Infocommunications Journal

A PUBLICATION OF THE SCIENTIFIC ASSOCIATION FOR INFOCOMMUNICATIONS (HTE)

ISSN 2061-2079

Special Issue

Special Issue on AI Transformation

Technically Co-Sponsored by

IEEE ComSoc™ IEEE Communications Society

hte

IEEE HUNGARY SECTION

Indexing information

Infocommunications Journal is covered by Inspec, Compendex and Scopus.
Infocommunications Journal is also included in the Thomson Reuters – Web of ScienceTM Core Collection, Emerging Sources Citation Index (ESCI)

Infocommunications Journal

Technically co-sponsored by IEEE Communications Society and IEEE Hungary Section

Supporters

FERENC VÁGUJHELYI – president, Scientific Association for Infocommunications (HTE)

The publication was produced with the support of the Hungarian Academy of Sciences and the NMHH

National Media and Infocommunications Authority • Hungary

Editorial Office (Subscription and Advertisements)**:**
Scientific Association for Infocommunications
H-1051 Budapest, Bajcsy-Zsilinszky str. 12, Room: 502
Phone: +36 1 353 1027 • E-mail: info@hte.hu • Web: www.hte.hu

Articles can be sent also to the following address:
Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
Phone: +36 1 463 4189 • E-mail: pvarga@tmit.bme.hu

Subscription rates for foreign subscribers: 4 issues 13.700 HUF + postage

Publisher: PÉTER NAGY

HU ISSN 2061-2079 • Layout: PLAZMA DS • Printed by: FOM Media

Guest Editors, Authors and Co-authors

Special Issue

Special Issue on AI Transformation

Guest Editors
Péter Baranyi, Viktor Dörfler

Authors
Krisztián Mihály, Gyula Kulcsár, Mónika Kulcsárné-Forrai, Osama A. Salman, Gábor Hosszú, Jinat Ara, Cecilia Sik-Lányi, Erika Baksáné Varga, Attila Baksa, Anita Agárdi, Károly Nehéz, Melinda Magyar, Dávid Burka, Olivér Hornyák

An Advanced Reactive Approach to Solve Extended Resource-Constrained Project Scheduling Problems

Krisztián Mihály, Gyula Kulcsár, and Mónika Kulcsárné-Forrai

***Abstract*—This paper introduces a fast and configurable method for solving resource-constrained multi-project scheduling problems, using a multi-aspect decision-making procedure that combines a schedule generation scheme with various task-selection values and priorities. The goal of fast scheduling generation is to support reactive scheduling environments. During calculation each decision aspect is computed to produce numerical values, reflecting the importance of each aspect for candidate selection. These priorities can be tailored to specific optimization objectives. The priorities can be customized according to the objective of the optimization problem. The method was tested on the PSPLIB RCPSP J30 benchmark series to minimize project completion time using eight decision aspects. The average relative deviation from lower bounds was used to evaluate the impact of different decision aspect priorities. Although the focus was not on determining optimal priority values, the study explores the effectiveness of using multiple priority rules simultaneously in a configurable way in reactive scheduling environment. Performance tests confirm that the proposed method is flexible, robust, fast, and effective in solving the examined problem type.**

***Index Terms*—Scheduling, Reactive Control, Priority Rule, Multi-Aspect Decision Making, Project Management**

I. Introduction

One of the core challenges in project management is creating an effective schedule that ensures the timely and efficient completion of projects. The Resource-Constrained Project Scheduling Problem (RCPSP) is a fundamental issue that impacts various industries, including manufacturing, software development, construction, logistics, and research and development. The primary goal of RCPSP is to allocate limited resources over time to a set of interdependent activities or tasks, optimizing performance indicators such as minimizing project duration, reducing lateness, minimizing costs, or maximizing resource utilization.

Our research focuses on the efficient solution of reactive scheduling and control problems in dynamically changing execution environments with numerous tasks. This paper presents a method suitable for reactive scheduling that simultaneously considers multiple decision aspects without iterative attempts.

Submitted on 01/06/2024

Krisztián Mihály, Gyula Kulcsár, and Mónika Kulcsárné-Forrai, Department of Information Engineering, University of Miskolc, Miskolc, Hungary (E-mail: krisztian.mihaly@uni-miskolc.hu – correspondence author, gyula.kulcsar@uni-miskolc.hu, monika.kulcsarne@uni-miskolc.hu)

In this paper, we first provide a literature review in Chapter II of the Resource-Constrained Project Scheduling Problem (RCPSP). Chapter III presents the base problem extension to multi-project scheduling. Chapter IV discusses generation schemes and their applicability as simulation module for reactive scheduling. Chapter V introduces a multi-aspect qualification method as an enhancement option to the generation schemes. Chapter VI presents the experimental results by implementing the method on the PSPLIB benchmark set. Chapter VII concludes with a summary of the results.

II. Literature Review

The definition of the RCPSP problem was introduced in 1969 [1] and was mathematically proven by Blazewicz et al. to be strongly NP-hard [2]. Several survey papers on RCPSP have been published [3]-[5]. Although the original RCPSP model is well-known and sufficient for many cases, practical applications require further extensions. An updated overview of these extensions is provided by Hartmann and Briskorn [6], who categorize model variants based on generalization of activities, alternative precedence constraints, network characteristics, and consideration of multiple projects.

One common approach to solving RCPSP involves using task selection priority rules with schedule generation schemes (SGS). These priority rules determine the task scheduling sequence, impacting the overall schedule efficiency. Priority rules are heuristic methods that order tasks based on specific criteria. Early examples include the earliest start time (EST), earliest finish time (EFT) [7], minimum slack time (MST) [8], and shortest processing time (SPT) [9]. Recent research explores the use of genetic-like evolution of task priority rules [10] and the automatic detection of the best applicable rules for RCPSP problems [11].

The literature presents approaches for handling multiple objective functions simultaneously [12]. Problems with more than three objectives are referred to as many-objective optimization problems [13], presenting new challenges, such as comparing candidate solutions using suitable performance metrics [14]. To address these challenges, researchers develop various methods based on existing approaches [15], advanced methods [16] and hybrid approaches [17].

DOI: 10.36244/ICJ.2025.5.1

III. Multi-Project Scheduling

The resource-constrained multi-project scheduling problem involves a set of activities to be executed on a set of resources, collectively forming projects. Each project is represented by an acyclic directed graph following an activity-on-node model. Nodes represent activities, which must be executed without interruption. Precedence relations, shown as arcs between nodes, indicate that a successor activity cannot start until all predecessors are completed.

The time horizon is divided into elementary time units (e.g., seconds, days, months), chosen based on the project-execution environment. Each activity's processing time is given as multiples of these units. Some activities may belong to multiple projects, creating interdependencies. Projects may have unique release times and due dates, and activities cannot start before their release times. Projects may differ in priority and have various scheduling goals modeled as objective functions, which project management uses for concurrent scheduling.

The execution system has a set of renewable resource types available for project activities. Each resource type has a time-dependent capacity constraint that specifies the available quantity in each time unit. These resources are not consumed but used by activities, then released upon completion, making them available again.

Each activity has specific resource requirements defining the type, quantity, and processing time needed. An activity can start only if the required resources are available for the necessary duration. Resource use begins simultaneously but may end at different times. Activities are non-interruptible, as pre-emption is not allowed. Each project includes virtual start and end activities, which require no resources and have zero processing time.

The investigated scheduling problems may involve multiple objectives with varying values, optimization directions, and importance levels. The objective functions can differ. To solve this extended problem type, a detailed schedule must be created specifying the exact start time for each activity. Our goal is to rapidly generate a feasible, near-optimal schedule that considers the objectives and meets all constraints.

This extended scheduling problem is referred to as ESP in this paper. In describing ESP, we draw on classical project scheduling concepts to establish its relationship to known models and highlight its unique features.

ESP includes RCPSP as a special case, making it also NP-hard. Additionally, ESP encompasses other classical scheduling problems like Single Machine Scheduling, Flow Shop Scheduling, and Job Shop Scheduling. While the literature often uses "operation" instead of "activity" in machine scheduling, and "job" to denote a set of operations, we use "task" to denote the elementary process and "project" to refer to a set of related tasks. This paper avoids the ambiguous term "job".

Our research objective was to develop a solution approach for making real-time decisions for ESP, especially in environments burdened with uncertainty and frequent unexpected events. Such environments require continuous adjustments and rapid scheduling decisions, such as in cyber-physical production systems or agent-based logistical systems. Considering these factors, we chose the reactive scheduling strategy as our fundamental approach.

IV. Generation Scheme as a Suitable Base for Reactive Scheduling

The Schedule Generation Scheme (SGS) is a well-known type of predefined, rule-based, constructive methods. Starting with an empty schedule, SGS iteratively adds one unscheduled task to the partial schedule until all tasks are scheduled. Table I presents the applied notations of variables used in the algorithm.

TABLE I
Notation for Generation Scheme

Notation	Description
$\boldsymbol{T}$	Set of tasks to be scheduled
m	Iteration of the generation scheme
$\boldsymbol{TGEN_m}$	$\boldsymbol{TGEN}_m \subseteq \boldsymbol{T}$; the already scheduled tasks in iteration m
$\boldsymbol{D_m}$	$\boldsymbol{D}_m \subseteq \boldsymbol{T}$; the decision set of tasks in iteration m
$SGEN_m$	$SGEN_m = (S_1, S_2, \dots, S_j, \dots, S_{NT})$, the starting time vector of already scheduled tasks in iteration m
$CGEN_m$	$CGEN_m = (C_1, C_2, \dots, C_j, \dots, C_{NT})$, the completion time vector of already scheduled tasks in iteration m

Algorithm 1. presents the pseudo code of generation schemes in general.

