of faulty unit(s). Determining the impact of the performed changes on the following diagnosis and on the execution of the system main tasks.
- Modeling and simulation of tests execution. Modeling of recovery procedure. Assessment of different methods of system reconfiguration after removing the faulty unit(s).

Introduction

The researches of our students are based on the results obtained in Block 3 and use these results. The main object of the students' research is the complex systems that satisfy the certain requirements, such as:

- system can be checked, diagnosed and recovered without external facilities;
- ability to work autonomously for a long time;
- ability to work in harmful environment which can cause multiple faults.

Examples of such systems are sensor networks, many-core processors, multi-agent systems, smart matter systems etc. [1,2,3,4].

The main problems and tasks that should be solved are as follows:

1) modeling the procedures that provide system fault-tolerance (i.e., checking, diagnosis and recovery);

2) determining the impact of system recovery on the consequent checking and diagnosis;
3) assessment of the achieved fault-tolerance;
4) recommendations for improvement of system fault-tolerance.

System fault-tolerance can be modeled by using different mathematic models. Mostly, for this purpose the system state transition diagram, particularly Markov model is used [5]. Then, Markov model is analyzed in order to determine the following characteristics:

- the probability of system being in a given state at a given point in time;
- the amount of time a system is expected to spend in a given state;
- as well as, the expected number of transitions between system states

By using these data it is possible to compute and estimate the system reliability and system fault-tolerance.

There are two main methods of how to provide system fault-tolerance:

- by using standby system units;
- by allowing system degradation (e.g., by reducing the system functionality or by reducing the quality of the performed system tasks).

Our students used the second method, i.e., system degradation. Simple example of this case is shown in Fig. 1.

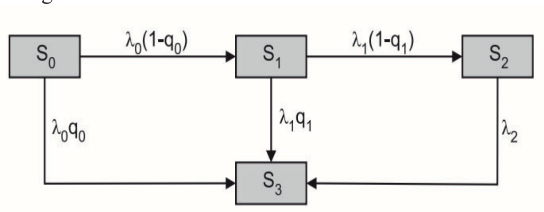


Figure 1. Initial model of system states.

In the figure, the following denotations are used:

S_0 – all of the system units are fault-free. System is fully operational.

S_1 – one of the system units is isolated. The system is slightly degraded but still continues to deliver degraded (but still acceptable) services.

S_2 – two system units are isolated. System is more degraded but still is able to deliver the acceptable services.

S_3 – system failure.

In the figure, it is shown the simple case when system can tolerate only two failed units. By λ_0, λ_1, λ_2 are denoted the rates of system transitions from one state to another. By q_0, q_1 are denoted the probabilities of the corresponding transitions. Values λ_i , i=0,1,2, depend on the reliability of system units, and values q_0 and q_1 depend on the efficiency of checking, diagnosis and recovery procedures. For example, for homogeneous system with five units which have $\lambda=10^{-4}$ 1/h, probability of system failure P_3 has the following dependence on time (see Fig. 2).

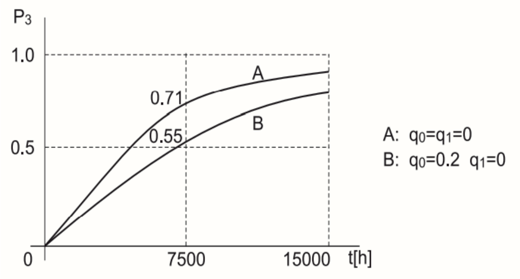


Figure 2. Probability of system failure, $P_3(t)$.

The case of q_0=q_1=0 corresponds to absolutely perfect checking, diagnosis and recovery. Thus, the impact of checking, diagnosis and recovery on the system failure is essential. In Fig. 3, the refine model is shown.

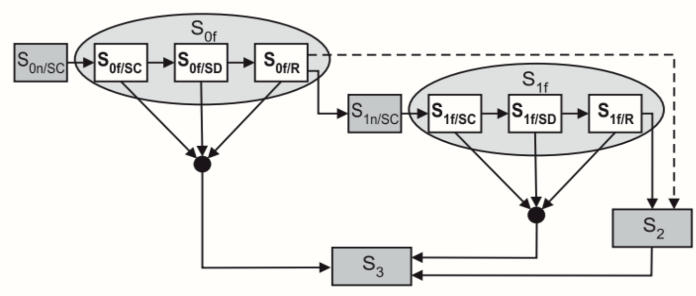


Figure 3. Refined model.

This model takes into account the following three system states:
- Checking: indicates the detection of failed unit;
- Diagnosis: indicates the localization of failed unit;
- Recovery: indicates recovery from faulty situation (error state).

In practice, for providing system recovery the following strategies can be used:
- Reduce the number of system tasks
- Use the simplified or shortcut algorithms (i.e., delivering less precise outcomes or less credible results)
- Decrease in system performance (i.e., system operations will slow down).

The choice of appropriate strategy is application-specific. The chosen strategy of system recovery has direct impact on the following checking and diagnosis. For example, it is expected that after system recovery the intensity of tests will decrease since the number of active units will be lesser as compared to the number of active units that were in the system before recovery. Moreover, it should be taken into account the fact that after recovery the system units will be more loaded.

Students in their researches paid much attention to the problem of how recovery can impact the consequent checking and diagnosis procedures [6]. For modeling and simulation checking, diagnosis and recovery procedures the following facilities can be used:
- Petri nets (tools that use PN [7,8,9], e.g., SHARPE [10]);
- mathematic modeling [11];

- computer simulation;
- real simulation.

Individual tests are executed randomly. Thus, the whole checking procedure is random. In the given case, the most appropriate model for this procedure is Generalized Stochastic Petri Nets. Petri Nets can be used for modeling checking procedure, particularly for modeling mutual testing. In Fig. 4, the simplified example of Petri Net that models only one test between two system units is shown.

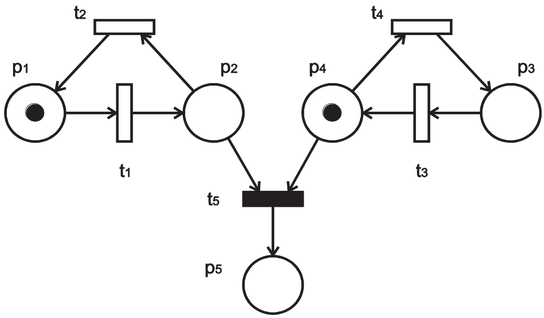


Figure 4. PN that models a test in the system.

In Fig. 4, places p_1, p_2, p_3, p_4 model the engaged and idle (i.e., free) states of system units. Whereas, place p_5 models the event that test has been executed.
As concerns modeling of diagnosis procedure, students worked on the task of how to evaluate and reduce the complexity of diagnosis and, consequently, the possible duration of diagnosis procedure.

For diagnosis, the following strategies can be chosen [5]:
- unique diagnosis;
- sequential diagnosis;
- excess diagnosis;
- probabilistic diagnosis.

Our students have chosen the unique diagnosis and perform modeling of diagnosis procedure that uses this strategy.

When considering fault-tolerance it is very important to define the concept of system failure. System failure may occur due to the following reasons:
- exhaustion of system redundancy;
- unacceptable system degradation;
- lack of time for system recovery (see Fig. 5).

Otherwise, we can say that system has tolerated the failure(s) of its units.

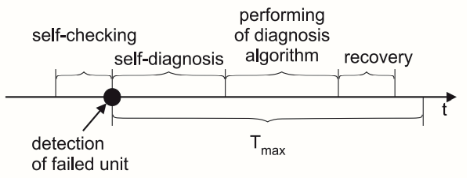


Figure 5. Duration of procedures which provide system fault-tolerance.

Research Questions and Methods

1. The first research question is the question of what changes in diagnosis algorithm should be made after detection of failed unit and how these changes influence the consequent system recovery.

Our students have developed the diagnosis algorithm based on the table of potential syndromes. The table of potential syndromes is formed before the testing procedure begins. The table contains all possible syndromes that can be obtained for any allowable faulty situation. Number of allowable faulty situations, Q, is restricted and depends on the value of t. Value t is the diagnosability measure which reflects the capability and quality of system diagnosis (given particular testing assignment). N is number of system units.

$$Q = \sum_{i-1}^{t} \binom{n}{i}$$

Diagnosis consists in comparing of syndrome obtained after tests execution with the syndromes of the table. A table of potential syndromes has the following columns:

- denotations of the faulty situations (S_i);
- sets of faulty units (e.g., $\{u_1, u_2\}$);
- denotations of the potential syndromes R_p^i. Potential syndrome R_p^i corresponds to the faulty situation S_i , i-1,2,..,Q.

Elements r_{ij} of the syndrome R_p^i are the elements of the table. The table of potential syndromes contains as many rows as there are faulty situations. The values r_{ij} are either 0 or 1 or X (where X can take values 0 or 1 according to the model of interpretation of test results proposed by Preparata [ ]). Syndrome obtained after tests execution, as distinct from the potential syndromes, contains only values of 0 or 1. Example of table of potential syndromes for the system with five units is shown in Table 2.

Having performed all tests and obtained actual syndrome, it is possible to perform a diagnosis algorithm. This algorithm consists in comparing actual syndrome with the potential syndromes of the table. The goal is to find in the table the syndrome which coincides with the actual syndrome. Comparing procedure can be executed according to different strategies. Basic and straightforward strategy consists in successive comparing the elements of actual syndrome with the elements of separate potential syndromes which are in the separate rows of the table. Comparing starts from the first row and goes on until the coincidence is found. Another strategy of comparing consists in the following. At the first step, only rows that contain the first element that coincides with the first element of actual syndrome are chosen. For the example of the system with five units, in which units u_3 and u_5 are faulty, these rows are 1,3,4,5,6,7,8,9,13,14,15. It is assumed that obtained actual syndrome is R_a={0,1,1,0,1,0,1,0,0,0}.

Table 2. Table of potential syndromes for the system with 5 units.

S_i	faulty units	potential syndrome	test results									
			r_{12}	r_{13}	r_{23}	r_{24}	r_{34}	r_{35}	r_{45}	r_{41}	r_{51}	r_{52}
S_1	u_1	R_p^1	X	X	0	0	0	0	0	1	1	0
S_2	u_2	R_p^2	1	0	X	X	0	0	0	0	0	1
S_3	u_3	R_p^3	0	1	1	0	X	X	0	0	0	0
S_4	u_4	R_p^4	0	0	0	1	1	0	X	X	0	0
S_5	u_5	R_p^5	0	0	0	0	0	1	1	0	X	X
S_6	u_1,u_2	R_p^6	X	X	X	X	0	0	0	1	1	1
S_7	u_1,u_3	R_p^7	X	X	1	0	X	X	0	1	1	0
S_8	u_1,u_4	R_p^8	X	X	0	1	1	0	X	X	1	0
S_9	u_1,u_5	R_p^9	X	X	0	0	0	1	1	1	X	X
S_{10}	u_2,u_3	R_p^{10}	1	1	X	X	X	X	0	0	1	1
S_{11}	u_2,u_4	R_p^{11}	1	0	X	X	1	0	X	X	0	1
S_{12}	u_2,u_5	R_p^{12}	1	0	X	X	0	1	1	0	X	X
S_{13}	u_3,u_4	R_p^{13}	0	1	1	1	X	X	X	X	0	0
S_{14}	u_3,u_5	R_p^{14}	0	1	1	0	X	X	1	0	X	X
S_{15}	u_4,u_5	R_p^{15}	0	0	0	1	1	1	X	X	X	X

At the second step, the second element of the rows chosen at the first step (i.e., rows 1,3,4,5,6,7,8,9,13,14,15) is compared with the second element of actual syndrome. Only rows that contain the second element which coincides with the second element of actual syndrome remain for further consideration. In the given case, these rows are 1,3,6,7,8,9,13,14. This procedure continues for the next elements and ends at the seventh element of actual syndrome when only one potential syndrome remains, particularly R_p^{14}. The first strategy of comparing requires 140 simple element comparisons, whereas the second strategy requires 105 such element comparisons only.

For assessment of diagnosis algorithm, the simulation of system state was conducted. The first step of such simulation was indicating (assignment) the faulty units and consequent generation of syndrome for this system state. There are two ways of syndrome generation. In the first case, syndrome is generated without account of units' reliability. The second case implies accounting the probabilities of faulty situations. In their research, the students considered column comparison strategy and syndrome generation which takes into account units failure probabilities (see two slides from student's presentation in Fig. 6 and Fig. 7).

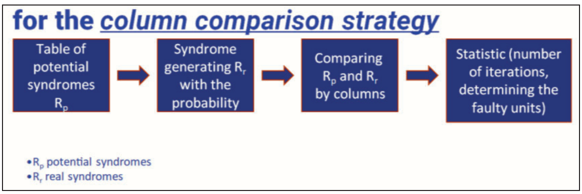


Figure 6. Slide from student's presentation.

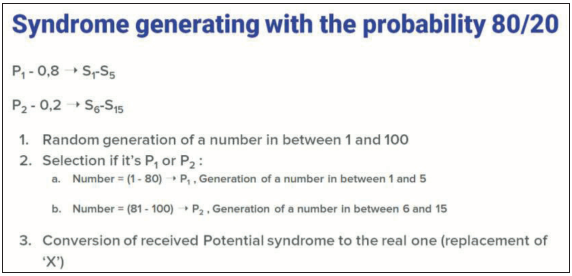


Figure 7. Slide from student's presentation.

After performing diagnosis algorithm and identification all faulty units, the initial diagnosis algorithm should account the fact that parameters of the system are changed (e.g., number of system units). After removing the faulty unit(s), the table of potential syndromes is reduced (see example in Fig. 8 in which unit u_2 is removed from the system).

S_i	faulty units	potential syndrome	test results									
			r_{12}	r_{13}	r_{23}	r_{24}	r_{34}	r_{35}	r_{45}	r_{41}	r_{51}	r_{52}
S_1	u_1	R_p^1	X	X	0	0	0	0	0	1	1	0
~~S_2~~	~~u_2~~	~~R_p^2~~	~~1~~	~~0~~	~~X~~	~~X~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~
S_3	u_3	R_p^3	0	1	1	0	X	X	0	0	0	0
S_4	u_4	R_p^4	0	0	0	1	1	0	X	X	0	0
S_5	u_5	R_p^5	0	0	0	0	0	1	1	0	X	X
~~S_6~~	~~u_1,u_2~~	~~R_p^6~~	~~X~~	~~X~~	~~X~~	~~X~~	~~0~~	~~0~~	~~0~~	~~1~~	~~1~~	~~1~~
S_7	u_1,u_3	R_p^7	X	X	1	0	X	X	0	1	1	0
S_8	u_1,u_4	R_p^8	X	X	0	1	1	0	X	X	1	0
S_9	u_1,u_5	R_p^9	X	X	0	0	0	1	1	1	X	X
~~S_{10}~~	~~u_2,u_3~~	~~R_p^{10}~~	~~1~~	~~1~~	~~X~~	~~X~~	~~X~~	~~X~~	~~0~~	~~0~~	~~1~~	~~1~~
~~S_{11}~~	~~u_2,u_4~~	~~R_p^{11}~~	~~1~~	~~0~~	~~X~~	~~X~~	~~1~~	~~0~~	~~X~~	~~X~~	~~0~~	~~1~~
~~S_{12}~~	~~u_2,u_5~~	~~R_p^{12}~~	~~1~~	~~0~~	~~X~~	~~X~~	~~0~~	~~1~~	~~1~~	~~0~~	~~X~~	~~X~~
S_{13}	u_3,u_4	R_p^{13}	0	1	1	1	X	X	X	X	0	0
S_{14}	u_3,u_5	R_p^{14}	0	1	1	0	X	X	1	0	X	X
S_{15}	u_4,u_5	R_p^{15}	0	0	0	1	1	1	X	X	X	X

Figure 8. Table of potential syndromes after removing unit u_2.