Algorithm 1: Generation Scheme Algorithm (SGS)

Input: ESP problem definition, task selection rules and priority values
Output: Feasible schedule
Begin
2.1 Create an empty schedule ;
2.2 m = 0; $\boldsymbol{TGEN}_0 = \emptyset$,$SGEN_0 = (0,0,\dots,0)$; $CGEN_0 = (0,0,\dots,0)$;
2.3 **while** ($\boldsymbol{TGEN}_m <> \boldsymbol{T}$)
2.4 m = m + 1;
2.5 Calculate $\boldsymbol{D_m}$
2.6 Select one task from $\boldsymbol{D_m}$
2.7 $\boldsymbol{TGEN}_m = \boldsymbol{TGEN}_m \cap \{t_m\}$
2.8 Calculate $SGEN_m$, $CGEN_m$
2.9 **end while**
2.10 Return schedule $SGEN_m$
End

The calculation of $\boldsymbol{D_m}$ depends on the generation scheme variant. This paper uses the serial generation scheme variant, where a task is included in the $\boldsymbol{D_m}$ decision set in the m.-th iteration if and only if the task has not been scheduled and all its predecessor tasks have been scheduled.

$$\boldsymbol{D_m} = \left(\{ t_j \} | t_j \notin \mathbf{TGEN_m} \wedge \forall t_{pre} \in \boldsymbol{PRE_j} \wedge t_{pre} \in \mathbf{TGEN_m}\right) \quad (1)$$

V. A Multi-Aspect Qualification Method to Select the Most Appropriate Task from the Decision Set

In our reactive scheduling model, multiple task-selection decision aspects (TSDA) can be used simultaneously. We assume the set of applicable TSDAs is not limited and can encompass various items with different priorities and optimization directions. Optimization direction indicates

whether a larger or smaller numerical value is desirable for the candidate task. In each execution of the solver, an actual system of applied TSDAs is given, and the actual value of each TSDA can be calculated.

Let s_x and s_y be two candidate tasks to be selected for adding to the partial schedule at iteration m. The calculated values of TSDAs are represented by a given vector containing K real numbers. The notations are given in Table II.

TABLE II
NOTATION FOR RELATIVE QUALIFICATION

Notation	Description
u	$u = (u_1, u_2, \dots, u_k, \dots, u_K), u_k \in \mathbb{R}$; u denotes the vector containing the values of TSDAs considering the given task to be compared.
z	$z = (z_1, z_2, \dots, z_k, \dots, z_K), z_k \in \{-1, 1\}$; z denotes the vector containing the optimization directions of TSDAs. The value of z_k is 1 if the smaller value of the k^{th} TSDA indicates the more favorable task. The z_k is -1 if the larger value pf the k^{th} TSDA indicates the more favorable task.
w	$w = (w_1, w_2, \dots, w_k, \dots, w_K), w_k \in \mathbb{Z}_0^+$; w denotes the vector containing priorities for the TSDAs. Each w_k is a non-negative real value ($w_k \geq 0$) that expresses the importance of the u_k value of the k^{th} TSDA.

A distance function D is defined as follows.

$$D : \mathbb{R}^2 \rightarrow \mathbb{R},\ D(a, b) := \begin{cases} 0, & if\ \max(|a|, |b|) = 0, \\ \frac{b-a}{\max(|a|,|b|)}, & otherwise \end{cases} \quad (2)$$

The relative qualification uses notations defined in Table II. Let x and y be two vectors with type u. These vectors contain the values of TSDAs, and they represent the absolute qualities of candidate tasks t_x and t_y to be compared. We define the F function to express the relative quality of y compared to x as a real number.

$$F : u^2 \rightarrow \mathbb{R},\ F(x, y) := \sum_{k=1}^{K} (w_k \cdot z_k \cdot D(x_k, y_k)) \quad (3)$$

Using the return value of F(x, y), we can expresses the relative quality of vector y compared to vector x as the following:

- y is better than x if $F(x, y)$ is less than zero.
- y is worse than x if $F(x, y)$ is larger than zero.
- y and x are equally good if $F(x, y)$ is exactly zero.

The presented F-based relative qualification model effectively solves the comparison of the candidate tasks from decision set $\boldsymbol{D_m}$ in the proposed solving approach.

VI. EXPERIMENTAL RESULT

We tested the presented method on the PSLIB benchmark RCPSP J30 problem set [18], which consists of 480 problem instances. The considered objective function was the maximum completion time of all tasks (C_{max}).

A. The applied set of TSDAs

Table III. presents the applied task selection decision aspects (TSDAs). During testing, eight TSDAs were examined. The RC_DEST is a dynamic (time-dependent) TDSA, whose value is updated by the construction algorithm through recalculation in changing decision situations. The others are static (time-independent) TDSAs, which are calculated once at the start of the construction algorithm.

TABLE III
USED TASK-SELECTION ASPECTS

Notation	Description
NSucc	Number of successors
ProcT	Processing time of the task
CPM_EST	Earliest start time calculated by Critical Path Method (CPM)
CPM_EFT	Earliest finish time calculated by CPM
CPM_LST	Latest start time calculated by CPM
CPM_LFT	Latest finish time calculated by CPM
RC_DEST	Dynamic earliest start time considering dynamically the actual resource constraints at the given time of decision making
DD	Due date of the task

B. Numerical results

To evaluate the performance of the reactive solver, we used the lower bound LB_p as reference value for each benchmark instance p. LB_p is provided for all instances in the PSLIB dataset. The result of the reactive solver executed on problem instance p is denoted by $C_{max,p}$.We calculated the the average relative deviation (ARD) of the reactive solver execution for the complete J30 benchmark dataset by Equation (4), where smaller ARD value indicates the better result.

$$\text{ARD} = \frac{\sum_{p=1}^{P} \frac{C_{max,p} - LB_p}{LB_p}}{P} 100\ [\%] \quad (4)$$

Table IV presents the measurement results of tests considering only one individual TSDA. The priority values are set to 1 or -1 depending on the optimization direction. For simplicity, we presented the multiplication of z_k and w_k as the priority of the k^{th} TSDA.

The CPM_LST achieved the lowest ARD value, which was approximately 0.04923.

TABLE IV
ARD EVALUATION RESULTS FOR INDIVIDUAL TASK-SELECTION ASPECTS

Test #	Task selection priority multiplied with optimization direction								ARD
	RC_DEST	*NSucc*	*ProcT*	*CPM_EST*	*CPM_EFT*	*CPM_LST*	*CPM_LFT*	*DD*	
1	1	0	0	0	0	0	0	0	0.078260
2	0	1	0	0	0	0	0	0	0.140896
3	0	0	1	0	0	0	0	0	0.174179
4	0	0	0	1	0	0	0	0	0.091479
5	0	0	0	0	1	0	0	0	0.115464
6	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0.049226**
7	0	0	0	0	0	0	1	0	0.051196
8	0	0	0	0	0	0	0	1	0.095355
9	0	-1	0	0	0	0	0	0	0.100995
10	0	0	-1	0	0	0	0	0	0.124379

TABLE V
ARD EVALUATION RESULTS FOR COMBINED TASK SELECTION ASPECTS

Test	Task selection priority multiplied with optimization direction								ADR
#	*RC_DEST*	*NSucc*	*ProcT*	*CPM_EST*	*CPM_EFT*	*CPM_LST*	*CPM_LFT*	*DD*	
11	1	0	0	0	0	2	0	0	0.045571
12	1	-1	-1	1	1	1	1	0	0.061593
13	1	1	1	1	1	1	1	1	0.092054
14	1	1	-1	1	1	1	1	1	0.072527
15	1	-1	1	1	1	1	1	1	0.077187
16	1	-1	-1	1	1	1	1	1	0.061593
17	1	-1	-1	1	1	10	1	1	0.051461
18	1	-1	-1	1	1	20	1	1	0.049511
19	1	-1	-1	1	1	5	1	1	0.052672
20	2	0	0	0	0	1	0	0	0.038408
21	5	0	0	0	0	4	0	0	0.040886
22	5	-1	-1	1	1	1	1	1	0.05537
23	6	0	0	0	0	3	0	0	0.038408
24	6	-0.1	0.1	0	0	3	0	0	0.03776
25	6	-0.5	0.5	0	0	3	0	0	0.037249
26	6	-0.5	0.5	0	0	3	0.2	0	0.038329
27	6	-0.5	0.5	0	0	3	-0.2	0	0.037479
28	6	-0.5	0.5	0	0	3	0.5	0	0.037442
29	6	-0.5	0.5	0	0	3	1	0	0.037848
30	6	-0.5	0.5	0	0	3	-1	0	0.038671
31	6	-0.5	0.5	0	0.1	3	0	0	0.037556
32	6	-0.5	0.5	0	-0.1	3	0	0	0.037324
33	6	-0.5	0.5	0	0.5	3	0	0	0.037772
34	6	-0.5	0.5	0.05	0.05	3	0.05	0.05	0.037635
35	6	-0.5	0.5	0.1	0	3	0	0	0.037843
36	6	-0.5	0.5	-0.1	0	3	0	0	0.037999
37	6	-0.5	0.5	0.1	0.1	3	0.1	0.1	0.038126
38	6	-0.5	0.5	0.5	0	3	0	0	0.039375
39	6	-0.5	0.5	-0.5	0	3	0	0	0.041056
40	6	-0.5	0.5	1	0	3	0	0	0.041141
41	6	1	1	0	0	3	0	0	0.048826
42	6	-1	1	0	0	3	0	0	0.043549
43	6	-1	-1	0	0	3	0	0	0.056383
44	6	-1	-1	1	1	3	1	1	0.050003

Table V presents the results of 34 different tests where multiple TSDAs were used simultaneously. The table includes the applied priority values for each TSDA and the calculated ARD values.

The third-best result was given by RC_DEST, with an ARD value of approximately 0.07826 (Table IV Test #1). When these two selection criteria were applied together, an even better result was obtained than when used separately. For example, with priority value of RC_DEST 1 and CPM_LST 2 (Test #11), the ARD value was approximately 0.04557. Swapping the priorities of the two selection criteria (Test #20) reduced the ARD to approximately 0.03841.

An interesting observation was that in the combined solution, it was advantageous to assign a higher priority to the criterion that performed worse individually. The specific values of the TSDAs were not as important as their relative ratio. For instance, increasing the priorities threefold (e.g., from 2:1 to 6:3) resulted in an unchanged ARD value.

This experimental finding can be proven mathematically. Considering the F function (3) used for relative comparison, multiplying the priority values by a constant is equivalent to multiplying the final result of the F function by the same constant. Since the result of F is compared to zero, multiplying by any non-negative real number does not change its relation to zero. If F was greater, less, or equal to zero, it remains so. This should be considered when fine-tuning the priority values of TSDAs.

Involving additional selection criteria further improved the results. Table V examples show that the ARD value could be significantly reduced with different priority values. RC_DEST and CPM_LST remained dominant, but other criteria, such as NSucc and ProcT, also proved useful with smaller priority values. These criteria influence the decision when the F function value is close to zero based on higher-priority criteria. The priority of NSucc is negative because the method favors candidate tasks with a higher number of successor tasks. For other criteria, smaller numbers are more favorable.

In this experiment, the best result was achieved with the following priorities: 6; -0.5; 0.5; 0; 0; 3; 0; 0, (Test #25). The ARD value achieved this way was approximately 0.037249. Table V also shows that an ARD value below 0.038 could be reached with many settings. This is favorable because, for a given specific system, the set priority value scheme can be used effectively, providing a sufficiently sharp solution while maintaining flexibility.

While lower ARD values can be achieved with search algorithms, they come with significantly longer computation times due to the need to generate a large number of solutions iteratively. In contrast, a reactive construction algorithm produces only one solution. Reactive scheduling quickly adapts to real-time changes, as it operates based on predefined priority rules and decision criteria, allowing for immediate decision-making. In the investigated situations, the reactive approach may be more advantageous, as it does not require waiting for the search algorithms to respond, enabling faster reactions to environmental changes.

The experiment demonstrated that combining multiple task-selection aspects yields better solutions than using a single aspect. The decision-making method based on relative qualification can handle any finite number of selection criteria together. Assigning priority values to TSDAs is straightforward and flexible.

VII. Conclusion

In this paper, we present a novel reactive scheduling approach for extended project scheduling problems, aiming to create feasible and fast schedules for multiple projects with detailed resource requirements. The model uses a serial generation scheme with a new multi-priority decision-making procedure that considers many different decision aspects for deterministically selecting tasks.

The proposed extension is adaptable to a wide range of scheduling problems due to its problem-independent nature and ability to incorporate diverse decision aspects. These aspects can be calibrated and incorporated similarly to classical priority rules.

Our performance tests, conducted on the PSLIB J30 benchmark series, demonstrated that the combination of multiple decision aspects outperforms single aspects in minimizing the latest completion time. This supports our hypothesis that combining decision aspects using the relative qualification model is advantageous.

Future research will focus on applying extensions in the model to address the reactive decision-making requirements related to scheduling in cyber-physical systems.The proposed approach can be flexibly adapted to various optimization objectives and effectively applied to other selection or optimization problems due to its problem-independent elements. The schedule generation scheme, directed by the next task selection method, handles problem-specific constraints and can work with any generation scheme.

References

[1] A. A. B. Pritsker, W. J. Lawrence and P. M. Wolfe, "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach," *Management Science*, vol. 16, no. 1, pp. 93–108, Sept. 1969.

[2] J. Blazewicz, J. K. Lenstra and A. H. Kan, "Scheduling subject to resource constraints: classification and complexity," *Discrete Applied Mathematics*, vol. 5, no. 1, pp. 11–24, Jan. 1983, DOI: 10.1016/0166-218X(83)90012-4.

[3] R. Kolisch, S. Hartmann, "Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis", *Project scheduling*, pp. 147–178, 1999., DOI: 10.1007/978-1-4615-5533-9_7

[4] R. Kolisch, S. Hartmann, "Experimental investigation of heuristics for resource-constrained project scheduling: An update", *European journal of operational research*, vol. 174, no. 1, pp. 23–37, 2006., DOI: 10.1016/j.ejor.2005.01.065

[5] R. Pellerin, N. Perrier and F. Berthaut, "A survey of hybrid metaheuristics for the resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 280, no. 2, pp. 395–416, 16 Jan. 2020, DOI: 10.1016/j.ejor.2019.01.063.

[6] S. Hartmann and D. Briskorn, "An updated survey of variants and extensions of the resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 297, no. 1, pp. 1–14, 16 Feb. 2022, DOI: 10.1016/j.ejor.2021.05.004.

[7] J. E. Kelley and M. R. Walker, "Critical-path planning and scheduling," *Proceedings of the Eastern Joint Computer Conference*, 1959. DOI: 10.1145/1460299.1460318

[8] A. A. Pritsker and W. Happ, *GERT: Graphical Evaluation and Review Technique*, Wiley, 1966.

[9] R. W. Conway, W. L. Maxwell, and L. W. Miller, *Theory of Scheduling*, Addison-Wesley, 1967.

[10] J. Luo, M. Vanhoucke, J. Coelho, W. Guo, „An efficient genetic programming approach to design priority rules for resource-constrained project scheduling problem", *Expert Systems with Applications*, vol. 198, 2022, DOI: 10.1016/j.eswa.2022.116753.

[11] W. Guo, M. Vanhoucke, J. Coelho, J. Luo, "Automatic detection of the best performing priority rule for the resource-constrained project scheduling problem", *Expert Systems with Applications*, vol. 167, 2021, DOI: 10.1016/j.eswa.2020.114116.

[12] K. Taha, "Methods That Optimize Multi-Objective Problems: A Survey and Experimental Evaluation," *IEEE Access*, vol. 8, pp. 80 855–80 878, 2020, DOI: 10.1109/ACCESS.2020.2989219.

[13] Y.-H. Zhang, Y.-J. Gong, J. Zhang and Y.-b. Ling, "A hybrid evolutionary algorithm with dual populations for many-objective optimization," *2016 IEEE Congress on Evolutionary Computation (CEC)*", Vancouver, BC, Canada, 2016, pp. 1610–1617, DOI: 10.1109/CEC.2016.7743981

[14] I. R. Meneghini, M. A. Alves, A. Gaspar-Cunha and F. G. Guimarães, "Scalable and customizable benchmark problems for many-objective optimization," *Applied Soft Computing*, vol. 90, May 2020, DOI: 10.48550/arXiv.2001.11591.

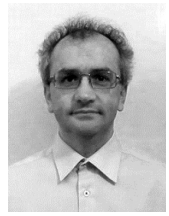
[15] H. Dai, W. Cheng and P. Guo, "An Improved Tabu Search for Multi-skill Resource-Constrained Project Scheduling Problems Under Step-Deterioration," *Arabian Journal for Science and Engineering*, vol. 43, no. 6, pp. 3279–3290, Jan. 2018, DOI: 10.1007/s13369-017-3047-4.

[16] S. Mane and M. R. Narasingarao, "A chaotic-based improved many-objective jaya algorithm for many-objective optimization problems," *International Journal of Industrial Engineering Computations*, vol. 12, no. 1, pp. 49–62, Oct. 2020, DOI:10.5267/j.ijiec.2020.10.001.

[17] K. Mihály and Gy. Kulcsár, "A New Many-Objective Hybrid Method to Solve Scheduling Problems", *International Journal of Industrial Engineering and Management*, vol. 14, no. 4, pp. 326–335, Dec. 2023.

[18] PSLIB Single Mode Scheduling Benchmark Dataset, last accessed 2024/04/03, https://www.om-db.wi.tum.de/psplib/getdata_sm.html.

Krisztián Mihály is an assistant lecturer at the Institute of Information Science, University of Miskolc (Hungary). He received his M.Sc. in Information Engineering from the Budapest University of Technology and Economics (Hungary) in 2008. He is currently working on his PhD thesis. He is working as a development architect at SAP Hungary Ltd., responsible for SAP PLM for Process Industry solutions. He is recipient of the "Award of the Best Trainer of the Year at SAP". His research interests include production planning and scheduling, project planning and scheduling, many-objective optimization, metaheuristics, and software architectures.



Gyula Kulcsár is an associate professor in the Institute of Information Science at the University of Miskolc (Hungary). He received an M.Sc. in Information Science from the University of Miskolc (Hungary) in 2001. and a Ph.D. in Information Science from the University of Miskolc (Hungary) in 2008. His research interests include production planning and scheduling, project planning and scheduling, many-objective optimization, metaheuristics, discrete event-driven simulation, and manufacturing control. He is a member of the public body of the Hungarian Academy of Sciences. He is the recipient of the Young Researcher Scientific Award from the Hungarian Academy of Sciences in 2008.