After performed changes, the diagnosis algorithm can be executed more quickly since required number of element comparison reduces. Summary of the performed simulation for the system with five units and one faulty unit (for different strategies of element comparisons) is shown in Fig. 9. Results of the performed simulation show that after removing of failed unit the diagnosis algorithm can be executed more quickly (35% on average). On the other hand, after removing of faulty unit(s) the system tasks should be redistributed among the remaining fault-free units. Such tasks redistribution may result in increasing the duration of the consequent mutual testing. As a consequence, probability of system failure may increase due to the third reason (i.e., due to the lack of time for system recovery (see Fig. 5)). In view of this, the task arises to determine such system recovery that minimizes the time of mutual testing.

Figure 9. Required number of comparisons before and after unit removing.

2. The second research question is the question of determining the variant of system recovery that provides minimal time of the consequent mutual testing. In other words, it is necessary to compare possible methods of system tasks redistribution and choose the "best" one.
For this purpose modeling of mutual testing for different options of system tasks redistribution after removing faulty unit was performed. For modeling of mutual testing the tool SHARP [ ] was used. Students did research on the system with five units. Petri net which models mutual testing in this system is shown in Fig. 10.

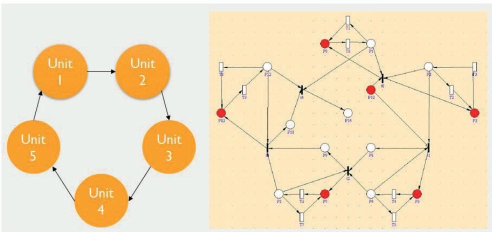


Figure 10. Petri net which nodels mutual testing for systém with five units.

After detectiing and removing a faulty unit four units remain in the systém. It is assumed that after detecting and removing a faulty unit system tasks should be redistributed among the remaining fault-free units. As a result of such tasks redistribution, parameters of Petri net which models mutual testing in the system with four units (see Fig. 11) will differ from parameters of the initial model.

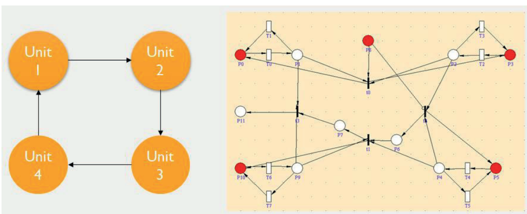


Figure 11. Petri net which models mutual testing in the system after removing faulty unit.

Students examined faulty situations in the system which include up to two faulty units. Petri net which models mutual testing in the system after removing two faulty units is shown in Fig. 12.

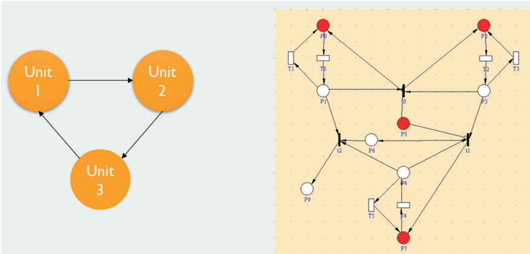


Figure 12. Petri net which models mutual testing in the system after removing two faulty units.

Three possible scenarios of system tasks redistribution after removing failed unit(s) were considered. These scenarios are appointed in student's slide (see Fig. 13).

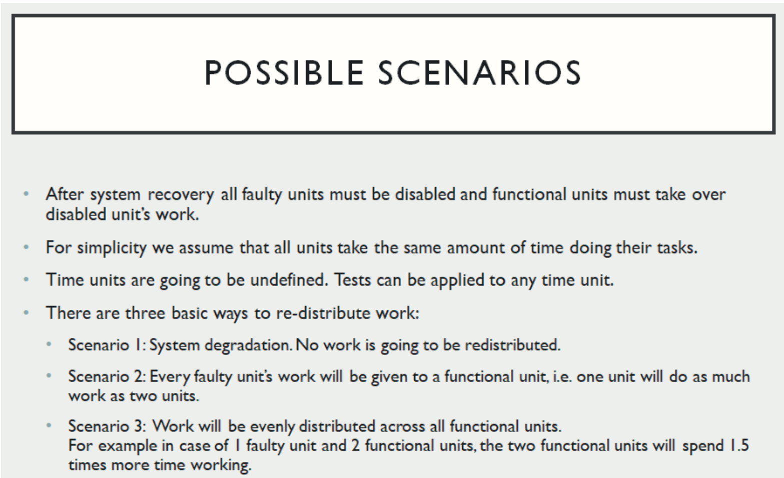


Figure 13. Possible scenarios of tasks redistribution.\

For each scenario the modeling of mutual testing was performed. Results of the performed modeling are as follows:

Scenario 1.

Parameters of model for system with 5 units:

- rate of transition from unit' engaged state to idle state is equal to 1 (conditional time unit, tu);
- rate of transition from idle to engaged state is equal to 10 tu.

Result of modeling: duration of mutual testing is equal to 60 time units.

Parameters of model after removing of failed unit (four units remain in the system):

- rate of transition from unit' engaged state to idle state is equal to 1 tu;
- rate of transition from idle to engaged state is equal to 10 tu.

Result of modeling: duration of mutual testing is equal to 80 tu.

Parameters of model after removing the next failed unit (three units remain in the system):

- rate of transition from unit' engaged state to idle state is equal to 1 tu;
- rate of transition from idle to engaged state is equal to 10 tu.

Result of modeling: duration of mutual testing is equal to 65 tu.
Parameters of model after removing the next failed unit (two units remain in the system):

- rate of transition from unit' engaged state to idle state is equal to 1 tu;
- rate of transition from idle to engaged state is equal to 10 tu.

Result of modeling: duration of mutual testing is equal to 55 tu.

Scenario 2.
Parameters of model for system with 5 units:

- rate of transition from unit' engaged state to idle state is equal to 1 (conditional time unit, tu);
- rate of transition from idle to engaged state is equal to 10 tu.

Result of modeling: duration of mutual testing is equal to 60 time units.
Parameters of model after removing of failed unit (four units remain in the system):

- rate of transition from unit' engaged state to idle state for one system unit is equal to 0.5 tu and for other units this rate is equal to 1 tu;
- rate of transition from idle to engaged state is equal to 10 tu.

Result of modeling: duration of mutual testing is equal to 100 tu.
Parameters of model after removing the next failed unit (three units remain in the system):

- rate of transition from unit' engaged state to idle state for two units is equal to 0.5 tu and for third unit is equal to 1 tu ;
- rate of transition from idle to engaged state is equal to 10 tu.

Result of modeling: duration of mutual testing is equal to 145 tu.
Parameters of model after removing the next failed unit (two units remain in the system):

- rate of transition from unit' engaged state to idle state for one unit is equal to 0.333 tu and 0.5 tu for another unit;
- rate of transition from idle to engaged state is equal to 10 tu.

Result of modeling: duration of mutual testing is equal to 225 tu.

Scenario 3.
Parameters of model for system with 5 units:

- rate of transition from unit' engaged state to idle state is equal to 1 (conditional time unit, tu);
- rate of transition from idle to engaged state is equal to 10 tu.

Result of modeling: duration of mutual testing is equal to 60 time units.
Parameters of model after removing of failed unit (four units remain in the system):

- rate of transition from unit' engaged state to idle state for all system units is equal to 0.8 tu;
- rate of transition from idle to engaged state is equal to 10 tu.

Result of modeling: duration of mutual testing is equal to 100 tu.
Parameters of model after removing the next failed unit (three units remain in the system):

- rate of transition from unit' engaged state to idle state for all units is equal to 0.6 tu;
- rate of transition from idle to engaged state is equal to 10 tu.

Result of modeling: duration of mutual testing is equal to 140 tu.
Parameters of model after removing the next failed unit (two units remain in the system):

- rate of transition from unit' engaged state to idle state for all units is equal to 0.4 tu;
- rate of transition from idle to engaged state is equal to 10 tu.

Result of modeling: duration of mutual testing is equal to 235 tu.
Summary of the performed modeling is shown in Fig. 14.

SCENARIO 1

Functional units	Working rate	Idle rate	Time spent
5	5x1	10	60
4 (1 faulty)	4x1	10	80
3 (2 faulty)	3x1	10	65
2 (3 faulty)	2x1	10	55

SCENARIO 2

Functional units	Working rate	Idle rate	Time spent
5	5x1	10	60
4 (1 faulty)	1x0.5,3x1	10	100
3 (2 faulty)	2x0.5, 1x1	10	145
2 (3 faulty)	1x0.3, 1x0.5	10	225

SCENARIO 3

Functional units	Working rate	Idle rate	Time sp
5	5x1	10	60
4 (1 faulty)	4x0.8	10	100
3 (2 faulty)	3x0.6	10	140
2 (3 faulty)	2x0.4	10	235

Figure 14. Results of the performed modeling.

Modeling of mutual testing was also performed for other system parameters. In the given case, the rate of transition from idle to engaged state was changed. For the case when this rate is equal to 5 tu, results of modeling are shown in Fig. 15.

SCENARIO 1

Functional units	Working rate	Idle rate	Time spent
5	5x1	5	40
4 (1 faulty)	4x1	5	40
3 (2 faulty)	3x1	5	40
2 (3 faulty)	2x1	5	35

SCENARIO 2

Functional units	Working rate	Idle rate	Time spent
5	5x1	5	40
4 (1 faulty)	1x0.5,3x1	5	60
3 (2 faulty)	2x0.5, 1x1	5	80
2 (3 faulty)	1x0.3, 1x0.5	5	125

SCENARIO 3

Functional units	Working rate	Idle rate	Time spent
5	5x1	5	40
4 (1 faulty)	4x0.8	5	60
3 (2 faulty)	3x0.6	5	80
2 (3 faulty)	2x0.4	5	130

Figure 15. Results of modeling (when rate of transition from idle to engaged state is equal to 5 tu).

For the case when this rate is equal to 1 tu, results of modeling are shown in Fig. 16.

SCENARIO 1

Functional units	Working rate	Idle rate	Time spent
5	5x1	1	25
4 (1 faulty)	4x1	1	25
3 (2 faulty)	3x1	1	20
2 (3 faulty)	2x1	1	15

SCENARIO 2

Functional units	Working rate	Idle rate	Time spent
5	5x1	1	25
4 (1 faulty)	1x0.5,3x1	1	25
3 (2 faulty)	2x0.5, 1x1	1	35
2 (3 faulty)	1x0.3, 1x0.5	1	55

SCENARIO 3

Functional units	Working rate	Idle rate	Time spent
5	5x1	1	25
4 (1 faulty)	4x0.8	1	25
3 (2 faulty)	3x0.6	1	30
2 (3 faulty)	2x0.4	1	45

Figure 16. Results of modeling (when rate of transition from idle to engaged state is equal to 1 tu).

1. The third research question is the question of how different variants of system recovery can influence the consequent diagnosis of intermittently faulty units.

For system self-diagnosis the assumptions about the possible types of faults are very important. Generally faults can be permanent or intermittent. Units can be either permanently or intermittently faulty. In most cases, researchers consider that units can be either fault-free or permanently faulty only. However, there are situations when some system units can be intermittently faulty. Such situations were called as "hybrid faulty situations".

Intermittent fault can be assessed only in relation to the duration of testing procedure (TP) (see Fig. 17).

Figure 17. States of unit while testing procedure.

Where TP_1 – evaluates the state of unit as permanently faulty;
TP_2 - evaluates the state of unit as intermittently faulty;
TP_3 - evaluates the state of unit as fault-free;
Usually, for modeling of intermittent fault the Markov process is used. In the given case, intermittent fault of a unit can be either in passive state (PS) or active state (AS) (see Fig. 18).

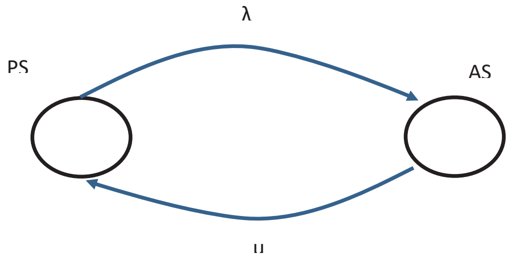


Figure 18. State transitions of intermittent fault.

After performing testing procedure, it is needed to make decision about the unit state. If testing procedure did not catch the intermittent fault in AS, then the problem arises either to continue the testing procedure or terminate it with the result that unit is fault-free.
In the given case, it is possible to consider the following two hypotheses:
H_1 - unit is fault-free;
H_2 – unit is intermittently faulty, and intermittent fault is in PS.
The posterior probabilities of these hypotheses can be computed by using Bayes' rule.
For computing the sought posterior probabilities $P(H_1/R)$ and $P(H_2/R)$ it is needed to have the conditional probabilities $P(R/H_1)$ and $P(R/H_2)$ determined.

$$P(H_1/R) = \frac{P(H_1)P(R/H_1)}{P(H_1)P(R/H_1) + P(H_2)P(R/H_2)}$$

$$P(H_2/R) = \frac{P(H_2)P(R/H_2)}{P(H_1)P(R/H_1) + P(H_2)P(R/H_2)}$$

Where R is the obtained syndrome. Conditional probabilities $P(R/H_i)$, $i=1,2$, depend considerably on the chosen testing procedure. In case, when testing procedure consists in performing mutual testing, these conditional probabilities depend on the structure of testing assignment among the system units.
For making decision whether a unit is fault-free or it possesses an intermittent fault, the following likelihood ration can be used

$$\chi = \frac{P(H_1/R)}{P(H_2/R)}$$

The decision is made on the basis of the chosen threshold ω.
Thus,
 if $\chi \geq \omega$, then the hypothesis H_1 is accepted;
 if $\chi \leq 1/\omega$, then the hypothesis H_2 is accepted;
 if $1/\omega < \chi < \omega$, then additional measures should be undertaken to increase our confidence in the decision about unit state.
In the performed research, the students conducted the testing procedure until the intermittent fault has been detected.
In order to assess how different methods of system recovery can impact the time of intermittent fault detection, the modeling of testing procedure was performed by using Petri Net.
For this purpose the students modeled the system with three units. One of the units has intermittent fault. Parameters of intermittent fault changed. Three cases were considered, particularly μ=1; μ=5; μ=10. In all cases λ=1 and rate of unit transition from working state to idle state is equal to 1. After faulty unit was detected, system recovery has been performed according above mentioned scenarios. Results of modeling are shown in the student's slide (see Fig. 19).

scenario 3

Time spent in idle states	Detectable time of failure	Time spent to find Failure
1/10	1/10	300
1/10	1/5	210
1/10	1	90

scenario 2

Time spent in idle states	Detectable time of failure	Time spent to find Failure
1/5	1/10	150
1/5	1/5	105
1/5	1	45

scenario 1

Time spent in idle states	Detectable time of failure	Time spent to find Failure
1	1/10	30
1	1/5	25
1	1	20

Figure 20. Results of modeling.

As the results of modeling show, scenario 2 of recovery is better than scenario 3. Scenario 1 can be considered as the best scenario, although some system tasks will not be performed.

Conclusion

Recovery from faults in the complex systems, which use mutual testing to detect the failed units, has impact on the following checking and diagnosis procedures.
Generally, many variants of system recovery can be suggested. To consider these variants of system recovery it is needed to have much time and perform great volume of research (i.e., simulation and modeling). Students in their researches have considered system recovery which is based on the strategy of gradual degradation (i.e., failed unit is removed from the system and, consequently, number of units in the system reduces). They considered only three variants of system recovery and performed their modeling.
After system recovery, checking and diagnosis procedures will change. Different variants of system recovery have different impact on these procedures. Mutual testing will last longer since the system units will stay more time in the engaged state (since they take over the tasks of the failed unit). Diagnosis procedure will last shorter since number of units in the system becomes lesser (i.e., complexity of diagnosis algorithm reduces).