Mónika Kulcsárné Forrai an associate professor in the Institute of Information Science at the University of Miskolc (Hungary). She received an M.Sc. in Information Engineering from the University of Miskolc (Hungary) in 2001, and a Ph.D. in Information Science and Technology from the University of Miskolc (Hungary) in 2018. Her research interests include scheduling, search algorithms, optimization, production planning and control, enterprise resource planning, and project scheduling. She is a member of the public body of the Hungarian Academy of Sciences.

TaxaTreeMapper: A Novel Algorithm for Phylogenetic Ancestral State Reconstruction Using Set Theory

Osama A. Salman[1], and Gábor Hosszú[2]

***Abstract*—To determine evolutionary relationships, it is crucial to conduct phylogenetic ancestral state reconstruction. Although widely used, existing algorithms, such as Fitch's, are challenged by the computational demands of complex datasets. This study introduces the TaxaTreeMapper algorithm, which presents a streamlined approach that optimizes phylogenetic analysis. TaxaTreeMapper reduces computational time without compromising accuracy by performing ancestral state reconstruction in a single 'leaf-to-root' traversal. Our comparative study shows that TaxaTreeMapper correlates strongly with the Fitch algorithm and demonstrates superior efficiency, especially in identifying global minima in extensive datasets. This makes it significant in large-scale evolutionary studies.**

***Index Terms*—Algorithmic efficiency, Ancestral state reconstruction, Data processing in phylogenetics, Evolutionary tree optimization, Fitch algorithm, Machine learning applications in phylogenetics, Parsimony score, Phylogenetic analysis**

I. INTRODUCTION

THE quest for accurate ancestral state reconstruction in phylogenetics often encounters significant challenges, particularly with algorithms like Fitch's, which, while being intuitive and simple, may falter in cases of complex evolution or convergence [1]. Ancestral state reconstruction combines information about the evolutionary relationships of phylogenetic trees with the observed state of individual nodes. Each node represents a single taxon (taxonomic unit) [2]. Complex evolution in this context implies scenarios where evolutionary paths are shaped by multiple factors such as frequent mutations, horizontal gene transfer, genetic drift, or hybridization. These elements introduce intricacies in evolutionary histories, making accurate reconstruction a challenging endeavor.

A common limitation among many algorithms that seek to reconstruct common ancestors and determine the minimum parsimony score for a given tree is their reliance on a two-stage process: the 'leaf-to-root' followed by the 'root-to-leaf' traversal.

[1] Budapest University of Technology and Economics, Faculty of Electrical Engineering and Informatics, and Doctoral School of Informatics, Department of Electron Devices, Budapest, Hungary (E-mail: osamaalisalman.khafajy@edu.bme.hu)

[2] Budapest University of Technology and Economics, Faculty of Electrical Engineering and Informatics, Department of Electron Devices, Budapest, Hungary (E-mail: hosszu.gabor@vik.bme.hu)

DOI: 10.36244/ICJ.2025.5.2

Bidirectional tree traversal is a technique where the traversal progresses both from the root to the leaves and from the leaves to the root of a tree. This approach is beneficial in scenarios requiring information aggregation from both directions to make decisions at each node. A notable application of bidirectional tree traversal is in robot motion planning, where such a strategy enhances efficiency and avoids complex boundary value problems [3].

This two-pronged approach, while effective in certain contexts, often leads to increased computational complexity and may not always yield the most optimized results in terms of global minimum parsimony score.

1) Novel Contribution

This study presents a new method, TaxaTreeMapper, which is designed to traverse a given phylogenetic tree and determine its minimum parsimony score directly. This approach contributes to a more optimized method for identifying the global minimum. Our method seeks to address the limitations by reducing the computational process to a single-stage traversal. This not only simplifies the analysis but also reduces the computational effort required, making it a significant step forward in the pursuit of efficient phylogenetic analysis. By streamlining the process of ancestral state reconstruction, our approach aims to enhance the efficiency of phylogenetic tree evaluations, particularly in complex evolutionary scenarios.

While the TaxaTreeMapper algorithm enhances the efficiency of phylogenetic analysis by significantly reducing computational time, often less than that required by the Fitch algorithm, it is also designed to reliably identify all global minima in a given set of phylogenetic trees. However, it should be noted that alongside these global minima, TaxaTreeMapper may also occasionally include some local minima, erroneously presenting them as global. Despite this, the set of solutions provided by TaxaTreeMapper will contain all the true global minima, ensuring comprehensive coverage of the most parsimonious trees.

The article is structured as follows: First, it presents the concepts necessary for the theoretical background of the research, including phylogenetic inference methods, primarily the Fitch algorithm. Second, it presents the

developed method called TaxaTreeMapper algorithm, followed by the achieved results and their discussion. Finally, the article concludes with a summary of findings and a list of references cited in the literature.

II. BACKGROUND

1) Pattern systems, pattern evolution and scriptinformatics

A pattern system is a type of symbolic communication that includes symbols, syntax, and layout rules. Some pattern systems, like Morse code and Unicode, have evolved over time. The study of the evolution of pattern systems is called pattern evolution research. Human writing systems, or scripts, are a distinct type of pattern system. The study of the evolution of scripts is known as scriptinformatics, a subfield of pattern evolution. The TaxaTreeMapper phylogenetic inference algorithm was initially developed for evolutionary modeling of scripts, but it can also be applied to other taxa. TaxaTreeMapper has a broad scope beyond just scriptinformatics and belongs to any kind of evolutionary research.

Understanding the evolution and classification of different taxa has always been at the heart of scientific research. Various mathematical, computational, and heuristic models have been proposed over the years, aiming at creating a more structured and accurate representation of the evolutionary process. Significant advancements have been made in pattern systems, especially when applied to historical scripts [4] established a three-layer logical relationship for glyphs, further improved by the addition of a semantic layer in [5]. Recent contributions by [6] introduced the style layer, enhancing the depth of the analysis. Hosszú's emphasis on glyph complexity as a metric for reliability in comparing graphemes provides a foundation for comparative studies [7]. Such advancements have found applications in differentiating scripts using cluster analysis [8] and leveraging neural networks for glyph similarity studies [9].

In this study, we address the terminology used to describe taxonomic traits in phylogenetic analysis, a matter of great importance for ensuring clarity and precision. While 'character' is a term traditionally used in phylogenetics to denote the attributes or traits of organisms, in the field of scriptinformatics and in certain phylogenetic contexts, the term 'feature' is often utilized interchangeably. This duality in terminology is evident in recent research, such as the work [10] where features in evolutionary analysis of script variants are critically examined. Similarly, [11] employ the term in the context of phylogenetic analysis of script varieties, demonstrating its relevance and application. Further, their 2022 study on a phenetic approach to script variants also underscores the interchangeable use of these terms [12]. For the purposes of this article, we adopt this dual terminology, using 'feature' and 'character' interchangeably, with the understanding that both refer to taxonomic traits in our phylogenetic analysis. This approach aligns with broader scientific discourse and avoids potential ambiguity, particularly in scriptinformatics where 'character' might otherwise be confused with 'grapheme' or 'symbol'.

2) Phylogenetic inference methods

Phylogenetic methods have expanded their applicability beyond just biological evolution. For instance, its usage in linguistics has paved the way for constructing evolutionary trees for languages [13]. Phylogenetic analysis, especially with its parsimonious approach rooted in the Ockham's razor principle [14], has been paramount in creating hierarchical taxonomic structures. Delineation of synapomorphies further emphasizes the model's capability to account for a vast number of features in a simplistic manner [15].

Two significant comparative criteria, Maximum Parsimony [16] and Maximum Likelihood (ML), have emerged as primary techniques for tree optimization [11]. While MP revolves around the parsimony principle, ML uses probabilistic models to evaluate evolutionary event likelihoods. The Bayesian approach, exemplified by MrBayes software, further exemplifies the nuanced relationship between data and tree probabilities [17].

When exact and exhaustive searches are too costly or time-consuming, heuristic methods become necessary. While these approaches aim to approximate optimal solutions in the solution space, they cannot guarantee the identification of the globally optimal phylogenetic tree. To enhance heuristic search efficiency and improve upon the phylogenetic trees constructed, a branch-swapping algorithm, known as 'swapping' [10].

The Subtree Prune and Regraft (SPR), the Nearest Neighbor Interchange (NNI), and Tree Bisection and Reconnection (TBR), each come with their unique attributes, with TBR being the most computational but potentially offering the shortest tree [17, 18]. The present research focuses on phylogenetic inference methods, which involve searching for optimal phylogenetic trees. Only models where the evolutionary process can be estimated with a tree, rather than a network, are considered.

The search for the most realistic phylogenetic tree, despite its comprehensiveness, faces challenges with larger datasets [18]. Alternative heuristic methods like the Wagner method [19], the Branch and Bound technique [20] and Hill-Climbing [21] offer solutions with varied degrees of optimality and computational efficiency. Regarding Hill-Climbing algorithm is effective for finding local optima in phylogenetic trees by refining initial configurations, focusing on measures like parsimony. However, it falls short of guaranteeing the global optimum, often getting trapped in local optima. This underscores the necessity for supplementary methods to circumvent such limitations and achieve a more comprehensive search for the optimal phylogenetic tree [10, 11]. Additionally, visualization tools like histograms provide insights into the distribution of tree lengths, aiding in the discernment of optimal trees.