As the research showed, the table diagnosis algorithm can be used efficiently for the system with small number of units (e.g., less than six units). After system recovery, the complexity of such diagnosis algorithm reduces essentially. This result is very important in order not to excess the maximal time allocated for all procedures (checking, diagnosis and recovery). However, it should be noted that other diagnosis algorithms can also be examined and evaluated.

Modeling of mutual testing was performed by using Petri Net. SHARPE was chosen as a tool for modeling. Results of modeling have showed that it is possible to determine of how different variants of system recovery influence the time of mutual testing which will be performed after system recovery.

Variant of system recovery when all tasks performed by the failed unit will be omitted is used as an item for comparison. It is evidently, that this variant allows providing the "best" characteristics of the following mutual testing. Variant 2 and 3 allow providing approximately equal characteristics of mutual testing for the systems with small number of units.

For each particular complex system when its main characteristics are known (i.e., number of units and their loading with system tasks), it is possible to determine the most efficient variant of system recovery and evaluate the system fault-tolerance. Some tasks of this difficult problem were considered in the students' researches.

References

[1] I. Qin, X. He, D. Yzou. "A survey of fault-diagnosis for swarm systems. Systems Science and Computer Engineering", Vol.2, pp. 13-23, 2014.

[2] V. Mashkov. „Task allocation among agents of restricted alliance", Proc. of IASTED ISC'2005 conference, Cambridge, MA USA, pp. 13-18, 2005.

[3] A. Mahapatro, P. Khilar. "Fault diagnosis in wireless sensor networks: a survey IEEE Commun Surv Tutorials, Vol. 15, pp. 2000-2026, 2013.

[4] M. Daigle, X. Koutsoukos, G. Biswas "Distributed diagnosis in formation of mobile robots", IEEE Transactions on Robotics, 23(2), 353369, 2007.

[5] V. Mashkov. „Selected problems of system level self-diagnosis", Lviv: Ukrainian Academic Press,, ISBN 978-966-322-365-0, 184 pages, 2011.

[6] A. Somani „A System Level Diagnosis: A Review" ", 1997. Available from www: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.52.9488.

[7] C. A. Petri "Kommunikation mit Automaten", Bonn:Institute fur Instrumentelle Mathematik, Schriften des IIM, Nr. 3, 19622.

[8] V. Mashkov, J. Barilla, P. Simr, „Applying Petri Nets to Modeling of Many-Core Processor Self-Testing when Tests are Performed Randomly", Springer, Journal of Testing, 2013, DOI: 10.1007/s10836-012-5346-8.

[9] V. Mashkov, P. Simr, J. Barilla. Chapter 5 „Applying Petri Nets to Modeling of Complex Systems Diagnosis" in book „Advances in Engineering Research" Ed. V.M. Petrova. NOVA science publishers Vol.13, pp.81-120, 2016. (URL: https://www.novapublishers.com)

[10] J. Dugan, K. Trivedi, R. Geist, V. Nicola "Extended stochastic Petri nets", in Proc. PERFORMANCE'84, Paris, 1984.

[11] A. Bobbio „System modeling with Petri Nets". In: A.G. Colombo and A. Saiz de Bustamante (eds.) System Reliability Assessment, Kluwer, pp. 102-143, 1990.

Artificial Intelligence in Industrial Automation

Abstract

Deep Learning has gained significant traction in recent years and has driven advancements in automatically recognizing patterns in data that surpass humans. Deep Learning has overcome the limitations of traditional machine learning algorithms as it has achieved astonishing results in a wide range of applications such as pattern recognition, speech recognition, computer vision, digital assistants, and IoT, etc.

On the other hand, deep learning models typically require specialized hardware due to higher memory and computational requirements. The solution is either more efficient hardware or less complex deep neural networks. A balance between these two aspects is essential to achieve optimal performance. Therefore, the underlying research question is: What kind of edge computing hardware and what kind of architectures should we combine to maximize the accuracy and speed of deep learning algorithms in order to satisfy the industry's demands?

Motivation

Registered students joined from the two departments: Automotive Software Engineering and Embedded Systems. In both departments, students have theoretical and practical knowledge of computer science, linear algebra, mathematics, and programming languages (Python/C++), as well as a basic background in artificial intelligence and deep learning algorithms. This course offers a series of topics within a deep-learning algorithms that provide students with an overview and comprehensive understanding of the state-of-the-art in industrial automation approaches in AI that will be applied to automotive and drone technologies.

The students studied how to build Deep Learning models and how to optimize performance for visual perception tasks such as multiple object detection, monocular depth estimation, visual odometry, and isolator object detection. The students developed and trained Lightweight Deep Learning models running on an embedded graphics processing unit (GPU) to identify objects in offline video/live stream cameras, such as roads, pedestrians, other vehicles, and isolator objects. Participants implemented the Deep Neural Network on the Nvidia Jetson Nano board platform. At the end of the course, students had the skills necessary to create AI applications for a variety of automotive and drone use cases, including traffic navigation, obstacle avoidance, and pattern recognition.

Introduction

Deep Learning delivered a tremendous boost to the already rapidly developing domain of computer vision. Many new applications of computer vision techniques have been pioneered with Deep Learning and have now become part of our daily lives. Such applications include multiple object recognition, tracking, pattern recognition, speech recognition, and natural language processing.

The main objective is to introduce students to computer vision, starting with the fundamentals and then moving to more advanced deep learning models, including image classification and labeling, object recognition, various object detection techniques, motion estimation, and object tracking in video or live streaming. Through the course projects, students taught how to build a different deep learning model and create a manipulation system to understand the internal mechanisms of these technologies.

Deep Learning is based on the Convolutional Neural Network (CNN) [1] and enables computer models with multiple processing layers to simultaneously learn and represent data at multiple levels of abstraction, mimicking our brain's perception and understanding of the way it perceives information to implicitly capture complex structures of large datasets. Deep learning has evolved into a rich family of methodologies that includes neural networks, hierarchical probabilistic models, and a variety of unsupervised and supervised feature learning algorithms.

The recent growth of attention to Deep Learning methods is due to the fact that they have been proven to outperform former state-of-the-art techniques on multiple tasks, as well as the abundance of sophisticated data from multiple sources (e.g., visual, audio, medical, social, and sensory data).

Research Questions and Methods

Different deep learning architectures such as YOLO [2], DispNet[3], ResNet [4] and VGG[5] are explored and implemented using deep learning frameworks such as TensorFlow [6], Keras [7] and libraries such as OpenCV[8], NumPy and Scikit-learn. For the training phase, several datasets such as KITTI [8], NYUv2 [9] or custom datasets are employed and used to build deep learning models.

In the context of the proposed topics, students gain knowledge of the most important aspects of deep and machine learning techniques in a practical, simple, and easy-to-understand way. The offered themes provide students with hands-on experience in training deep learning and machine learning models with real-world datasets. These Topics cover multiple techniques in a practical way that include, but are not limited to, projects:

- **Automotive Technologies:**
  - Estimation Camera Trajectory with Visual Odometry
  - Estimation a Robust Lane Departure Warning System
  - Lightweight Monocular Depth Estimation On Embedded Systems
  - Toward CNNs Visualization For Estimation Depth From Monocular Camera

- **UAV (Drones) Technologies**
  - Insulator Color Change Detection
  - Insulator Holder Detection

Implementations and Demonstrations

The enrolled students reviewed the relevant literatures and gathered a strong overview of the state of the art on each of offered topics. To realize this goal, they implemented their models for Deep Learning using cloud servers such as Google Colab or standalone GPUs. In the evaluation phase, they tested their CNN model on an embedded device such as the NVIDIA Jetson Nano.

Google Colab is an operational AI platform for creating, training, and evaluating machine learning and deep learning models. It could be used as a free cloud-based Jupyter notebook environment to train our machine learning and deep learning models on CPUs, GPUs, and TPUs. It is capable of running all popular machine learning frameworks, such as TensorFlow, PyTorch, and Keras.

On the other hand, Jetson Nano is basically a small, full-featured Linux computer designed for entry-level AI applications and devices, which makes it very flexible in terms of software usage. It is capable of running all the popular machine learning frameworks, such as TensorFlow, Caffe, PyTorch, and Keras. Generally, the code that is used to deploy a model on a desktop GPU or in cloud services can be adapted to the Jetson Nano with minor adjustments. Due to its small size, the Nano can be easily used in mobile scenarios where there is no space for a large desktop system.

Results

Estimation Camera Trajectory with Visual Odometry:

Visual odometry (aka. Ego-motion estimation) is the process of determining the location and orientation of a camera by analyzing a sequence of frames obtain from motion camera. Visual odometry is used in a variety of applications, such as mobile robots, self-driving cars, and unmanned aerial vehicles. The main challenge was to design a light deep learning model to estimate monocular visual odometry using deep neural networks. Finally, the proposed deep learning model should be able to run on NVIDIA Jetson Nano broad in real-time capabilities.

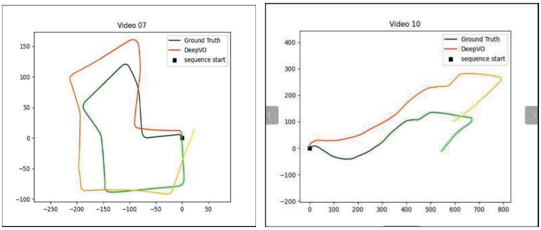


Fig.1.Visual Odometry generated for deep VO for trained model over dataset

The evaluation of the achieved results is based on two main factors according to the evaluation metrics of the KITTI dataset. The evaluation of the achieved results is based on two main factors according to the evaluation metrics of the KITTI dataset.

- t_{rel}: Average translational RMSE drift, this error basically defines the average point to point Root Mean Square Error (RMSE) when measured the total distance travelled by the camera.
- r_{rel}: Average rotational RMSE drift (deg/100m) this error denotes the average point to point RMSE of the heading error or to be put in layman terms change in direction from the actual path in deg/100m.

Table 1: Comparison with monocular visual odometry

Sequence	Deep-MVO		VISO2_M	
	r_{rel}	t_{rel}	r_{rel}	t_{rel}
04	0.015	9.8	0.0449	4.69
05	0.092	42.57	0.1758	19.22
06	0.172	44.07	0.0614	7.30
07	0.104	11.42	0.2911	23.61
10	0.031	42.5	0.3299	41.56
mean	0.082	29.84	0.1806	19.27

Comparing the results of trel of both models (Deep-MVO and VISO2M), the overall reliability of VISO M is better but consider a case such as sequence 7 where deep-MVO is more reliable. Comparing the results of rrel of both models, the mean results of deep-MVO are better but considering a sequence such as 05, VISO M is a better model to use. Hence, again in this case there is a requirement for consistency for the model to be more reliable.

Estimation a Robust Lane Departure Warning System:

Lane departure warning systems have become an integral part of vehicle safety when producing these systems. Manufacturers have to conform to strict regulations with NHTSA standards [9] stating that the driver must be alerted 0.75 meters before and 0.30 meters after the lane crossing.

The proposed task was to design a light deep learning model to allow test the lane departure system robustly with estimated steering wheel angle. The proposed model should be able to estimate the distance between the vehicle's fronts corner and the lane edge within a predefined threshold position accuracy from data captured from the front-facing camera. This information is seamlessly integrated with an estimated steering wheel angle. Finally, the proposed deep learning model should be able to run on NVIDIA Jetson Nano broad in real-time capabilities. When the lane has curve towards left and the vehicle is getting nearer to any of the lane lines the green indication changes to red showing that vehicle is near to the lane and if it keeps on going in that direction it will be departed from the lane. Similarly, when the lane curve towards right side. For the lane which is straight, it is not showing any indication.

Fig. 2. Before applying the model for the lane departure warning system

Fig. 3. After applying the model for the lane departure warning system

Lightweight Monocular Depth Estimation On Embedded Systems:

Monocular depth estimation is essential for understanding 3D scene geometry and is a challenging task for computer vision. Recently, CNNs have been used to deal with this challenge, which has shown promising performance and led to a significant improvement in estimation accuracy. On the other hand, it is widely unclear why and how CNNs are able to estimate the depth of a scene from their monocular image age. Basically, they are black boxes like other tasks. However, memory and computing power constraints remain as major challenges that must be addressed in these deep learning models. The target task was to develop an efficient and lightweight deep learning of fast depth estimation on embedded systems (such as the Jetson nano board) and apply network pruning to further reduce computational complexity and latency.

The acquired model delivered state-of-the-art results for monocular images on the KITTI dataset, outperforming some supervised learning techniques. Despite the great results, there is some occlusion at the edges of the images.

Fig.4. Depth prediction result on KITTI 2015. At left, single view input image. At right, the obtained result of depth map prediction.

The students also provided an overview of previous monocular depth estimation models and the latest dataset that can satisfy the requirements to train a model. Furthermore, a series of experiments were conducted to optimize selected hyperparameters of a monocular depth estimation model. As a result, the best hyperparameters were found with low computational cost. The evaluation metrics showed the best training accuracy with a learning rate of 1e-5 and a loss weight of 0.55.

Toward CNNs Visualization For Estimation Depth From Monocular Camera:
Recently, psychophysical studies have shown that human vision uses several cues for estimating monocular depth, such as linear perspective, relative size, interposition, texture gradient, light and shadow, aerial perspective, etc. The fundamental question is, of course, do CNNs actually use these cues? Answering and studying this question will help us understand why CNNs can / cannot estimate the depth of the RGB image. The main task was to identify and visualize the image pixels that are most relevant for depth estimation, which plays an important role in depth estimation processes. The proposed method has been tested on the pre-trained model on KITTI and NYUv2 dataset. Also have obtained the corresponding faithful depth predictions for both dataset as shown in fig5.

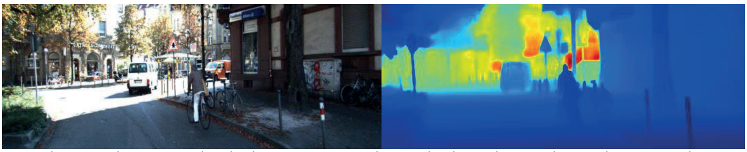


Fig 5. left: RGB image KITTI dataset. Right: Corresponding depth image using Jet colormap, the shorter distance object to the camera, colormap represent blue, the farthest object, the colormap represents red and object between short and farthest distance object represents green

Insulator Color Change Detection:
The goal of the research internship was to implement a deep-learning based high voltage power line insulator color change detection model from given input data (image/video).

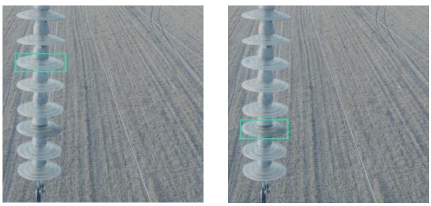


Fig 6 shows an example of a false positive on the left image as the bounding box is surrounding a functional insulator and a true positive on the right in which the discolored insulator is detected and bounded by a box.

Insulator Holder Detection: The goal of the research internship was to implement a deep-learning based high voltage power line insulator to detect an insulator holder from given input data (image/video).

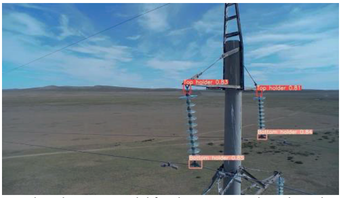


Fig 7 shows a deep learning model for detecting insulator based on YOLOv4.