Matrix-based approaches in phylogenetics offer a variety of methods to derive evolutionary relationships among taxa. One classical group of methods, distance matrix methods, such as the Neighbor-Joining (NJ) and UPGMA, directly work with matrices that represent pairwise distances between taxa to infer a phylogenetic tree [22]. Alternatively, spectral methods exploit the eigenvalues and eigenvectors of matrices derived from genetic data. Cavender and Felsenstein's method, based on eigendecomposition of sequence similarities, is a prime example [23]. As another approach, quartet methods like the Q-method employ matrices showcasing relationships between quartet sets of taxa to infer broader trees [24]. Another avenue, character compatibility, creates a taxa by feature (aka character) matrix, checking feature compatibility to infer relationships [25]. Recent research has also highlighted the potential of algebraic statistics in phylogenetics, where algebraic techniques decode phylogenetic problems using matrix operations [26]. Lastly, phylogenetic networks, which encapsulate complex evolutionary patterns like hybridization, can be understood and analyzed using matrix representations [27].

In conclusion, the domain of phylogenetic inference has witnessed extensive research, with a multitude of evolutionary models and phylogenetic inference algorithms proposed. The ultimate objective remains the construction of accurate and representative evolutionary trees, aiding in a deeper understanding of taxa evolution. As computational power and methodologies continue to evolve, it's estimated that even more sophisticated models will emerge, bridging any existing gaps in the space of phylogenetics.

3) Fitch parsimony and algorithms

Fitch's contributions to the field of phylogenetics are evident through his development of distinct methods that address the reconstruction of evolutionary histories. One such method is the *Fitch parsimony*, which operates on a parsimony principle aiming to discern the evolutionary tree with the least number of feature state (aka character state) changes. Crucially, this method accommodates multistate features, allowing them to be disordered and unpolarized, meaning that transitions between any feature states are possible in a single evolutionary step. This principle is computationally manifested in the *Fitch algorithm*, which calculates the parsimony score, indicating the total number of feature state transitions for a specific tree topology [28]. The Fitch algorithm is a fundamental method in the field of phylogenetics. It was introduced by Walter Fitch in the 1970s and has since become a fundamental tool for ancestral state reconstruction based on parsimony principles [28]. On the other hand, Fitch, in collaboration with Margoliash, devised the *Fitch-Margoliash Phylogenetic Inference Algorithm*. Instead of feature states, this method is grounded on genetic distance data. Utilizing a weighted least squares clustering approach, it emphasizes the accuracy of genetic distances between species in the tree, giving more weight to closely related sequences. This method offers higher accuracy, albeit at the expense of computational efficiency when compared to alternatives like the neighbor-joining technique [29].

Fitch algorithm is a commonly used tool for ancestral state reconstruction based on parsimony methods. This algorithm works by minimizing the number of evolutionary changes [28] along the branches of a phylogenetic tree. Fitch algorithm, while not directly calculating the total length or Maximum Parsimony [16] score of a phylogenetic tree in a single computation, effectively minimizes the number of evolutionary changes across the tree. This minimization is achieved indirectly through the algorithm's two-pass process. In the first pass, the algorithm performs a bottom-up traversal of the tree, during which it identifies the possible feature states for each internal node without assigning specific branch lengths. In the second pass, a top-down traversal assigns definitive states to these nodes [30, 31].

During this process, the Fitch algorithm seeks to minimize the number of state changes at each step. The branch lengths, defined as the number of feature state changes between nodes, are indirectly determined through this process. The overall tree length, representing the sum of these branch lengths, is thus a result of the algorithm's optimization of state changes at each individual node and branch, rather than a direct calculation of the total tree length [30].

Fitch algorithm, originally developed for the parsimony-based reconstruction of phylogenies, is inherently designed to handle bifurcating or binary trees. Its two-phase traversal approach, involving postorder and preorder tree traversals, is optimized for dichotomous branching. When faced with polytomous trees, or trees with nodes having more than two descendants, the Fitch algorithm encounters challenges. Polytomies, which can be seen as either unresolved evolutionary relationships (soft polytomies) or simultaneous divergence events (hard polytomies), don't fit neatly into the binary framework of Fitch's method [32]. Adapting the algorithm to cater to these non-binary nodes introduces complexities and requires additional considerations or modifications. While some phylogenetic software tools have developed strategies to handle or resolve polytomies, the inherent limitation of Fitch's original design concerning polytomies remains a recognized challenge in the field of phylogenetics [16].

The Fitch algorithm employs a two-stage process for ancestral state reconstruction, beginning at the leaf nodes with known genetic states and moving toward the root to infer the most parsimonious common ancestor at each internal node as illustrated in Figure 1. For example, if taxa A and B both have a state of '1' for a particular characteristic, their common ancestor is presumed to also have the state of '1'. When discrepancies arise (e.g., A=1, B=0), the ancestor may inherit a set that includes

both states. In the next stage, the algorithm resolves these sets by working from the root back to the leaves, selecting states that minimize changes across the tree. While effective, the Fitch method can be computationally intensive for large datasets.

4) Pearson correlation

The Pearson correlation coefficient (also known as Pearson's r or simply the correlation coefficient) measures the linear relationship between two variables, typically denoted as X and Y. The formula for calculating the Pearson correlation coefficient is as follows:

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}}, \quad (1)$$

where n is the number of data points (the size of the dataset), X_i and Y_i are the individual data points of the variables X and Y, and finally $\bar{X}$ and $\bar{Y}$ are the mean (average) values of X and Y, respectively.

The incorporation of Pearson correlation in our study is integral to the assessment of phylogenetic relationships and evolutionary analysis. Pearson's correlation, a measure of the linear correlation between two variables, provides a quantifiable means to assess the degree of similarity or divergence between different taxa based on their phylogenetic profiles. This statistical tool is particularly effective in discerning the strength and direction of a linear relationship between two sets of data, which in our context, are the phylogenetic traits or features of different organisms. By applying Pearson correlation, we can systematically compare these traits to draw inferences about evolutionary patterns and relationships. Our approach aligns with contemporary advancements in phylogenetic profiling, where measures like Pearson correlation have been used to infer global protein-protein interactions and handle large genomic datasets effectively, as demonstrated in the study of Saccharomyces cerevisiae and Escherichia coli genomes [33]. This method's efficacy, especially in comparison to other measures such as mutual information and distance correlation, underscores its relevance and utility in our analysis.

III. METHOD

In this study, we introduce a novel methodological approach to determine the maximum parsimony in phylogenetic trees, distinguished by its efficient single-stage process. This approach is compared with the established Fitch algorithm, a keystone in phylogenetic analysis, known for its two-stage 'leaf-to-root' and 'root-to-leaf' traversal process. Notably, the Fitch algorithm, which has been widely used for phylogenetic tree reconstruction and serves as a benchmark in our comparative analysis.

Unlike conventional two-stage methods such as the Fitch algorithm, which operate through both 'leaf-to-root' and 'root-to-leaf' stages, our method simplifies the analysis by focusing exclusively on a 'leaf-to-root' traversal. This innovation effectively halves the computational steps typically required, as it eliminates the need for the subsequent 'root-to-leaf' stage. Conceptually, if Y represents the computational effort of a traditional method like Fitch's, then our method, X, can be said to operate at $\frac{Y}{2}$ in terms of computational, or mathematically, $f(X) = 2 \cdot X$ in terms of efficiency.

This significant enhancement not only accelerates the analytical process but also maintains the accuracy and robustness needed for phylogenetic studies. Our approach (TaxaTreeMapper) represents a substantial advancement in phylogenetic analysis, offering a more streamlined and time-efficient solution for uncovering evolutionary relationships. This methodology, with its single-stage focus, is not only a testament to the potential for innovation in phylogenetic analysis but also a practical solution that addresses the computational challenges often encountered in extensive biological datasets.

The TaxaTreeMapper algorithm applies set theory operations to clarify phylogenetic relationships. It starts with a 'leaf-to-root' assessment, as in Figure 1 where taxa *A*, *B*, and *C* are compared for their feature states. Discrepancies between taxa, like $A = 1$ and $B = 0$, lead to an interim ambiguous state $Z = 2$. The algorithm resolves this by checking the overlap with any resolved neighboring node states. If taxon *C* also has the state of 1, then the algorithm concludes the ancestral state *R* to be 1, through the intersection with the ambiguous state. Illustrated in Figure 1, this method streamlines the determination of the most likely internal node states, improving the precision of phylogenetic tree reconstruction.

The TaxaTreeMapper streamlines the process for ancestral state reconstruction into a single stage. It also starts at the leaves, but as it ascends the tree, it uses information from the subsequent ancestor (e.g., ancestor of A & B derives its state from C) to determine the states of intermediate ancestors directly refer to Figure 1. This approach not only simplifies the state determination process but also allows for simultaneous calculation of the tree length and the total number of changes, enhancing efficiency particularly for extensive datasets.