Conclusion

Deep Learning is a cutting-edge technology that is becoming common and widely used in various industries. It is also one of the most intensively researched areas in computer science. There are several neural network architectures that have been implemented for different types of datasets. Of these architectures, convolutional neural networks have reached the state of the art in the fields of image processing techniques. In this context, students in the IKON project were able to develop several functions related to automotive and drone technologies based completely on deep learning approaches. They also acquired a strong knowledge of neural networks, which they could employ on any project or company in the future. They learned how to implement image processing algorithms using OpenCV and how to train convolutional neural networks, such as detecting multiple objects in real time, while working on the embedded hardware such Jetson Nano Board.

The primary goal was to teach and mentor students with programming skills, mathematics, and the fundamentals of artificial intelligence algorithms and techniques to develop and enhance their skills to meet future technical challenges.

References

[1] Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: an overview and application in radiology. Insights Imaging 9, 611–629 (2018). https://doi.org/10.1007/s13244-018-0639-9
[2] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
[3] Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; Brox, T. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation., 27–30 June 2016; pp. 4040–4048.
[4] He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
[5] Simonyan, K.; Zisserman, A. Intense convolutional networks for large-scale image recognition. arXiv Prepr. 2014, arXiv:1409.1556.
[6] Abadi, Martín et al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems." ArXiv abs/1603.04467 (2016).
[7] Gulli A, Pal S. Deep learning with Keras. Packt Publishing Ltd; 2017.

[8] OpenCV. Open Source Computer Vision Library. 2015.
[9] NHTSA issues Federal Motor Vehicle Safety Standards (FMVSS) to implement laws from Congress. https://www.nhtsa.gov/laws-regulations/fmvss

Automation and remote control of micro hydro using Scada system

Abstract

Micro hydro plants generate clean electricity by converting the mechanical energy of water into electrical energy using sets of turbines and generators. This transformation is made possible by turbine and generator sets. The article describes how these sets, but also other key components of the power plant can be controlled using programming logic controllers and how to remotely control these controllers and visualize the operation of the power plant via Internet using Scada software. The individual sensors are selected and on basis of the data from them the developed control program reacts accordingly to ensure safe reliable and efficient operation of the power plant.

Key worlds: PLC. Scada, control, sensors

Motivation

Our group consists of Information technology students focused mainly on hardware programing and networking from Department of Informatics, Faculty of Science Jan Evangelista Purkinje University. This project offers us a great opportunity to use our theoretical knowledge of our field of study and put it into practice and deepening our knowledge in industrial automation in general.

We organized our team into two development groups one for PLC development and second for Scada development.

Hydropower generates major part of renewable electricity, we believe that it has its place in the future of energetics.

Introduction

Micro hydro power plant

Micro hydro power plant generates electricity uses the energy of falling water. The head is created by the damming of the water flow. The water flows through the inlet object to the engine room where the turbines are located. Water is discharged into the turbine through the stator blades, thereby rotating the turbine rotor blades. The rotating energy is transmitted to the generator shaft by transmission, the shaft rotates and due to magnetic field in generator electric energy is generated.

Programable logic controllers

Programable logic controllers shortly PLCs are industrial computers used to control the operation of equipment. They are equipped with input and output cards, which are inserted into the PLC slots, so number of input and outputs can be configured as desired. Input cards are used to collect data from individual connected sensors. Output cards are used to control devices.

SCADA

Supervisory Control and Data Acquisition server communicates with PLC and logs data from it. It allows the operator to see live data from sensors, visualizes program operation and allows operator to control the program.

Sensors

The sensors are used to obtain data from the individual parts of the device needed for the program to run and provide this data to the control system.

Induction sensors

PLC

Productivity 1000

Research Questions and Methods

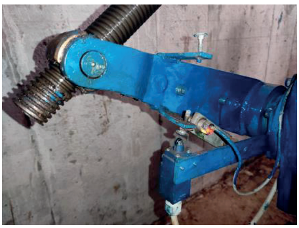
Our goal is to find a solution for creating an autonomous control system to control a small hydropower plant. We must first determine what devices we need to control with our program. Based on this, we must determine what input data we need to obtain for the operation of the program, then we must find an aqueous technical solution of sensors that will allow us to obtain this data.

Then we must choose a suitable control system for our solution with the known inputs and outputs of our program, we will create program using selected PLC and sensors.

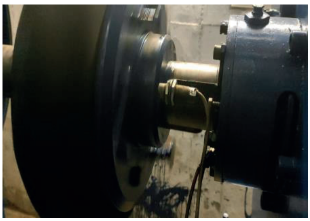
Implementations and Demonstrations

Hardware selection

We have determined that we need to control inlet gates, trash rack cleaners and turbine sets.



Limit switch on control arm of the turbine



Induction sensor for RPM measuring

For the machines to be successfully controlled, we need to know their end positions, so we chose to use inductive sensors. These sensors act as a switch and, when switched on, send us a binary signal of +24 V DC. We also need to know the water levels at the dam and behind the screens. For these purposes, we chose immersion pressure probes that have an analog output of 4-20ma. We also need to know the temperatures of the bearings of the turbine sets, for this measurement we have chosen PT100 temperature sensors. And finally, we will need signals from the auxiliary contacts of electrical elements, which will also be binary 24V DC.

We decided to use binary outputs with 24 V DC for the outputs. But because, for example hydraulic units are switched 400V AC we also use contactors and relays to control higher voltages. Most outputs are used for switching the coils of hydraulic units for controlling turbine blades and switching the control coils of contactors.

We ended up using a configuration with one binary output card with 15 outputs + 24V DC. Two binary input cards + 24V with 16 inputs, two cards with analog inputs 4-20ma and a special "fast" input card handling 100kHz binary input.

For our use, we chose PLC Productivity 1000 form Automation Direct for its easy scalability, the ability to configure it with cards for all necessary inputs and outputs, for its ability to communicate using the ModBus TCP protocol and integrated simulation tool within its programing software.

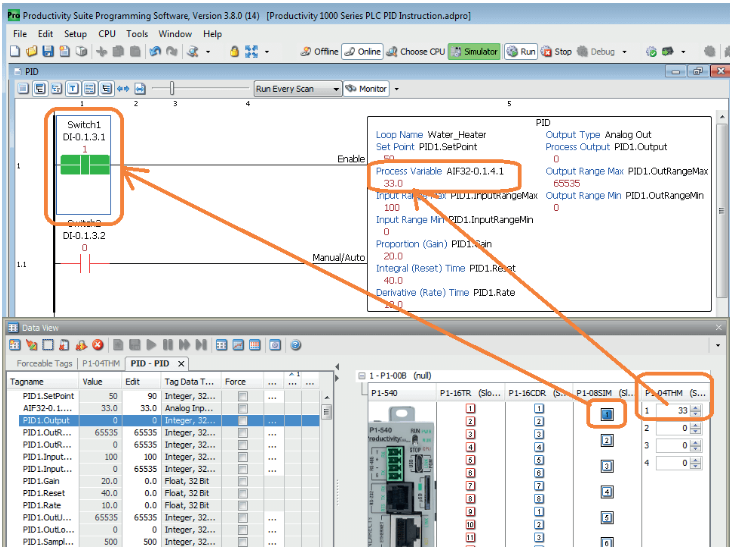
Software development

With PLC also comes its development environment called Prductivity suite, which is free and has proved very useful programing tool using ladder diagram programing language with instruction boxes and it has made it easy to work remotely in our team. As a first step, we did initial setup in our new project. We set the hardware configuration to match the actual configuration of our plc and then we connected to the PLC with a USB cable and set the IP address of the Ethernet card.

Then began the long process of developing our program. The individual parts of the program were developed and then tested using a simulator, where we tried to test all possible states that our program can get into and then modified our program so that even unfavorable or even emergency states were successfully solved by our program. It is important to say that our program was based on the handling and operating rules that our program had to follow. Our team also tried to solve possible problems caused by external influences, such as power outages

When it was concluded that the program is ready for live deployment. The PLC was installed and connected by a professional supplier for the power plant and the so-called dry tests began, ie water was not allowed into the power plant. The functionality of the input and output connections was tested. The individual sensor ranges were then tested and calibrated. First found bugs were fixed

Next real-life tests began. The individual parts of the program were gradually tested, and their correct function was verified, then emergency situations were artificially created, and it was tested whether the response of the program was correct. After tests we fixed all issues and our program was put into permanent operation, but it was still monitored by us for possible errors



PLC simulator

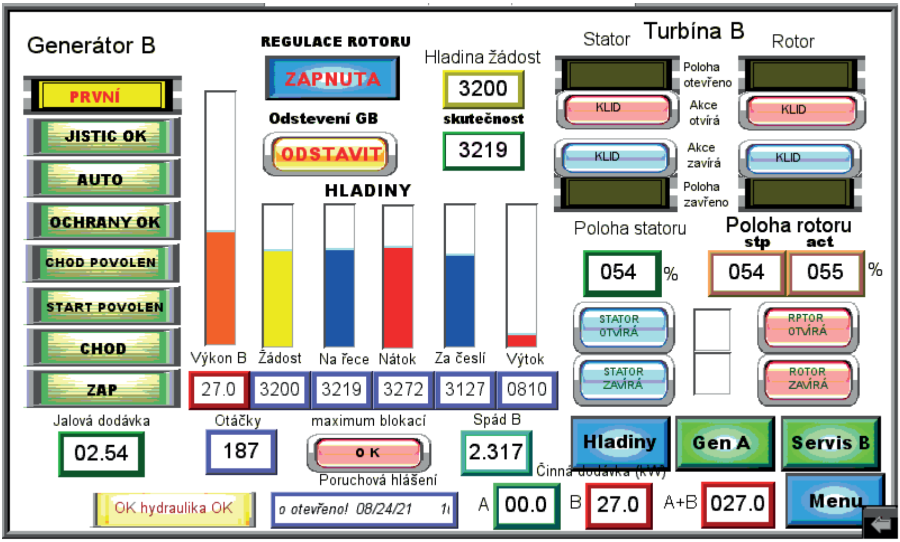
SCADA development

Simultaneously with the development of the PLC program, visualization was also developed using Scada software. Reliance Scada was chosen. Based on the requirements of the operator, the initial version of the visualization was created and according to comments and later experience from the operation, it was modified to the final form.

The Scada server communicates with the PLC via the local network using the Modbus TCP protocol, which is the communication protocol for device communication in the industry. Its advantage is that it is universal and is supported by many devices. With this protocol we also communicate with the PLA (power line analyzer), from which we display the current state of the electrical network: voltage, power factor, etc.

The visualization in the menu is divided into individual windows with parts of the program according to their focus. The primary screen is the screen of the unit located a summary of important information for the operator. In addition to information on performance, there is also information on the positions of the blades, there is also a status panel and an overview of individual levels. Other screens are already focused on specific parts.

The visualization is remotely accessible via the Internet using the VNC protocol and is designed so that it can be controlled using both PCs and mobile devices. From operational experience, our solution has been enriched with an HMI (human machine interface), ie a touch control panel present directly on the door of the Distribution cabinet.



One of final Scada windows

Results

After few months of our solution running and controlling whole power station. We can happily say that that we have successfully met our original goal. In the end, we managed to create a complete solution for the automation of a small hydropower plant. We chose measuring instruments for measuring values. We programmed, tested, and deployed the control program using the selected PLC and designed the visualization for remote operation.

EMG Signals and Processing

Abstract

The main objective of the project is to validate the usability of a low-cost and potentially wearable hardware for real-time detection of simple hand motions (grasping on the steering wheel) e.g. potentially useful for monitoring the driver's reaction in case of danger. Myoware Muscle sensor, Arduino Uno (A/D converter), Rasberry Pi (data storage) were used for the measurements. The annotated signal was used as input to several machine learning models (Logistic Regression, SVM, Simple Neural Network, LSTM) and the models with the highest accuracy were selected. Some of these models were ported to Raspberry Pi Pico (by MicroPython and uLab) and used for real-time prediction. The low temporal resolution of the sensor, as well as the different measurement conditions led to inconclusive results, but nevertheless demonstrated the applicability of low-cost microcontrollers for physiological signal processing.

Our students group consists of one first year student and three second year students and one with combined study (partial distance learning) which helped us especially with presentation. All are enrolled in applied Informatics mainly in software development ending with Bachelor's degree and are focused on programming, UI, web development, Python. Our group consists only of men.

The topic of our project was EMG signals and their processing. Electromyography (EMG) is a technique for evaluating and recording the electrical activity produced by skeletal muscles. EMG is used to diagnose neuromuscular diseases and injuries, kinesiology, and musculoskeletal disorders. EMG signals can also be used to control prosthetic replacements such as hands or feet.
It can be also used to detect human movement and reactions in various situations, which for example can be used to detect drivers behavior in car. Unlike an ECG, an EMG has a disadvantage that signal does not have clear features that are easy to interpret and thanks to that we can apply modern technologies such as machine learning. Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. These technologies were characterized by expensive equipment, but thanks to the development of modern technologies, sensors and microcontrollers can be used on a mass scale.

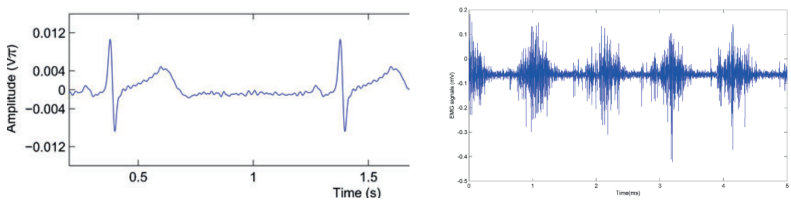


Figure 1: EKG (left) versus EMG

Introduction

For practical demonstration of cooperation of EMG sensors and machine learning the real-time detection of hand motion is classical use case. The solution uses three stage system:

1. initial measurement, processing and storage of digital data (including data and annotation)
2. design of concrete machine learning models based on the annotated data
3. using the model for real-time event detection and signaling

For the first stage, the use of hardware is important, as it is necessary to provide
measurement of the analogue EMG signal (surface and needle measurements are possible, due to the focus of the project surface measurement was chosen, which only requires placing electrodes on the skin surface)
conversion of the analogue signal to digital via an A/D converter
data transfer to a data storage device (e.g. a file on a connected computer)

A special requirement of this stage is the annotation of the acquired signal, determining at what time point the hand movement has been initiated. There are several possible solutions. The simplest is to use an audio signal to which the test subject responds with an appropriate movement.

Design of machine learning models requires the following sub-stages:

1. data preprocessing (various artifacts are commonly found in the data)
2. design of the metamodel
3. creating a concrete model based on the stored annotated signals using supervised learning
4. model and metaparameter optimization

The last stage combines the integration of hardware and software, which has the following basic functions:
measurement of the EMG signal (using the same sensor and conditions as the initial measurement).
real-time event detection based on the input signal
event signaling (visual or by informing a higher-level information system)

Real-time EMG signal processing to detect different body part movements has been addressed in different projects focusing on different aspects of EMG and using different machine learning approaches (e.g Zhang 2019, Côté-Allard 2019, Zhang 2020).

Most of them are characterized by the use of relatively complex sensors and processor-intensive processing (software signal filtering, use of challenging machine learning methods).
However, if we assume that the EMG signal can potentially be used to monitor the muscular responses of people during common activities, for example, a driver driving a car, this approach is not very suitable as:
the sensor and preferably the whole device should be wearable.
the processing must be real-time, but with minimal power consumption, so that e.g. power from conventional batteries can be used
the whole device must be cheap enough to allow mass deployment (on the order of tens of Euros)

A model case are the fitness watches, which meets all the above assumptions and already today uses some physiological signals including machine learning based processing. Fitness watches currently do not include an EMG sensor and their placement is inappropriate for the purpose of hand motion detection, but similar devices with more appropriate placement and support for additional sensors can be expected to emerge.

Research Questions and Methods

Based on existing hardware, existing solutions and students' knowledge, we have chosen the following min goal:

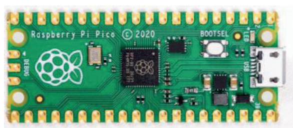
The applicability of simple hand motion detection using potentially wearable sensors and microcontrollers with a total cost in the low tens of euros. (an exception is the process of creating machine learning models including training, but this can be done on existing student laptops. For hand motion detection, a relatively simple motion to define was chosen: the strong grasp (of fist). This movement made on the steering wheel can signal that the driver is instinctively reacting to danger.

In the case of the hardware used, the choice of a device that would perform real-time muscle movement detection was crucial. This device must have the characteristics of a microcontroller board (Arduino is a typical representative) and at the same time have the performance to apply a machine learning model (i.e. perform fast vector and matrix operations and handle transcendental functions).

Direct support of the appropriate frameworks or libraries is also advantageous, allowing easier implementation of the corresponding model. After a search of existing libraries, we chose two:

Tensor Flow Lite for Microcontrollers (https://www.tensorflow.org/lite/microcontrollers)
MicroPython including extension uLab (analogy of NumPy, see https://github.com/v923z/micropython-ulab).

For this reason, we mainly considered the following two microcontroller boards.

feature	MH-ET LIVE ESP32 MiniKIT, Wemos D1	Raspberry PI Pico
		
Processor	dual core, Tensilica LX6 240 Mhz**	Dual-core Arm Cortex M0+, 133 MHz
SRAM	520 KiB	264 KiB
Flash memory	4 MiB	2 MiB
Wifi /Bluetooth	yes	no
MicroPyton port	yes	yes
uLab	yes	yes

Tensor Flow Lite	yes	beta
Price*	9 €	5 €
Size	21×51 mm	28×37 mm

* September 2021 real prices on https://www.laskarduino.cz
** only one core is available for user code

Even though the ESP32 Wemos board has better hardware specs (with a slightly higher price) and stable support for Tensor Flow Lite, we ultimately chose the Raspberry Pi, whose hardware is substantially open-source and above all well documented (even though it was released in January 2021).

Implementations and Demonstrations

Therefore, the following hardware solution was chosen for the final implementation:

A) Initial measurement, processing and storage of digital data
Myoware Muscle sensor (https://www.sparkfun.com/products/13723)
Arduino Uno as ADC
Raspberry PI as signal storage

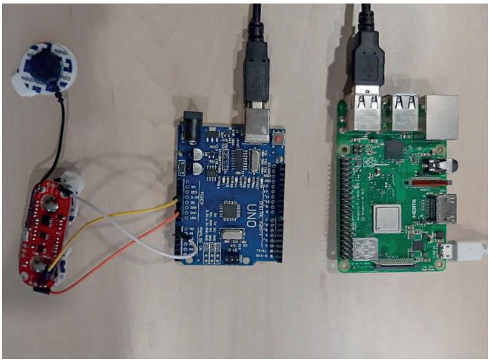


Figure 2: Myoware Muscle sensor, Arduino Uno and Raspberry Pi

The assembly could be optimized by using Raspberry PI also as an A/D converter, but Arduino is better suited for processing of analog output of sensors.

B) design of specific machine learning model based on annotated data
common student laptops

C) using the model for real-time event detection and signalling
Myoware sensor
Raspberry PI Pico

The Raspberry PI Pico could act as an A/D converter and even a primary data storage device, but at the time the device was designed and implemented for the initial stage it was not available.

The initial measurements used a game wheel to mimic the target environment and a simple randomly spaced sound signal generator (provided by smartphone). The measurements took into account some basic requirements for sensor placement (approximately at the position of the ulnar end of the muscle Flexor digitorum profundus, see Figure 3). Two subjects were followed with measurements of ten minutes (grip intervals were a round of ten seconds). Further hand motions were very limited or slow. The acquired signal reached a sampling frequency of 100 Hz at most.

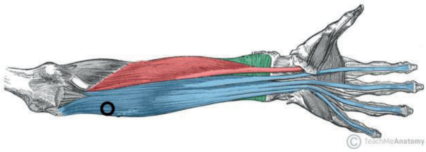


Figure 3: Approximate location of the surface sensor (black circle)

Figure 4: Initial measurement (the sensor is at alternative location)

Signal preprocessing involved two operations:
removal of the signal fragments with significant noise (caused by insufficient contacts)
signal normalization (raw values from A/D converter are normalized to range [0,1])
normalization of the annotation (the reaction time i.e. time between the sound signal and its manifestation in the signal has a large variance from about 0.2 sec to 1.2 sec, after a preliminary automatic finding of the leading edges and reaching a fixed threshold, the timing of the first peak was manually added)

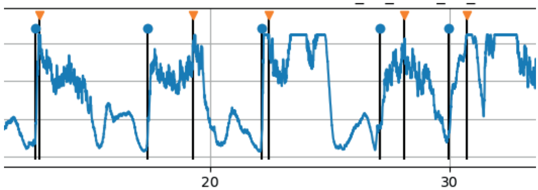


Figure 5: Annotated signal (blue circle – threshold, orange triangle first peak)

This preprocessed data was then used as input for both the training and validation phases of the machine learning. Sliding windows with widths in the order of tens of measurements were used for recognition (the number is one of the input metaparameters of the model).
Using the python libraries scikit-learn and keras, the following models has been created:

- Logistic Regression
- Support Vector Machine (SVM)
- Neural network with one hidden layer (the number of elements of hiden layer is another metaparameter)
- LSTM (recurrent neural network with convolution) with a larger number of layer metaparameters (see Figure)

The models and their metaparameters were then optimized. The optimization is based on various characteristics derived from the number of true/false positive/negative detections (e.g. accuracy, precision). Another characteristic taken into account is he statistical distribution of the detection delay and especially its maximum value (however, this value was only used in the determination of the sliding window length).

The results of the individual methods are presented in the following overview:

NAME	TRUE	FALSE POSITIVE	FALSE NEGATIVE	TRUE POSITIVE	MAX. DELAY
Convolution	891	8	23	78	307.844
Convolution (first max)	885	18	23	74	459.2
Deep learning	849	16	12	123	215.712
Deep learning (first max)	899	17	18	66	407.283
Support Vector Machine	889	5	14	92	108.995
Support Vector Machine (first max)	904	12	22	62	165.261

As can be seen, the detection is not very successful and therefore not applicable in practice. This may be due to the limited amount of input data, but is more likely due to the low resolution of the sensor, which does not allow to distinguish compound movements (e.g., pressing while rotating). This is probably also the reason why more complex models do not provide better results.

In the next phase, the possibility of porting the optimized model to the Raspberry Pi Pico microcontroller board was investigated. Two strategies were tested: directly porting the model from the keras library to Tensor Flow Lite and implementing a custom library for simple models (logistic regression, neural network) in the uLab framework.
The first approach, which is more flexible and less demanding in terms of coding, could not be verified as it was not possible to compile the model for the target platform. Therefore, the second approach was used in which a model for linear regression and a simple neural network was manually coded. (based on parameters exported form keras library). By using the vector and matrix operations provided by the uLab library sufficient performance is achieved even for a microcontroller (unfortunately, the MIMD parallelism provided by the Rasbbery Pi Pico processor is not supported).
During the testing, the correctness of the implementation was verified (by comparing it with the predictions provided by scikit-learn and keras libraries) and the execution time of detection was checked. The processing time for both logistic regression and neural networks with a smaller number of nodes (with a central matrix between the input and internal layers of dimension 30×60) is below the 1ms threshold. Thus, the implementation can be used to process signals with a frequency of 1 kHz or more complex models can be implemented.
In the final phase of the project, we connected the Rasperry PI Pico with the implemented logistic regression model and EMG sensor and used the built-in LED to signal the detected strong grasps. This allows to check the correctness of the detection in real time.

Summary and Results

The main outputs of the projects:

- annotated signal data for several test subjects
- machine learning model with optimized metaparameters
- implementation of simple inference (prediction) part of simple ML models by MicroPython and uLab

- testing prototype of detector of hand motion (based on Rasberry Pi Pico)

The current accuracy of real detection is not satisfactory and cannot be applied in practice (except for the trivial detection of arbitrary hand movement) because the proportion of false positive cases is too high.

The main reasons for this behavior, in our opinion, are:

- low temporal resolution of the sensor
- use of only one channel (key muscles are used in many movements)
- the difficulty of ensuring the same conditions in different measurements (i.e., measurements used to train ML models and measurements used for prediction). This problem is amplifired by the use of two different A/D converters (linear re-scaling is only partial solution).

Future work is also needed to address the installation of Tensor Flow Lite on Raspberry Pi or to test the use of alternative microcontrollers for ML inference in practice. This issue will be addressed in the bachelor theses of the students participating in the project.

However the computer capabilities of microcontrollers with modern CPUs (and accelerators) make possible real-time processing and machine learning predictions on signal with frequencies up to 1 kHz by high-level programming environment as MicroPython an uLab.

References

Zhang2019
Zhang, Z., Yang, K., Qian, J., & Zhang, L. (2019). Real-Time Surface EMG Pattern Recognition for Hand Gestures Based on an Artificial Neural Network. Sensors (Basel, Switzerland), 19(14), 3170. https://doi.org/10.3390/s19143170
Zhang2020
Zhou Y, Chen C, Ni J, Ni G, Li M, Xu G, et al. EMG Signal Processing for Hand Motion Pattern Recognition Using Machine Learning Algorithms. Arch Orthop. 2020; 1(1): 17-26.
Côté-Allard2019
Côté-Allard, U.; Latyr Fall, C.; Drouin, A.; Campeau-Lecours, A.; Gosselin, C.; Glette, K.; Laviolette, F.; Gosselin, B. Deep Learning for Electromyographic Hand GestureSignal Classification Using Transfer Learning. arXiv 2019, arXiv:1801.07756v5.

Situation Awareness based on Vision Systems Using the "CE-Box" of Computer Engineering, TU Chemnitz

Abstract

The automotive industry is moving towards the highest level of autonomy with rapid pace. One of the major obstacles reaching the highest level of autonomy is object detection and collision avoidance. This project concentrates on different object detection algorithms using image processing and machine learning techniques while maintain the definition of situation awareness. All of the algorithms are implemented and evaluated under the Raspberry pi 3b+ models which are part of the demonstrative hardware "CE-Box", developed under the department of computer engineering of TU Chemnitz.

Motivation

The focus of this topic is to develop concepts and implement the developed concepts of different types of object detection, which are required for higher level of autonomy. For this purpose, this topic is divided into 5 individual tasks. They are:

Task 1: Pedestrian Detection using Image Processing and Machine Learning techniques.
Task 2: Traffic Light and Backlight Recognition using Deep Learning and Image Processing.
Task 3: Optimization of Different Image Processing Algorithms.
Task 4: Design of a GUI using python QT and Realization.
Task 5: Design of GUI for Expert System Simulation and Rule Generation tool.

Tasks 1 to Task 3, belongs to the vision based systems focusing on image processing and machine learning algorithms. Task 3 specifically works around the optimization area of the algorithms to enhance the performance and detection rate of the objects. One of the levels of situation awareness is called Perception, where the information both from externally or internally is perceived from the extracted and perceived. While Task 1, 2 and 3 extracts information using vision based techniques, task 4 mimics other sensors through a developed Graphical user Interface (GUI) and provides data in a time triggered manner. This date we refer to as "Synthetic Sensory Data". The realization of situation awareness is done through a forward chained ruled based system, known as the Expert System. Identification of the situation along with what commands to execute, these decisions are taken via the expert system. Task 5 had been given the job to design a Graphical User Interface where an expert system can be simulated and the situations can be evaluated.

As mentioned before, all of the tasks have been performed based on the Raspberry Pi 3b+ model. It was known from the beginning, that the Raspberry pi 3b+ models has some lacking while it comes to processing capabilities along with no graphical processing unit as well. It was only used to showcase as a prototype for the algorithms to run. Hence, it was decided that individual algorithms will run in individual Raspberry Pis. Therefore, a communication between the Raspberry Pis became a must. To solve the problem of communication, a Master Slave communication using adhoc was developed and this can be showcased below:

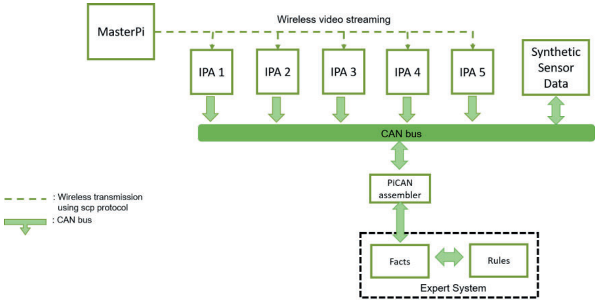


Figure 1: Communication System for the Raspberry Pi's.

Student Background

A total number of 10 students registered to participate in this project. These are all Masters Students from the departments of Embedded Systems and Automotive Software Engineering of TU Chemnitz. Out of these 10 students 3 students were from the department of "Automotive Software Engineering" and 7 of the students were from "Embedded Systems".

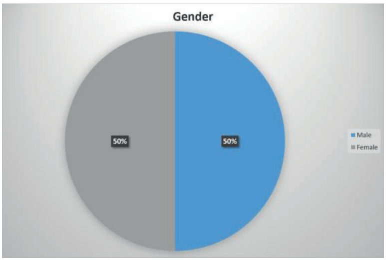


Figure 2: Gender ratio of the students

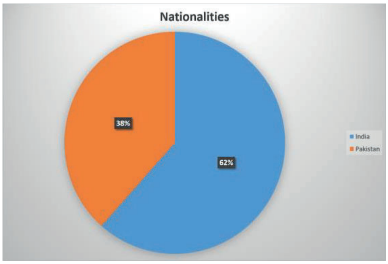


Figure 3: Nationality of the students

The Gender ratio was balanced to 50% male and 50% female students (Figure 2). The students' nationalities are a mix from India and Pakistan, where 80% of the students were from India and 20% of the students were from Pakistan (See Figure 3). Students participated in this Project in order to successfully complete their Research Internship (Automotive Student) and Research project (Embedded System), which is compulsory for them to register for a consecutive Master Thesis. The following table shows the background of the students of Digital processes block.
Groups of 2 students were created and they registered under the topics mentioned above. Below the detailed information of the students are provided:

Table 1: Students' Demographic Details

Student	Gender	Faculty	Field of studies	Country of Origin
1	Female	Faculty of Computer Science	Master Automotive Software Engineering	India
2	Female	Faculty of Electrical Engineering and Information Technology	Master Embedded Systems	India
3	Female	Faculty of Electrical Engineering and Information Technology	Master Embedded Systems	India
4	Female	Faculty of Electrical Engineering and Information Technology	Master Embedded Systems	India
5	Female	Faculty of Electrical Engineering and Information Technology	Master Embedded Systems	India
6	Male	Faculty of Computer Science	Master Automotive Software Engineering	India
7.	Male	Faculty of Computer Science	Master Automotive Software Engineering	India
8.	Male	Faculty of Electrical Engineering and Information Technology	Master Embedded Systems	India
9.	Male	Faculty of Electrical Engineering and Information Technology	Master Embedded Systems	Pakistan
10.	Male	Faculty of Electrical Engineering and Information Technology	Master Embedded Systems	Pakistan

Introduction

Tasks description for each of individual tasks are given below:

Task 1: **Pedestrian Detection using Image Processing and Machine Learning Techniques**

When it comes to safety protocols for autonomous driving, the highest level of priority is given towards Pedestrians detection. In order to maintain a safe distance to the pedestrians, detection has to be fast and accurate. For this reason, the algorithm is developed using with the combination of machine learning and image processing techniques. With the combination of Support Vector Machines (SVM) and Histogram of Oriented Gradients (HOG), an accurate algorithm has been developed with semi real time performance on the raspberry pi 3b+ model.