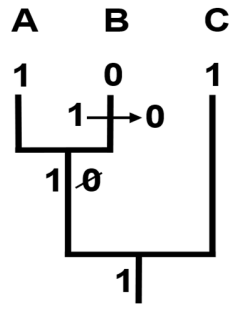


Figure 1: 'Leaf-to-Root' ancestral state reconstruction in TaxaTreeMapper

Algorithm 1: The main steps of the TaxaTreeMapper

Input: A node of a phylogenetic tree (root node to start), dataset
Output: The inferred state for the node, treeLength for the entire tree

Function TaxaTreeMapper(node, dataset):
1. If node is a leaf:
  1.1. Return the state of the leaf node from the dataset, and 0 as the treeLength.
2. Initialize Gab as an empty set for accumulating the node's inferred state.
3. Initialize localTreeLength = 0 to track state changes at this node.
4. For the first child of the node, establish a reference state (Ga):
  4.1. Ga, childTreeLength = TaxaTreeMapper(first child, dataset).
  4.2. Set Gab to Ga initially.
  4.3. Update localTreeLength += childTreeLength.
5. For each remaining child k (starting from the second child to the last):
  5.1. Gb, childTreeLength = TaxaTreeMapper(k, dataset).
  5.2. Update localTreeLength += childTreeLength.
  5.3. Perform intersection and union operations:
  5.4. Intersection: If Ga ∩ Gb is not empty, Gab = Gab ∩ Gb.
  5.5. Union with resolution: If Ga ∩ Gb is empty, then Gab = Gab ∪ Gb, but resolve `{2}` where possible:
  5.6. For each feature in Gab marked as `{2}`, if Gb has a known state, replace `{2}` in Gab with Gb's state.
  5.7. Conversely, for each `{2}` in Gb and known in Ga, update Gab accordingly.
  5.8. Determine unique changes: uniqG = (Ga Δ Gb) - `{2}` elements, where Δ represents the symmetric difference.
  5.9. Update localTreeLength for each unique change not involving `{2}`, as these represent evolutionary events.
  5.10. Mark unresolved differences as `{2}` in Gab for the next iteration.
6. After processing all children, the dataset is updated with the resolved state Gab for the internal node.
7. Return Gab as the node's state and localTreeLength.

Ambiguous states, denoted by a predetermined value within the dataset, are systematically managed by the TaxaTreeMapper algorithm, particularly in steps 5.4 and 5.6 These steps incorporate ambiguous states into the intersection operation, ensuring that uncertainties in data do not compromise the accuracy of the phylogenetic analysis.
Furthermore, the calculation of the symmetric difference between sets identifying features unique to each taxon occurs in step 5.8. This difference highlights the evolutionary divergence and is crucial for calculating the tree length, representing the extent of evolutionary adaptations since the taxa branched from their last common ancestor.
The algorithm then updates the dataset with a new set representing hypothetical ancestral taxa in step 6. This new set, a combination of intersected and unique elements, is pivotal for updating the feature set for each node in a 'leaf-to-root' traversal of the phylogenetic tree. The 'leaf-to-root' traversal, the primary focus of the TaxaTreeMapper algorithm, simplifies the analysis process and enhances the precision of phylogenetic inference by compiling shared and distinctive traits accurately. This is illustrated in steps 4 through 7 of the TaxaTreeMapper Algorithm 1.
Initially, the TaxaTreeMapper determines the number of taxa from the given dataset and initializes various variables and counters to their respective default values. The primary focus then shifts to traversing the phylogenetic tree sequence, where each feature is processed in sequence.

Beginning at the top of Algorithm 1, the algorithm initializes the necessary variables, including the dataset. The main input and output in this case the phylogenetic tree *node, dataset* as input and *treeLength* and inferred state for the phylogenetic tree as an output.

IV. RESULT AND DISCUSSION

This section presents a comparative analysis of the TaxaTreeMapper algorithm against Fitch algorithm, focusing on tree length determination, computational efficiency, and the handling of cladograms in phylogenetic analysis. We then discuss the inherent advantages of the TaxaTreeMapper algorithm, underpinned by the empirical results.

1) Comparative Analysis

In Figure 2, we illustrate the comparative analysis of tree lengths generated by the TaxaTreeMapper algorithm and the Fitch algorithm. The histogram in Figure 2.a, the sorted length curves in Figure 2.b, and the boxplot in Figure 2.c collectively highlight the similarity in tree length calculations and the distinct efficiencies between the two methods. Our findings suggest that TaxaTreeMapper consistently identifies the global minimum for maximum parsimony trees more efficiently than the Fitch process, which relies on a two-phase approach.

Figure 2 underscores the significant overlap in tree length evaluations between TaxaTreeMapper and Fitch across approximately 2.5 million diverse phylogenetic trees. This comparison validates the efficiency of TaxaTreeMapper in closely matching the established Fitch method while using a single-phase approach.

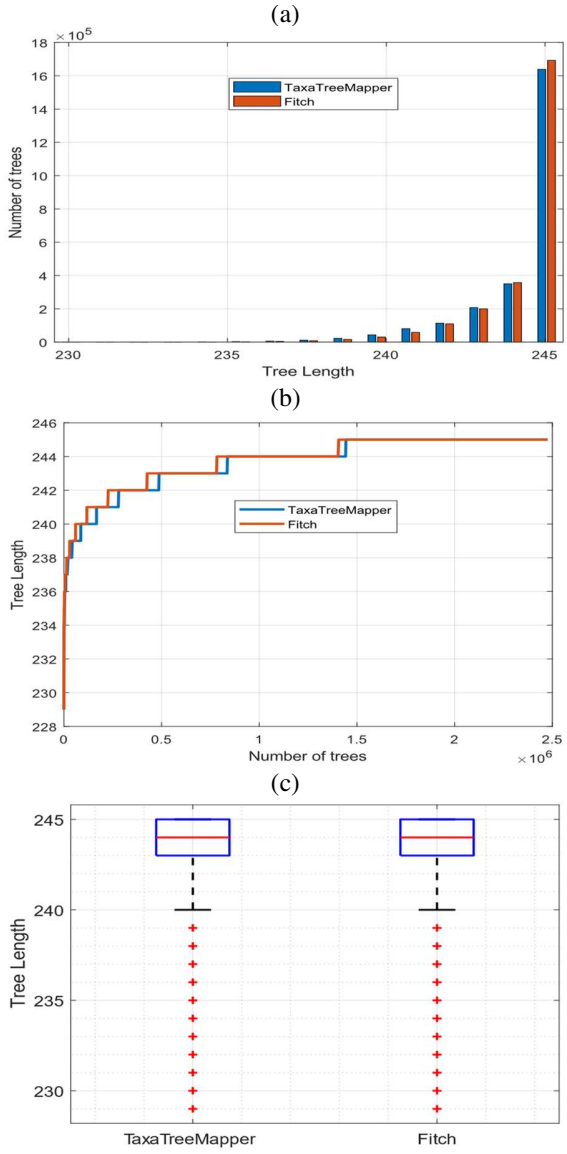


Figure 2: Comparative Efficiency of Tree Length Determination between TaxaTreeMapper and Fitch Algorithms

Despite the foundational differences in their operational stages—TaxaTreeMapper using a single-phase approach versus the two-phase process of Fitch—both methods consistently identified the global minimum tree length and exhibited a significant overlap in their evaluations of near-optimal tree lengths. This comparison not only demonstrates the algorithms' capability to accurately determine the most parsimonious tree but also validates the efficiency of TaxaTreeMapper in achieving results that align closely with the established Fitch method.

2) Pearson Correlation Coefficient Analysis

A Pearson correlation coefficient (r) analysis further substantiates the similarity between the algorithms. With r values of 0.91 for the same amount of trees that been tested in the comparative analysis. The analysis confirms a strong positive linear relationship between the tree lengths determined by TaxaTreeMapper and those by Fitch, indicating a convergence towards a global minimum by the TaxaTreeMapper algorithm.

It is noteworthy that when r equals 0.91, the estimated number of trees was approximately 2.5×10^6. Conversely, when r equals 0.956, the number of trees was precisely 2988. This observation provides a clear indication that the TaxaTreeMapper algorithm converges towards a global minimum.

3) Computational Efficiency

In Figure 3, we present of the running time of TaxaTreeMapper algorithm and Fitch algorithm. The elapsed time measurements clearly illustrate that TaxaTreeMapper outperforms the Fitch algorithm in terms of computational efficiency.

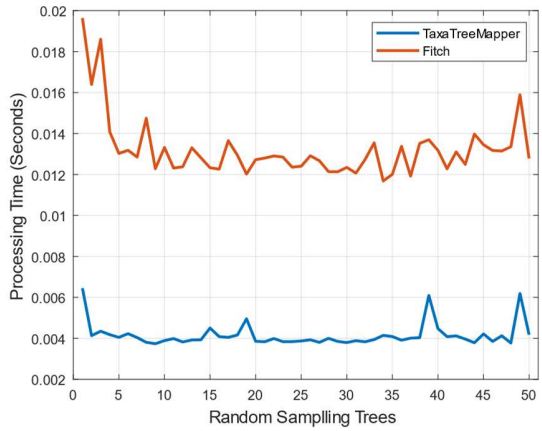


Figure 3: Tree processing time for TaxaTreeMapper algorithm verses Fitch algorithm.

The experimental evaluation was conducted on a system running Windows 10 Pro Version 22H2, equipped with an Intel(R) Core(TM) i7-2720QM CPU at 2.20GHz and 20GB of RAM, operating on a 64-bit architecture. The performance metrics for both the TaxaTreeMapper and Fitch algorithms were obtained using implementations coded in MATLAB R2023b. This hardware and software environment was chosen to ensure a consistent and controlled platform for benchmarking the computational efficiency of the phylogenetic analysis algorithms under investigation.

In Figure 4, the results show that TaxaTreeMapper generated four cladograms with identical minimum parsimony scores 229, also known as tree length. In contrast, the application of the Fitch algorithm yielded only two cladograms (a and b) as a global minimum. On other hands, the TaxaTreeMapper algorithm identified cladograms Figure 4.c and Figure 4.d as having a global minimum parsimony [16] score of 229. However, the Fitch algorithm attributed these same cladograms with higher tree lengths of 234. This divergence in scores initially suggests that TaxaTreeMapper incorrectly assess

these cladograms as optimal. While it may appear as a limitation, a closer examination of the Pearson correlation coefficient (r) between the tree lengths calculated by TaxaTreeMapper and Fitch reveals a high degree of correlation across the dataset. This indicates that, despite the identified discrepancies, the TaxaTreeMapper algorithm performs consistently with the Fitch algorithm for most cases.