Task 2: **Traffic Light and Backlight Recognition using Deep Learning and Image Processing**

The main goal of this internship is to develop an algorithm which not only can detect and recognize traffic lights and status but also can detect and recognize the taillight of other vehicles in the front and inform whether the taillight status is in brake or directional indicators. A synchronized algorithm was developed by two different students working in a single group for this topic which provides accurate results.

Task 3: **Optimization of Different Image Processing Algorithms**

This internship group was given different image processing algorithms. Their task was to update the developed code with different optimization techniques in order to improve the frame rate performance of the algorithms. They re-developed the algorithms, keeping the detection algorithm intact and upgrading the frame rate performance with their approach.

Task 4: **Design of a GUI using python QT and Realization**

Due to the lack of sensors in hand the idea of developing a Graphical User Interface which mimics sensory data was created. The task of this group was to develop this GUI which will work as the interface for the autonomous vehicles, while providing information such as current vehicle
speed, engine and battery status etc. Finally, these information will be transformed in to a format called "Facts", which is an input type of format for Expert Systems. The group also created some generic rules to test the expert system as well.

Task 5: **Design of GUI for Expert System Simulation and Rule Generation Tool**

This group was tasked with development of a Graphical User Interface, in which a drive plan can be created from one point of the map to another. The GUI enables the user to drag and drop objects, such as traffic signs, lights etc. in to the map. Each object creates a "Fact" and the drive plan can also create Rules. Finally, these rules are matched with the Facts and tested using the expert system. The output is a prototypical tool which is used to simulate and test the expert systems.

Implementations and Demonstrations

The implementation was done on individual Raspberry pi 3b+ models, and showcased using the „CE-Box", which part of a is a demonstrative hardware unit called the „BlackPearl" from the department of computer engineering at TU Chemnitz. It is consisted of six slots for Raspberrpy Pis. Each of the Raspberry pi has a PiCan2 Modele mounted on them. It also has a additional touch screen for output visualization, which is also ran by a raspberry pi. The „CE-Box" is showcased in the figure below:


Figure 4: „CE-Box“ (left) & „Black-Pearl (right)“

1. Pedestrian Detection using Image Processing and Machine Learning Techniques

Different Datasets were used for the generation of the machine learning models for the inference to detect pedestrians. Initially, Daimler dataset was used, but it only performed well on images and not on videos. Then focus was given to INRIA dataset but it was also performing well on known test images and videos. Hence, a custom model was created using customized images. The training was done using SVM along with HOG from the image processing part.

2. Traffic Light and Backlight Recognition using Deep Learning and Image Processing

A combination of image processing and deep learning was used to detect the traffic light and the backlight status recognition. For the traffic light detection, color segmentation was used along with circular shape detection. For the backlight recognition, deep learning model was used to train and detect vehicles. From that point, the tail light was detected using the region of interest technique and color segmentation was done to filter red any yellow colored lights.

3. Optimization of Different Image Processing Algorithms

Different algorithms based on image processing and machine learning was provided to this group for optimisation. The group used a multi threading approach along with piplelining process to improve the frames per second rate, while maintaining the accuracy of the algorithms.

4. Design of a GUI using python QT and Realization

The tasks for this group was divided into two sections, one student worked on developing a Graphical User Interface and the other had worked on Rules and situations for an intelligent system. Graphical user interface was developed based on QT and the backend was supported using python. On the other hand, another student developed some rules as situations and evaluated them using expert systems.

5. Design of GUI for Expert System Simulation and Rule Generation Tool

The objective for this task to develop a Graphical User Interface for the evaluation and simulation of expert system rules for a certain drive plan. The students of the group intigrated a live map using open maps and developed a GUI which enables the user to drag and drop icons, which represents traffic signs, lights, pedestrians etc. This helps them to create drive route from one point to another. Using this points, the rules of certain situations can be evaluated and tested, whether an atonomous vehicle would be able to drive through this route or not.

Results

1. Pedestrian Detection using Image Processing and Machine Learning techniques

The performance of the algorithm varied for different resolution of videos. Finally, the resolution with 360 pixels was agreed on for the tested videos. The evaluation was done on Raspberry Pi 3b+ models and it provided with an average FPS of 4.7 along with accuracy of approximately 87%.

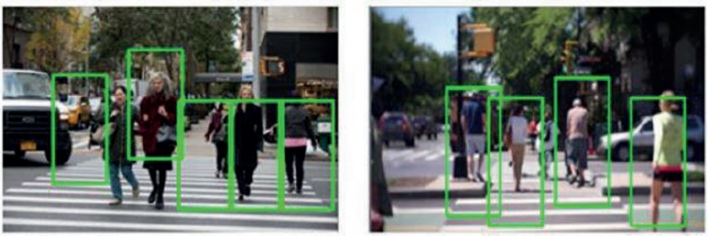


Figure 5: Pedestrian Detection

2. Traffic Light and Backlight Recognition using Deep Learning and Image Processing

The provided algorithms were able to detect both Traffic Light and Backlight of any vehicle but the performance of the combined algorithm has a significant impact on the low processing unit of Raspberry Pi 3b+. Individual algorithms had an average FPS of 5 but the combined algothm had an average FPS around 3, which can not be considered semi real time as well. A lot of optimisation along with upgrade of hardware would improve the performance.

Figure 6: Traffic Light and Backlight Detection

3. Optimization of Different Image Processing Algorithms

A handful of algorithms detection traffic signs, lights and German Highway specific signs were provided to this group and with the multi-threading approach, they were able to optimize the algorithm with a very high frame rate while maintaining the high detection accuracy. The combined algorithms had an average FPS of 7 and it was upgraded to an average of 13 fps for the same hardware platform.

Figure 7: Optimisation of Algorithms

4. Design of a GUI using python QT and Realization

A GUI was developed in order to generate sensor synthetic data on a time triggered basis. This was developed with Python QT and performed with only approximately 50% processing consumption of the Raspberry pi 3b+ model, which is a very good achievement.

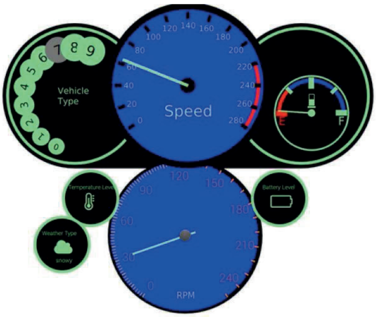


Figure 8: GUI for Sensory Synthetic Data

5. Design of GUI for Expert System Simulation and Rule Generation Tool

This group developed the GUI using Python and QT as well and this also consumed less than 50% of the total processing capabilities of Raspberry Pi. The positive aspects of this development is that the GUI is dynamic in approach, New rules can be added to modified along with new facts using the drag and drop options.

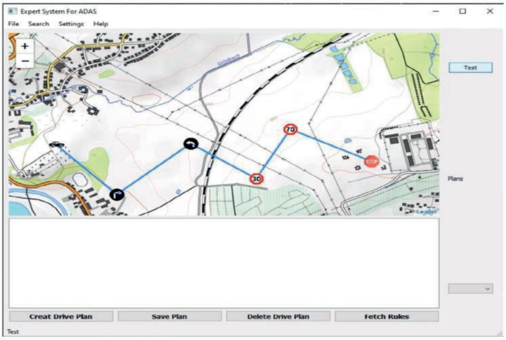


Figure 9: GUI for expert system simulation and evaluation

Conclusion

The purpose of this project was to create a prototypical system based on Vision Based approaches with the concept of situation awareness. Most of the concentration was provided to the perception of data and providing an understanding towards that data. For the perception purposes, image processing and machine learning based algotihms were developed, optimized and evaluated and for the understadning of the data, Rules based systems were created and simualted using the developed Graphical User Interfaces. Though the algoirthms had high accuracy in most of the cases, they were not even semi-real time, due to the lack of processing capabilities of the taget hardware of Raspberry Pi 3b+ models. For the future aspects, these shortcomings can be overcome with the improvment of the hardware units along with the implementaiton of more Optimized algorithms.

Digital Processes for Interactive Virtual Tutoring

Abstract

Virtual tutoring submerged with learning analytics in the learning system is beneficial to students as well as to the lecturer, course designer, curriculum planner at higher education institution. In one hand it helps facilitate students and on the other hand, supports the instructors in reducing their workloads. In this paper, six different tutoring technologies have been presented to incorporate the virtual tutoring system. Each of the technologies have been planned and implemented according to the scientific research methodology which starts with literature review and state of the art analysis and justifies the results with evaluation. The results curated by the student of this block: Digital Processes are able to present the primary version of each technology. This paves the way for future refinement and implication of the interactive virtual tutoring system.

Motivation

Tutoring has been recognized as the highly effective, purposeful and systematic way to support learning. It gives the learning opportunity to the learner to master the specific topic or problem. Rather than following didactical cycle, tutoring involves dialogue pattern consisting of initiating question, feedback on preliminary answer, scaffolding to improve the learning performance and finally evaluate learner's understanding of the topic. With the advancement of learning technologies, teaching learning process at the higher education institutes are now being conducted via Learning Management Systems (LMS). It gives the provision of tutoring the students using multimedia besides traditional lectures. This type of tutoring is a combination of identifying the tutee and intervening by computer based tutoring system [5]. This can be called virtual tutoring. In virtual tutoring, the learning deficiency of the learner is identified from learning analytics [1] and interventions are provided automatically by computer systems like Intelligent tutoring systems. The digital processes require to develop the virtual tutoring system are the topics provided as internship tasks for the second round of IKON project in this Digital Process block. The topics are as follows:

Topic 1: Descriptive Analytics from ONYX Test Data
Topic 2: Predictive Analytics from Log data of the OPAL Course
Topic 3: 3D Cloud Point Generation Using Survey Data
Topic 4: Interactive FAQs Generation and Visualization (Chatbot)
Topic 5: Interactive Tutoring Agent (Chatbot) Generation for Research Seminar
Topic 6: Design and Development of a GUI for a Tutoring Workbench

Topic 1 and 2 are directly related to database creation for learning analytics and topic 6 is related to the visualization of those analytics. An OPAL course created for the first round has been used for all necessary data as examples for artificial data generation following the given data structure. It has mainly 6 types of elements: Registration-Element, Upload-Folder-Element for the files from supervisors, 6 types of Calendar-Elements, two types of Blog-Elements, five types of Checklists-Elements and five types of ONYX Test-Elements. These ONYX test data has been used for topic 1 to be implemented. The Log data for topic 2 is acquired from all course elements and these different types of elements provide different types of learning experience data of the students.

For conducting learning analytics, these data are exported from the data archiving tool and exported function from the "Course evaluation" and "Statistics" functions of the course. The data are exported as XLS Files. In order to, represent the findings of learning analytics tasks in topic 1 & 2, topic 5 is given to design and develop a graphical user interface (GUI) for visualization. Alongside, topic 3 is set to develop the visualization strategy for multidimensional survey data using anonymized survey dataset.
Furthermore, two topics related to developing chatbots as interactive agent have been given with two different dialogue dataset for intervening as virtual tutors.

Student Background

There are 11 students registered to complete the tasks of this block. They are all Master students of Chemnitz University of Technology (TUC). Five of the intern students are studying “Master Automotive Software Engineering” at the faculty of Computer Science and another five students are studying in the program “Master Embedded Systems” and only one of the students is studying in the program “Master Information and Communication Systems” at the faculty of Faculty of Electrical Engineering and Information Technology.

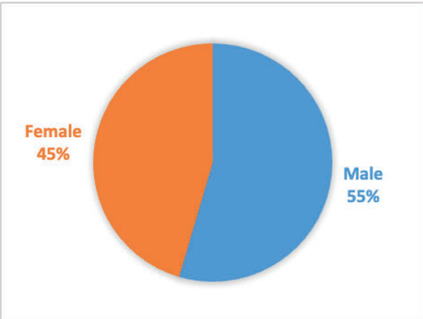


Figure 1: Gender ratio of the students

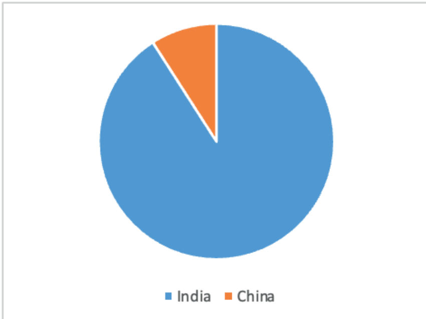


Figure 2: Nationality of the students

The gender ratio can be considered as balanced with 6 male students and 5 female students (Figure 1). The students are from an international study group and 10 of the students of this block are from India and only one student is from China (See Figure 2). All of the students are completing this internship in order to proceed with their thesis for the completion of the master studies. The following table shows the background of the students of Digital processes block.
The 11 students registered for this block have chosen the 6 aforementioned topics. For five of the following topics are implemented by formulating five groups of students and each groups contain two members.

- Descriptive Analytics from ONYX Test Data
- 3D Cloud Point Generation Using Survey Data
- Interactive FAQs Generation and Visualization (Chatbot)
- Interactive Tutoring Agent (Chatbot) Generation for Research Seminar
- Design and Development of a GUI for a Tutoring Workbench

So, 10 of the students have worked in groups. Only one student has worked on his own with the topic "Predictive Analytics from Log data of the OPAL Course".

Table 2: Students' Demographic Details

Student	Gender	Faculty	Field of studies	Current Semester
1	Female	Faculty of Computer Science	Master Automotive Software Engineering	6
2	Female	Faculty of Computer Science	Master Automotive Software Engineering	6
3	Female	Faculty of Computer Science	Master Automotive Software Engineering	4
4	Male	Faculty of Computer Science	Master Automotive Software Engineering	6
5	Male	Faculty of Computer Science	Master Automotive Software Engineering	4
6	Male	Faculty of Electrical Engineering and Information Technology	Master Embedded Systems	6
7.	Male	Faculty of Electrical Engineering and Information Technology	Master Embedded Systems	4
8.	Male	Faculty of Electrical Engineering and Information Technology	Master Embedded Systems	4
9.	Male	Faculty of Electrical Engineering and Information Technology	Master Information and Communication Systems	6
10.	Female	Faculty of Electrical Engineering and Information Technology	Master Embedded Systems	4
11.	Female	Faculty of Electrical Engineering and Information Technology	Master Embedded Systems	6

Introduction

Educational technologies have been evolved with the advancement of Web 4.0 technology and brought numerous changes in e-learning strategies by incorporating learning analytics and virtual tutoring for facilitating learner centered learning. Learning analytics is now being applied in the micro level learning settings [3] in the educational institutions for investigating learning potential and deficiency among the learners, learning systems and pedagogical strategies. It is the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs [1].

This process includes three steps: Firstly, data collection and pre-processing step that comprise data collection, data cleaning, data integration, data transformation, reduction, modeling, identification etc. [2].

Secondly, analytics and action which involve statistical analysis, visualization, monitoring, analysis, prediction, intervention, assessment, adaptation, personalization, recommendation, and reflection [1]. Thirdly, the post-processing step concludes with compiling new data from additional sources, refining the data set, determining new attributes required for the new iteration, identifying new indicators/metrics, modifying the variables of analysis, or choosing a new analytics method [1]. The learning setting informed with learning analytics can mediate virtual tutoring as intelligent tutoring system, as virtual peer tutor or a virtual avatar as tutor [4]. This mediation can support by providing instructions in the problem solving context [5], by providing immediate response to learner errors, by supporting successive approximations to competent performance and providing reminders of the learning goals.

In spite of theoretical advantages, learning analytics informed automatic or semiautomatic virtual tutoring system is novel in the field of learning technologies and thus require technical implementation for its evaluation in real life context. For this purpose, the aforementioned topics in motivation has been developed and their task description are stated below:

Descriptive Analytics from ONYX Test Data

The Learning Management System OPAL of TU Chemnitz offers the possibility to integrate online assessment tool ONYX into courses. Various short quizzes or the whole examination can be administered by ONYX. It has also the provision of automatic correction. The test set in ONYX has all its data in its repository. These tests' data can be extracted by the archive function from OPAL as Excel files. For learning analytics, it is necessary to automatically read this data and store it in a database. Therefore, a script or program has to be developed which reads all information of the exported data file, prepares it and stores it in a database. In the first step, the data must be understood and the information that can be extracted has to be identified. On this basis, the database design is to be defined and implemented. The data will then be automatically extracted and prepared for database storage. Finally, to generate descriptive analytics from the database a script needs to be programmed which is reusable for similar kind of dataset and can answer "what has happened?" in the dataset in the database.

Predictive Analytics from Log data of the OPAL Course

For learning analytics, it is necessary to automatically read the log data of the course that has the access time and usage frequency of different course elements and store it in a database. Therefore, a script or program has to be developed which reads all information of the exported data file, prepares it
and stores it in a database. In the first step, the data must be understood and the information that can be extracted has to be identified. On this basis, the database design needs to be defined and implemented. The data will then be automatically extracted and prepared for database storage and predictive analytics. After the data is automatically stored in the database, basic descriptive analytics of the provided dataset has to generated and predictive analytics using algorithm like Linear regression, model fit, Naive Bayes algorithm etc. needs to be performed to answer "what could happen?" in the given dataset and similar dataset.

3D Cloud Point Generation Using Survey Data

The survey data acquired from lectures for improving the learning quality needs to easily visible to the lecturers by using specific single visualization strategy rather than multiple bar diagram or charts from survey results. So, the effort of the lecturers and course designers can be minimized and the correction initiatives can be taken within short span of time. For this reason, the survey data having more than 3 variables and the correlation needs to be visualized. Therefore, a database design has to be developed for the given unsorted datasets and then by setting up the environment, the data will be loaded there. Afterwards, the visualization strategy needs to be chosen from the literature research and finally it needs to process the similar data, export and visualize. An example similar to this visualization is given here: https://vasturiano.github.io/3d-force-graph/example/large-graph/

Interactive FAQs Generation and Visualization (Chatbot)

For enabling virtual tutoring, a FAQ chatbot needs to be created. For this purpose, a database needs to be created for the dialog manager from the given data source. The dataset needs to be refined and then implemented in a server which should receive text message can generate corresponding response. Finally, the chatbot needs to be globally deployed as pop up window of a website.

Interactive Tutoring Agent (Chatbot) Generation for Research Seminar

As a part of virtual tutoring system, a tutoring chatbot needs to be created. For this purpose, a database needs to be created for the dialog manager from the given material which will act as an instructional tutor for writing a scientific research report. The dataset needs to be created, refined and then implemented in a server which should receive text message can generate corresponding response. Finally, the chatbot needs to be globally deployed as a pop up window of a website.

Design and Development of a GUI for a Tutoring Workbench

The data exported from task 1 and 2 stored in the database are to be visualized in this task for tutoring workbench which will have two interfaces. One for lecturer and another for individual students.Therefore, a database design has to be developed together with the colleagues of those tasks. Afterwards the data should be presented as web components on a website. For this purpose, statistical analysis has to be conducted to visualize the temporal evolution (e.g. by semester) and the current state of the semester.

Methodology for Task Completion

The basic steps for executing all the tasks include the steps in the following figure. The task implementation has been initialized with literature research related to the task. With the findings of the literature research the solution concept has been developed and the implementation has been carried out.

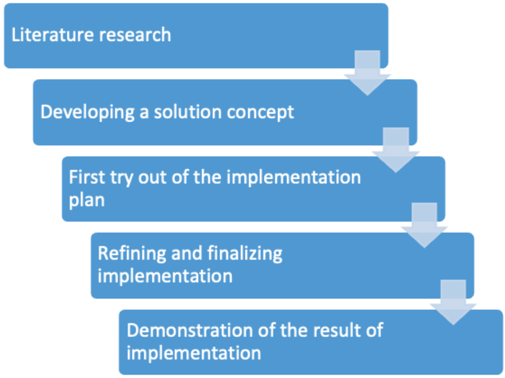


Figure 3: Basic working steps for task completion

The detail working strategy involves for all the topics is to recognize the data and filter the information first. Specifically, for topic 1,2, 3 & 6 the data has to be automatically extracted and prepared for database storage and statistical evaluation. The exported data has to be visualized in the sixth task using the data of the database from topic 1 & 2 for the prototyping process. Subsequently, the data has to be presented as components of a website. This is similar task implementation step for topic 3 with survey dataset using 3d cloud point visualization strategy.

Task four and five required dialogue dataset and for the FAQ agent it has been partly provided and for topic five the material to develop the dataset have been given. The group of students have to develop their own dataset from the provided material. Afterwards, with the application of state of the art framework the chatbots needs to be programmed and embedded in a website.

Task Implementation and Results

Descriptive Analytics from ONYX Test Data

The approach that was followed in implementing the task from the provided data mainly comprises of 4 steps. They are namely,

1. Data Extraction
2. Data Transformation
3. Loading into the Database and
4. Applying Descriptive Analytics

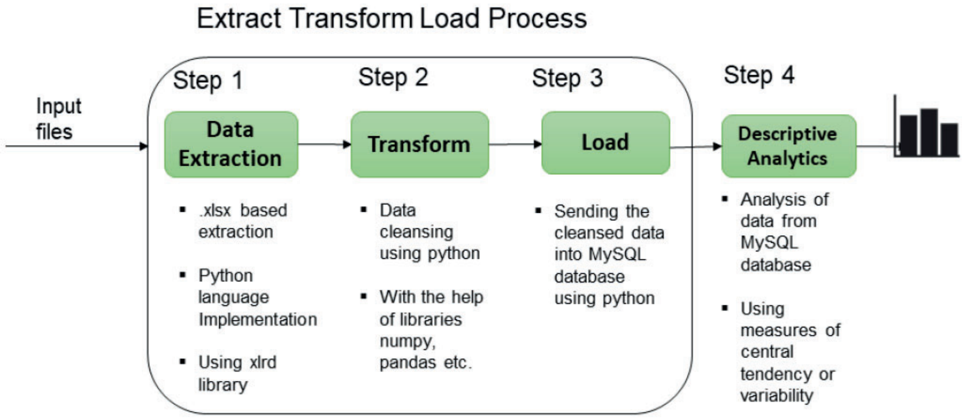


Figure 4: Implementation steps of Descriptive Analytics

The data mining process as shown in the figure involves the steps data extraction, transformation of uncleansed datasets into cleansed datasets, loading the cleansed data into required stored location and then finally making the data available for the descriptive analytics to get useful insights. The first three steps extract, transform and load are called ETL. The aim of the ETL process is to load the database with the integrated and cleaned data. The data frame object cleansed need to be stored in a database. MySQL database of TU Chemnitz is used because it is available for free and is open source.The SQLAlchemy library is responsible for the communication between the python program and database. The data from the database can be extracted using mysql queries. The database consists of total nine tables. Each table consists of the data regarding particular test responses with each question points, all the test reponses, student details, question format details etc.

Table 2: Tables in MySQL Database

SNo.	Table Name	Table Description	No. of Columns
1	Student Data	To store the student personal data	6
2	Student Response	To store the student responses of all the tests	22
3	ARS Test Table	Contains the test data of all the tests in ARS Module	10
4	Self Test Table	Contains the test data of all the tests in Self Module	10
5	Reference Table	Contains data about various question types in ONYX test Suite	3
6	Discussion Response	Table contains the responses of discussion test	24
7	Presentation Response	Table contains the responses of presentation test	60
8	Report Response	Table contains the responses of report test	16
9	Search Response	Table contains the responses of search test	30

Now the data from the MySQL database can be extracted with the mysql queries and are stored in a data frame. The extracted data is used to make some useful insights in the form of bar graphs,pie diagrams and line diagrams.

With the use of python library matlplotlib which is used to produce 2D visualizations so that it aids in the crucial step of gaining a basic understanding of data relationships. For performing Descriptive Analytics, mathematical measures such as "Measures of Central Tendency" and "Measures of Variability" were used. After applying these mathematical measures, the outcomes were visualized in the form of charts or graphs. These help the end user to easily understand the hidden patterns or trends present in the input data.

Predictive Analytics from Log data of the OPAL Course

At first Python is used for the process of data extraction and database design. Supervised learning algorithms such as regression and classification models to make prediction from the given dataset and then compared for best possible outcomes. The implemented steps are as follows:

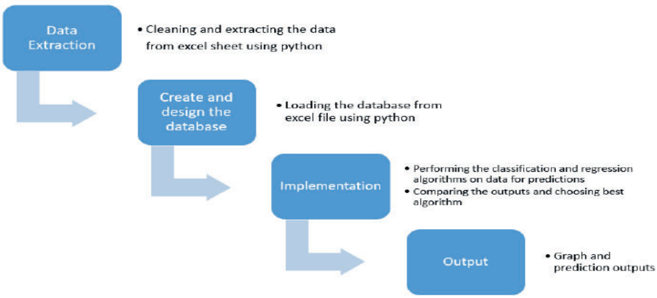


Figure 6: Implementation steps to generate Predictive Learning Analytics

The methodology that is used can be divided into three stages: the first stage is the evaluation of frequencies for each node. This not only limits the potential of the nodes and its highlights to various categories. The second model is the classification model, given the time of the day and the other variables in the model, would predict which node would be most likely opened in the mix.

Additionally, it can be observed that there is another model, the regression model, and this model evaluates on the given set of conditions, the number of views that the node would receive at any given circumstance.

The classification yielded the following results indicating the future importance of the variables.

```
Test course for IKON Internship                               55
ONYX Test 2                                                   47
ONYX test 1                                                   46
OPAL Check-List Data Extraction 1                             29
ONYX Test 3                                                   26
Book your online test schedule                                26
ONYX test 5                                                   25
Blog element data extraction 1                                24
Mark the dates for your Group meeting for project report      22
Book the dates for your holidays                              21
Mark the dates to schedule your weekly seminars               21
ONYX test 4                                                   19
Mark your appointments for the presentations                  18
iCAL Calendar Data extraction                                 18
Blog element data extraction 2                                16
Mark the dates to attend the virtual lectures                 14
OPAL Check-List Data Extraction3                              14
Registration                                                  12
OPAL Check-List Data Extraction4                              11
OPAL Check-List Data Extraction 2                              8
OPAL Check-List Data Extraction 5                              6
Name: predict_class, dtype: int64
```

Figure 7: Predicted number of views using classification model

It can be said that the tests of all kinds are forecasted to have more visits or views, as compared to the rest of the nodes.

The regression models do predict the number of views, and it can be seen that the number of views predicted is as follows

```
Blog element data extraction 1                              21.0
Blog element data extraction 2                              20.0
Book the dates for your holidays                            21.0
Book your online test schedule                              26.0
Mark the dates for your Group meeting for project report    22.0
Mark the dates to attend the virtual lectures               14.0
Mark the dates to schedule your weekly seminars             21.0
Mark your appointments for the presentations                18.0
ONYX Test 2                                                 37.0
ONYX Test 3                                                 31.0
ONYX test 1                                                 41.0
ONYX test 4                                                 30.0
ONYX test 5                                                 28.0
OPAL Check-List Data Extraction 1                           17.0
OPAL Check-List Data Extraction 2                           15.0
OPAL Check-List Data Extraction 5                           15.0
OPAL Check-List Data Extraction3                            14.0
OPAL Check-List Data Extraction4                            11.0
Registration                                                12.0
Test course for IKON Internship                             55.0
iCAL Calendar Data extraction                               18.0
Name: predict_views, dtype: float64
```

Figure 8: Predicted number of views using a regression model

This also indicates that the test figures are higher in views than the rest of the variables. Supervised learning algorithms such as regression and classification models to make prediction from the given dataset and then compared for best possible outcomes. After comparing the both models, the regression model provides the best accuracies and the random forest regressor has high accuracy.

Further, any new data/observations can directly feed to this model to make possible predictions to know what the learners are most like to use from a course.

Cloud Point Generation Using Survey Data

As the amount of data processed increases daily, it is critical to have all the relevant information in the form of a data visualisation strategy that enables people to readily grasp the link between various sets of data. This brings us to the use of network graphs. They are mostly used to uncover relationships between data that would otherwise go unnoticed if viewed on a spreadsheet.

The implementation of building a Networkx graph to define relationships is done using python and its inbuilt libraries. It is done by importing all the required libraries and the CSV file containing the dataset from google drive by mounting it. Then the process Feature Engineering is conducted. Feature engineering is the process of using domain knowledge to extract features (characteristics, properties, attributes) from raw data. Studying a small part of the dataset using head() and finding the unique values for every column using unique() is done here. After that, the Participant column is set as the index so that it makes the dictionary creation process easier. This change is made in place, i.e, in the same dataset. Following that, creating dictionaries for key: value pair, for eg-participant: gender, participant: nationality etc. For Questions Dictionary, store the same dataset in two new variables, i.e, for each question and take only the required columns in it, i.e, question, answer, and participant, and set index to Questions.

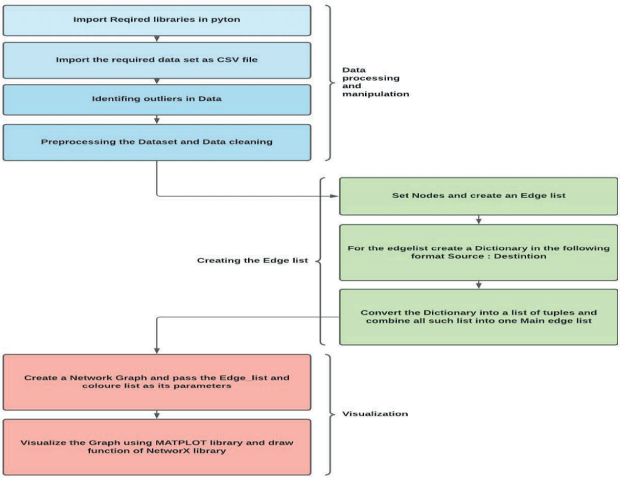


Figure 9: Implementation of cloud point generation of survey data

Then questions data are converted to dictionaries and once dictionaries are created, they are converted into a list of tuples. All these lists of tuples are combined to form one large list / main list which is done using extend the function of python and this list acts as the edge list for the networkx graph.

The multi-line adjacency list format is useful for graphs with nodes that can be meaningfully represented as strings. With the edge list format, simple edge data can be stored but node or graph data is not. There is no way of representing isolated nodes unless the node has a self-loop edge. Finally, by drawing the networkX graph using graphlist as edge list and assigning different colors to different values the final visualization is generated.

We can see from the result that all the required relationships are defined and it is observed that spring layout(default) is the best fit for our dataset and hence it was taken as the final diagram and colors for nodes were changed using conditional statements (if-else).