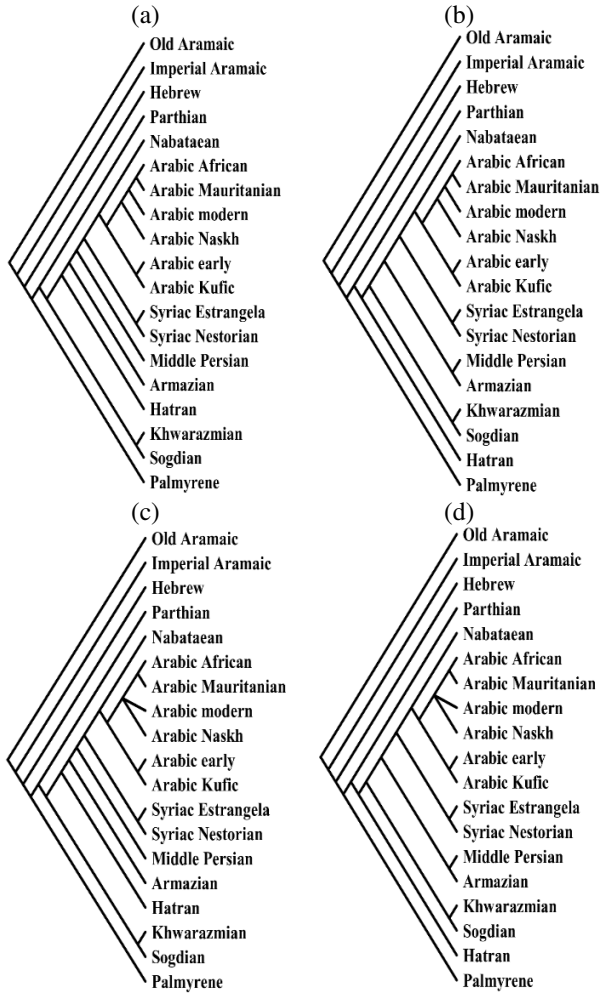


Figure 4: Comparative Cladograms of Semitic Scripts Generated by TaxaTreeMapper Algorithm

4) Empirical Validation

Table 1 quantifies the runtime efficiency of the TaxaTreeMapper algorithm, denoted as α, against the Fitch algorithm, denoted as β, across 50 random sampling trees. The use of α and β provides a simplified notation that facilitates the mathematical comparison of runtimes. The column labeled α to represent the runtime of TaxaTreeMapper, while the column labeled β for the Fitch algorithm's runtime. This symbolic representation streamlines the discussion and calculation of the efficiency metric, particularly in the subsequent computation of the Mean Squared Error (MSE).

The mean squared error (MSE), calculated is equal to 6.48×10^{-6}, is derived from the squared differences between α and $\frac{\beta}{2}$. A lower MSE corroborates the hypothesis that TaxaTreeMapper is about twice as fast as the Fitch algorithm.

TABLE I
COMPARATIVE RUNTIME ANALYSIS OF TAXATREEMAPPER (TTM) AND FITCH ALGORITHMS ACROSS PHYLOGENETIC TREES

	Runtime			$\left(\alpha - \frac{\beta}{2}\right)^2$
	α	β	$\frac{\beta}{2}$	
1	0.006	0.02	0.01	0.000016
2	0.004	0.016	0.008	0.000016
3	0.004	0.019	0.0095	0.00003025
4	0.004	0.014	0.007	0.000009
5	0.004	0.013	0.0065	0.00000625
6	0.004	0.013	0.0065	0.00000625
7	0.004	0.013	0.0065	0.00000625
8	0.004	0.015	0.0075	0.00001225
9	0.004	0.012	0.006	0.000004
10	0.004	0.013	0.0065	0.00000625
11	0.004	0.012	0.006	0.000004
12	0.004	0.012	0.006	0.000004
13	0.004	0.013	0.0065	0.00000625
14	0.004	0.013	0.0065	0.00000625
15	0.005	0.012	0.006	0.000001
16	0.004	0.012	0.006	0.000004
17	0.004	0.014	0.007	0.000009
18	0.004	0.013	0.0065	0.00000625
19	0.005	0.012	0.006	0.000001
20	0.004	0.013	0.0065	0.00000625
21	0.004	0.013	0.0065	0.00000625
22	0.004	0.013	0.0065	0.00000625
23	0.004	0.013	0.0065	0.00000625
24	0.004	0.012	0.006	0.000004
25	0.004	0.012	0.006	0.000004
26	0.004	0.013	0.0065	0.00000625
27	0.004	0.013	0.0065	0.00000625
28	0.004	0.012	0.006	0.000004
29	0.004	0.012	0.006	0.000004
30	0.004	0.012	0.006	0.000004
31	0.004	0.012	0.006	0.000004
32	0.004	0.013	0.0065	0.00000625
33	0.004	0.014	0.007	0.000009

34	0.004	0.012	0.006	0.000004
35	0.004	0.012	0.006	0.000004
36	0.004	0.013	0.0065	0.00000625
37	0.004	0.012	0.006	0.000004
38	0.004	0.014	0.007	0.000009
39	0.006	0.014	0.007	0.000001
40	0.004	0.013	0.0065	0.00000625
41	0.004	0.012	0.006	0.000004
42	0.004	0.013	0.0065	0.00000625
43	0.004	0.012	0.006	0.000004
44	0.004	0.014	0.007	0.000009
45	0.004	0.013	0.0065	0.00000625
46	0.004	0.013	0.0065	0.00000625
47	0.004	0.013	0.0065	0.00000625
48	0.004	0.013	0.0065	0.00000625
49	0.006	0.016	0.008	0.000004
50	0.004	0.013	0.0065	0.00000625

Figure 3 visually depicts the data presented in Table 1, illustrating the runtime comparison between the TaxaTreeMapper and Fitch algorithms for each of the 50 phylogenetic trees sampled. The graphical representation allows for an immediate visual grasp of the runtime dynamics where TaxaTreeMapper consistently outperforms Fitch, as indicated by the shorter processing times.

V. CONCLUSIONS

The TaxaTreeMapper algorithm employs set theory to enhance accuracy and efficiency in processing phylogenetic trees. By integrating with associated datasets, it simplifies analysis and accurately identifies evolutionary features. It adeptly handles complex relationships and large datasets, providing outputs such as tree length and hypothetical taxa.

By condensing the ancestral state reconstruction into a single traversal from leaf to root, TaxaTreeMapper not only simplifies the computational process but also proves to be computationally twice as efficient as the Fitch algorithm. This remarkable increase in efficiency does not come at the cost of accuracy, with TaxaTreeMapper demonstrating a strong correlation with Fitch's results in identifying global minima. The foundational principles of TaxaTreeMapper emphasize streamlining phylogenetic analysis, making it especially advantageous for handling large datasets where computational resources are at a premium.

The TaxaTreeMapper algorithm offers an innovative approach that enhances efficiency and reduces complexity. Its ability to quickly and accurately construct phylogenetic trees represents a substantial leap forward from the traditional, more time-intensive methods.

Though TaxaTreeMapper may occasionally yield false positives due to its heuristic approach diverging from Fitch's conservative estimations, its overall computational efficiency and ability to quickly converge on global minima present a compelling advantage. In extensive phylogenetic analyses, where computational resources are constrained, TaxaTreeMapper's speed and general accuracy provide a favorable balance between performance and resource utilization.

Acknowledging differences between TaxaTreeMapper and Fitch, it's crucial to weigh overall performance metrics. TaxaTreeMapper's emphasis on efficiency and speed makes it valuable in high throughput phylogenetic analysis. Thus, considering its performance profile and correlation with Fitch's results, TaxaTreeMapper stands as a robust alternative, especially in scenarios requiring rapid tree length estimations.

REFERENCES

[1] J. D. Washburn, K. A. Bird, G. C. Conant, and J. C. Pires, "Convergent evolution and the origin of complex phenotypes in the age of systems biology," in *International Journal of Plant Sciences*, 2016, vol. 177, no. 4, pp. 305–318.

[2] R. N. Randall, C. E. Radford, K. A. Roof, D. K. Natarajan, and E. A. Gaucher, "An experimental phylogeny to benchmark ancestral sequence reconstruction," in *Nature communications*, 2016, vol. 7, no. 1, p. 12 847.

[3] S. Nayak and M. W. Otte, "Bidirectional sampling-based motion planning without two-point boundary value solution," in *IEEE Transactions on Robotics*, 2022, vol. 38, no. 6, pp. 3636–3654.

[4] R. E. I. Pardede, L. L. Tóth, G. Hosszú, and F. Kovács, "Glyph Identification Based on Topological Analysis," in *Book Glyph Identification Based on Topological Analysis*, 2012, pp. 99–103.

[5] R. E. Pardede, L. L. Tóth, G. A. Jeney, F. Kovács, and G. Hosszú, "Four-layer grapheme model for computational paleography," in *Journal of Information Technology Research (JITR)*, 2016, vol. 9, no. 4, pp. 64–82.

[6] G. L. Hosszú, Scriptinformatics, in Book Scriptinformatics, Nap Kiadó, 2021.