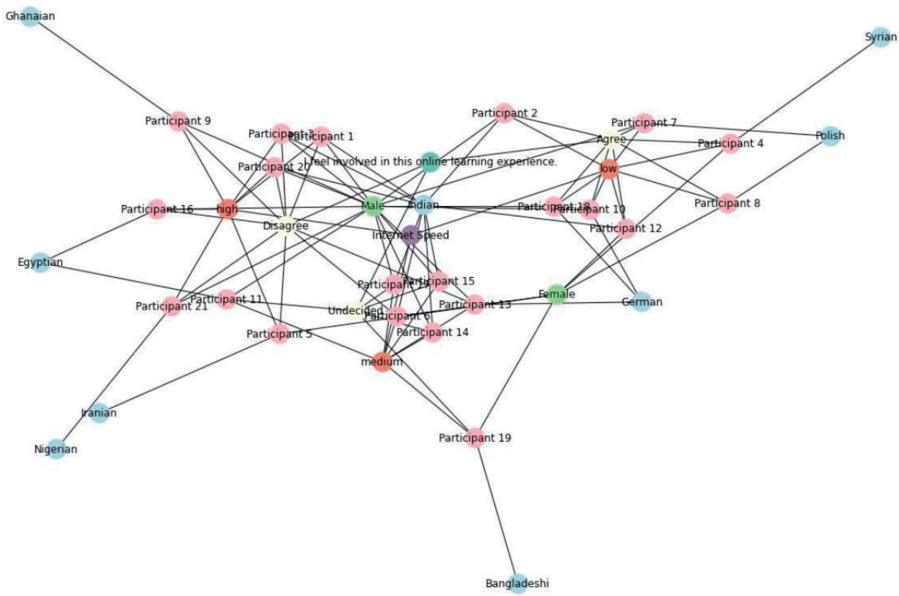


Figure 10: Network Diagram of Dataset with colored nodes

We can see from the result that all the required relationships are defined accordingly in the form of edges and nodes with similar properties or categories and have the same color.
Here the plotted graphs are for 20 participants, when the number of entities become more or less the size and layout techniques may differ. In the future, 3D implementation of the graph can be done. Implementing it with the other project is a must to make use of it. At the current stage, the system can automatically add the entities ("rows") but is not able to add categories ("columns"), which can be done to make the system fully automated. An AR barcode can also be created by using AR.js of the 3D graph for better visualization. A use of forces in graphs makes it more valuable to extract relevant information.

Interactive FAQs Generation and Visualization (Chatbot)

The chatbot is an Artificial Intelligence (AI) technology for natural language communication with humans, the most accessible solution for all users.It enables human language to be understood by Natural Language Processing (NLP). It is a branch of artificial intelligence that deals with natural language interaction between computers and humans. NLP's main goals are to read, decipher, understand, and make sense of human language in a useful way. The interactive FAQs generation, multiple datasets (Frequently asked questions) have been given which cover major topics such as presentation, seminar, and report details. In that given topic we have divided more subtopics (the possibility of topics which students might seek help for) and expanded the datasets to help students to get all kinds of information regarding the presentation, seminar, and report. We have created 140 samples of natural language data (the possible way of asking questions) with 32 distinct intents. For the rasa installation, Ujson and TensorFlow packages need to be installed inside the project directory. In nlu.yml, the intents and utterances are saved. In stories.yml, we will find conversations' stories. Short segments of discussion that should always follow the same direction without the interference of ML policy but with rule policy are referred to as rules which are stored in rules.yml. In config.yml, we have the pipeline and policies. The domain.yml file specifies the chatbot's domain. Intents, entities, slots, utterance responses, actions, and dialogue session settings are all specified in the domain. A dialogue session's duration is set to 60 seconds.

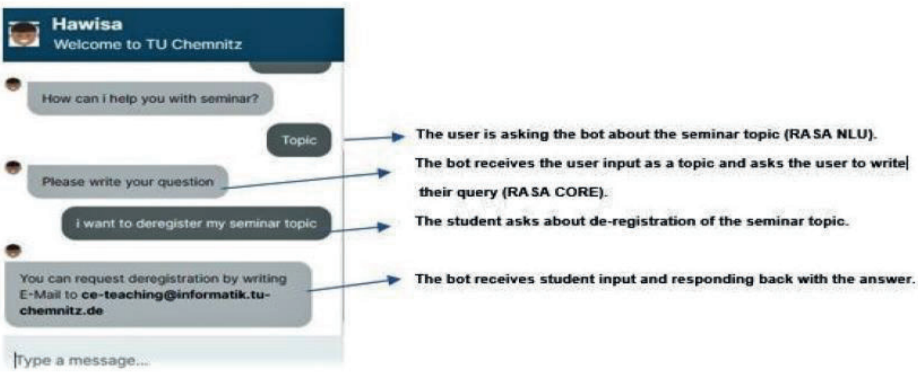


Figure 11: Demonstration of the generated FAQ bot

To develop a user chat interface for the chatbot, we have used Chatroom, a React-based component for the Rasa stack. It is a widget that we incorporated into the custom webpage that we developed using an Html snippet. Custom integrations are done through the Rest Input channel. They give a URL where we can send messages and asynchronously get responses via a webhook. We will receive a REST endpoint via the REST channel. This is stored in credentials.yml and the Socket IO channel uses WebSocket's and is real-time. To use the Socket IO channel, we have added the credentials to our credentials.yml. The chatbot's performance accuracy is judged by the precision of creating sentences, which can be properly read in response to the user. Moreover, the findings of the chatbot model evaluation reveal fairly good results close to the perfect precision, reminder, and F1 score as mentioned in the section result and analysis. In the future, by maintaining a university database we can handle multiple student requests at a time and obtain an insight into the bot to understand to what extent the Machine learning-based chatbot can respond to the students by understanding the context properly.

Interactive Tutoring Agent (Chatbot) Generation for Research Seminar

The Tutoring AI agent not only guide the students, but also help them to get a feedback on their presentation slides and report structure. The Tutoring AI agent is basically implemented by using Rasa framework. The model is trained on a customized data set (information/feedback from students) and it runs locally on a web page.

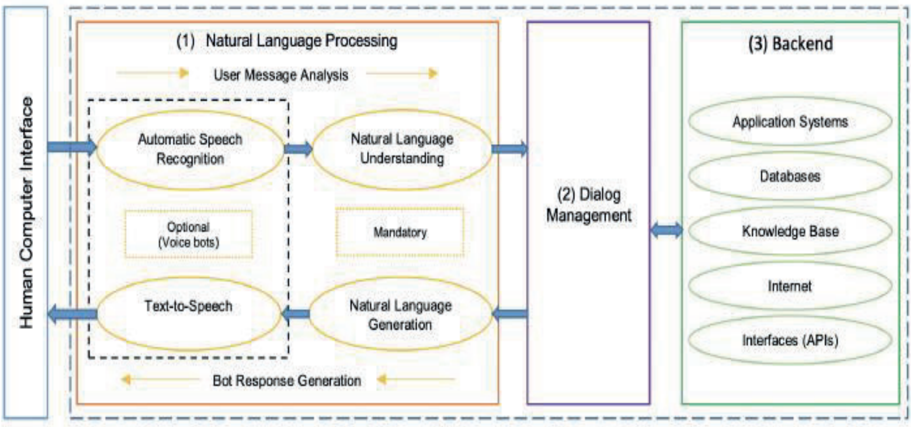


Figure 12: General Structure of the Chatbot [6]

For the implementation, three modules have been used. These are- NLU data: Intents are used as training data for assistants Natural Language Understanding model. This data is used to store structured information about user messages. It contains the user's intent and entities; Responses: Chatbot understands the user's message and it needs a response to the user. It includes responses for corresponding intents. Stories specifies the action that chatbot should execute; Stories: These are the example conversation that is used to train a chatbot to respond correctly depending on what user query. Story format shows the intent of user messages followed by chatbot response or action and this helps the chatbot to retrieve relevant messages back to users.

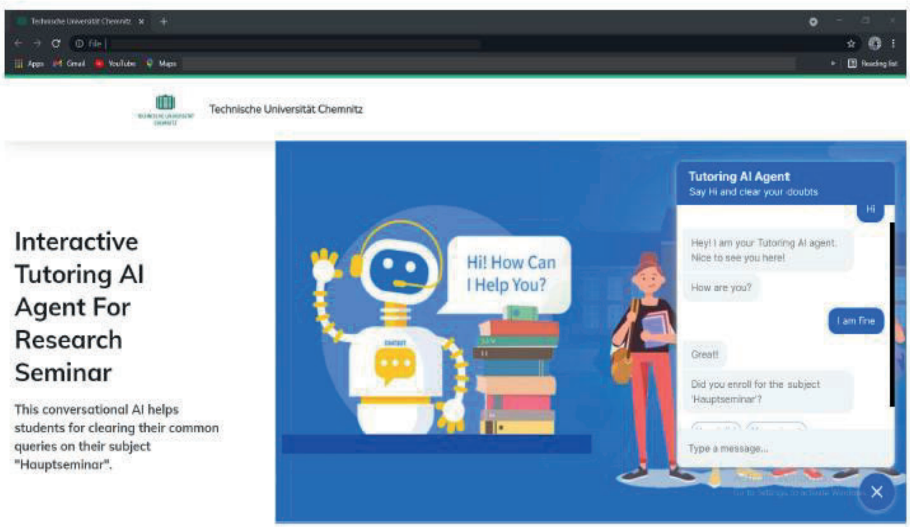


Figure 13: Tutoring AI agent Web Interface

The current model has been thoroughly trained and tested within the Rasa environment. The _nal model contains zero outliers as a result of internal testing. The agent is currently running on a local server and has already been examined by limited users. The data set has been refined several times to improve the model interaction. The model was improved again after each round of user testing. The final model of the Tutoring AI agent responds to the queries quickly and efficiently.
The limitations of this project are the proposed model runs only on the local server which is deployed in a web page and the model is not able to save the conversation history of different users.

Design and Development of a GUI for a Tutoring Workbench

Since the tutoring workbench involves both the student and the tutor, two different views have to be created. The first is the Tutor's view where the tutor should be able to view the overall statistics of the students who are participating in the tutor's class as well as every student's individual performance. Finally, the student's view where the student should be able to view his/her performance in the course.
General workaround for this GUI is to access the database using one of the server-side languages. For example: PHP. Then create a new CSV file and store the data from the database table into this file. CSV files are used instead of normal spreadsheet files as CSV files with the same content would occupy at least 100 times smaller memory than other spreadsheet files used by software like Microsoft excel, Apache Open office and so on.

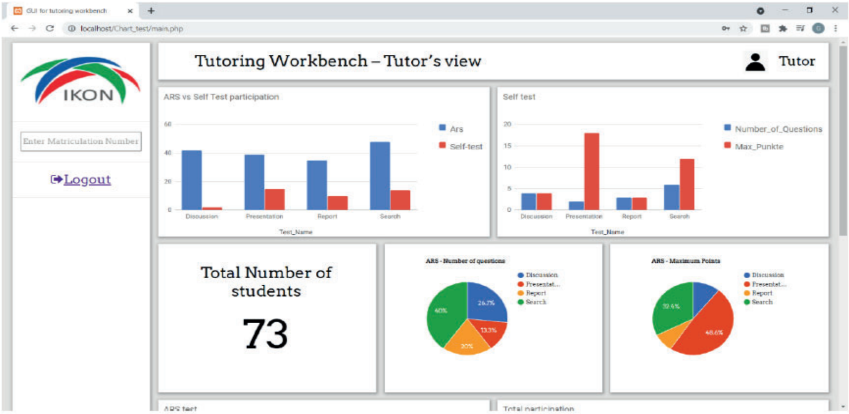


Figure 14: GUI of the Tutoring workbench (Tutor's View)

The obtained CSV file will have huge chunk of data as it will hold the entire database table in it. In order to remove only the needed data, we have to convert the existing CSV file into text file as programming languages can handle text files better than the other files. The conversion of file is done through python and the needed data is filtered out of the text file. Then a new CSV file is created, again through python to store the data. Python is chosen for this as file handling will be bit easier than in traditional object-oriented languages like C++, PHP and so on. The CSV file is then given to the frontend stack which is done through HTML, CSS and JavaScript. HTML provides the basic skeleton for the webpage which will then be styled with CSS. JavaScript is used to integrate the chart library (in our case the google charts) which will take the input as a CSV file and then display the data graphically. Google charts were chosen for this purpose as it is easy to integrate the basic charts into a web page or a web app using Google charts and has a huge community forum support for the users who face problem with the integration.

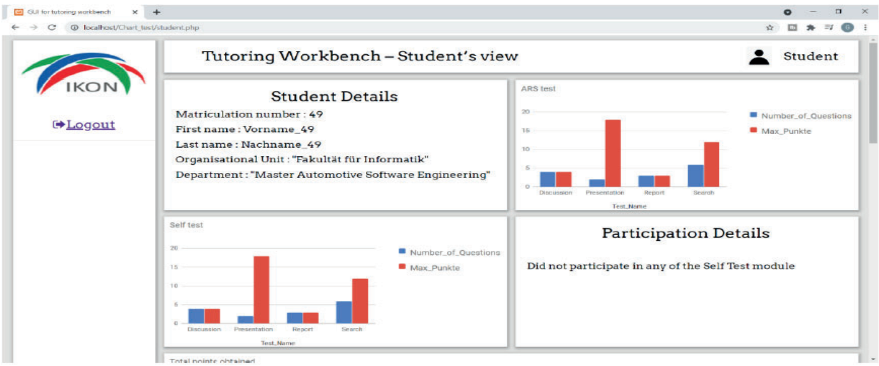


Figure 15: GUI of the Tutoring workbench (Student's View)

Currently there are eight visualizations graphs in tutor's view namely-

- ARS vs Self-test participation
- Self-test's number of questions and maximum points
- Total number of students
- ARS's number of questions percentage calculation
- ARS's maximum points percentage calculation
- ARS's number of questions and maximum points
- Total participation
- Predicted views

There are many more visualizations that can derived from the databases in addition to the above eight visualizations. The same is applicable to student's view where additional visualizations can be derived and the same can be implemented.

Conclusion

Virtual tutoring initiatives requires multiple technologies to be implemented in order to reduce the workload of traditional tutoring. It includes detection of learning problem of the group of students or individual student along with taking solution measures to solve those problems for the study success. In the second round of IKON project, a primary motivation has been taken to build the prototype of these technologies. The obstacles and loopholes have been identified in the way of developing and implementing this protypes of learning analytics system, graphical user interfaces for various data visualization and direct tutoring intervention as virtual agent (Chatbot). Above all, each of the finding of the separate task has been successfully implemented as first version of prototype . It indicates that these technologies are realizable in the learning context. Further development can be possible with the needed refinement and after the evaluation of the virtual tutoring system,it has the scope to be implemented in the respective digital learning environment.

References

[1] G.Siemens, "Learning analytics: The emergence of a discipline", *American Behavioral Scientist*, vol. 57(10), pp. 1380-4000, 2013.
[2] T. Elias,"Learning Analytics: The Definitions, the Processes, and the Potential [EB/OL]",2011.
[3] S. Park, and C. Kim,"Boosting learning-by-teaching in virtual tutoring", *Computers & Education*, *82*, pp.129-140, 2015.
[4] M. E. Osman,"Virtual tutoring: An online environment for scaffolding students' metacognitive problem solving expertise", *Journal of Turkish Science Education*, *7*(4), pp.3-12, 2010.
[5] José Joaquim Moreira, António Lencastre Godinho, Ana Paula Pinto, Carlos Miguel Oliveira and Manuel Pinto Teixeira, "Web application for support tutoring system" presented at the 12th Iberian Conference on Information Systems and Technologies (CISTI), 2017.
[6] M. Biswas, "Beginning AI Bot Frameworks: getting started with bot development", Springer,2018.
[7] A. Wong, and M. Lavrencic, H. Khosravi, "An overview of learning analytics in higher education",Institute for Teaching and Learning Innovation, 2017.