[7] G. Hosszú, "A novel computerized paleographical method for determining the evolution of graphemes" in *Encyclopedia of Information Science and Technology, Third Edition* (IGI Global, 2015), pp. 2017–2031.

[8] L. L. Tóth, R. E. I. Pardede, G. A. Jeney, F. Kovács, and G. Hosszú, "Application of the cluster analysis in computational paleography" in *Handbook of Research on Advanced Computational Techniques for Simulation-Based Engineering* (IGI Global, 2016), pp. 525–543.

[9] S. Daggumati and P.Z. Revesz, "Convolutional Neural Networks Analysis Reveals Three Possible Sources of Bronze Age Writings between Greece and India," in *Information*, 2023, vol. 14, no. 4, p. 227.

[10] O. A. Salman, G. Hosszú, and F. Kovács, "A new feature selection algorithm for evolutionary analysis of Aramaic and Arabic script variants," in *International Journal of Intelligent Engineering Informatics*, 2022, vol. 10, no. 4, pp. 313–331.

[11] O. A. Salman and G. Hosszú, "Cladistic Analysis of the Evolution of Some Aramaic and Arabic Script Varieties," in *International Journal of Applied Evolutionary Computation (IJAEC)*, 2021, vol. 12, no. 4, pp. 18–38.

[12] O. A. Salman and G. Hosszú, "A Phenetic Approach to Selected Variants of Arabic and Aramaic Scripts," in International Journal of Data Analytics (IJDA), 2022, vol. 3, no. 1, pp. 1–23.

[13] L. L. Cavalli-Sforza, A. Piazza, P. Menozzi, and J. Mountain, "Reconstruction of human evolution: bringing together genetic, archaeological, and linguistic data," in *Proceedings of the National Academy of Sciences*, 1988, vol. 85, no. 16, pp. 6002–6006.

[14] P. D. Ashlock, "The uses of cladistics," in *Annual Review of Ecology and Systematics*, 1974, vol. 5, no. 1, pp. 81-99.
[15] I. J. Kitching, Cladistics: the theory and practice of parsimony analysis, (Oxford University Press, USA, 1998).
[16] C. Semple and M. Steel, Phylogenetics, (Oxford University Press on Demand, 2003).
[17] W. C. Wheeler, Systematics: a course of lectures, (John Wiley & Sons, 2012).
[18] E. O. Wiley and B. S. Lieberman, Phylogenetics: theory and practice of phylogenetic systematics, (John Wiley & Sons, 2011).
[19] J. S. Farris, "Methods for computing Wagner trees," in *Systematic Biology,* 1970, vol. 19, no. 1, pp. 83–92.
[20] E. J. Henley and R. A. Williams, Graph Theory in Modern Engineering: Computer Aided Design, Optimization, Reliability Analysis, (Academic Press, Inc., 1973).
[21] G. Ganapathy, V. Ramachandran, and T. Warnow, "Better hillclimbing searches for parsimony," in *Better hill-climbing searches for parsimony* (Springer, 2003), pp. 245–258.
[22] N. Saitou and M. Nei, "The neighbor-joining method: a new method for reconstructing phylogenetic trees," in *Molecular biology and evolution,* 1987, vol. 4, no. 4, pp. 406–425.
[23] J. A. Cavender and J. Felsenstein, "Invariants of phylogenies in a simple case with discrete states," in *Journal of classification*, 1987, vol. 4, pp. 57–71.
[24] G. F. Estabrook, F. McMorris, and C. A. Meacham, "Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units," in *Systematic Zoology,* 1985, vol. 34, no. 2, pp. 193–200.
[25] W. J. L. Quesne, "A method of selection of characters in numerical taxonomy," in *Systematic Zoology*, 1969, vol. 18, no. 2, pp. 201–205.
[26] B. Sturmfels and S. Sullivant, "Toric ideals of phylogenetic invariants," in *Journal of Computational Biology*, 2005, vol. 12, no. 4, pp. 457–481.
[27] D. H. Huson and D. Bryant, "Application of phylogenetic networks in evolutionary studies," in *Molecular biology and evolution*, 2006, vol. 23, no. 2, pp. 254–267.
[28] W. M. Fitch, "Toward defining the course of evolution: minimum change for a specific tree topology," in *Systematic Biology*, 1971, vol. 20, no. 4, pp. 406–416.
[29] W. M. Fitch and E. Margoliash, "Construction of phylogenetic trees: a method based on mutation distances as estimated from cytochrome c sequences is of general applicability," in *Science,* 1967, vol. 155, no. 3760, pp. 279–284.
[30] J. Felsenstein, Inferring phylogenies. Sunderland, Massachusetts: Sinauer Associates, 2004.
[31] D. Penny, M.D. Hendy, and M.A. Steel, "Progress with methods for constructing evolutionary trees," in *Trends in ecology & evolution,* 1992, vol. 7, no. 3, pp. 73–79.
[32] A. Purvis and T. Garland, "Polytomies in comparative analyses of continuous characters," in *Systematic Biology*, 1993, vol. 42, no. 4, pp. 569–575.
[33] G. Sferra, F. Fratini, M. Ponzi, and E. Pizzi, "Phylo_dCor: distance correlation as a novel metric for phylogenetic profiling," in *BMC bioinformatics,* 2017, vol. 18, no. 1, pp. 1–7.

Osama A. Salman completed his Bachelor's and Master's degrees in Computer Science at the University of Baghdad, in 2014 and 2018, respectively. He is currently a fourth-year doctoral student at the Budapest University of Technology and Economics (BME), Faculty of Electrical Engineering and Informatics.
The focus of his PhD research is on conducting a phylogenetic analysis of the evolution of writing systems including Scriptinformatics, Machine learning applications in phylogenetics, Computational biology, Data analytics and Data processing in phylogenetics. In MCS His research examines into Machine Learning, Neural Networks, and Deep Learning with a special emphasis on Behavioral Biometrics, and Data Science applications. His academic journey is marked by a strong foundation in both theoretical concepts and practical applications.

Gábor Hosszú received his M.E. degree in Electrical Engineering and his Ph.D. in Technical Sciences from the Budapest University of Technology and Economics in 1992. He also obtained an MSc in Law from Pázmány Péter Catholic University, Budapest in 2011. Currently, he is an associate professor at the Faculty of Electrical Engineering and Informatics at the Budapest University of Technology and Economics, where he has been a member since 1990. In 2013, he was awarded the title of Dr habil. in recognition of his contributions. His main activities involve statistical evaluation of bioelectronic signals and research on pattern evolution, including scriptinformatics. He has published over 250 technical papers.

Towards developing a framework for automated accessibility evaluation of web content from expert perspectives

Jinat Ara[1], and Cecilia Sik-Lanyi [1,2]

***Abstract*—The current set of web accessibility evaluation tools requires a certain specification of information that requires user or expert perspectives. To improve the correctness and effectiveness of the evaluated result, expert perspectives can lead to great success, especially for the information that requires great effort, knowledge, and broadening research to set their determinator. Also, from the literature, not much effort is being observed to develop solutions for web accessibility evaluation addressing expert perspectives. Besides, the correctness of the evaluation report also depends on the used methods and technologies. Thus, consideration of advanced techniques might improve the performance of the assessment report. Therefore, in this paper, we aim to propose a framework to evaluate the accessibility of web content considering several evaluation criteria from expert perspectives considering several advanced techniques specifically Artificial Intelligence (AI) techniques. The proposed framework includes fifteen criteria that we obtained from consulting web experts and researchers that have a great effect on assessing the accessibility from the user's point of view. The proposed methodology evaluates accessibility following three phases: (a) identification of evaluation criteria from expert perspectives, (b) execution of the web accessibility evaluation process involving different evaluation algorithms incorporating different AI techniques, and (c) validate the framework through experimental and user-centric study to follow-up its computational ability. The proposed method is dynamic in nature and can be applied to different platforms to evaluate multiple web pages.**

***Index Terms*—Web accessibility evaluation, algorithmic evaluation, automated evaluation, user-centric design, social inclusion.**

I. INTRODUCTION

With the rapid growth of digital opportunities, interconnecting and processing information from web platforms (e.g., webpages) is becoming a common aspect of our daily activities. However, WebAIM reported that in 2023, across the world, 96.3% of webpages do not ensure full accessibility as it is quite difficult to offer a completely accessible platform that requires great effort and careful observation [1]. With this in mind, many studies addressed the importance of accessibility needs that should be focused on in the development stage for providing complete support of accessibility criteria [2].

[1] Department of Electrical Engineering and Information System, University of Pannonia, Veszprem, Hungary.

[2] Hungarian Research Network, Budapest, Hungary. (E-mail: {jinat.ara@mik.uni-pannon.hu, lanyi.cecilia@mik.uni-pannon.hu)

From this perspective, they suggested incorporating the Web Content Accessibility Guideline (WCAG) as it has a great role in facilitating the evaluation process of web content. Besides, it is a valuable resource to identify many aspects of the web that are relatively impossible to detect without having proper guidelines for the end users. However, along with numerous potentialities of WCAG, studies reported that WCAG does not cover every aspect that may cause accessibility issues [3]. This is might happen as the web are dynamic platform and it's continuously changing, also developers are injecting several advanced prototypes into the web. Therefore, after a long-term debate, ongoing research, and enormous effort, many web researchers concluded that consideration of additional criteria along with WCAG might be a wise decision to improve and make the web content accessibility evaluation result reliable and effective, similar to an approach proposed by Josefin Carlbring [4].

Focusing on this particular aspect, we evaluated several recent studies from the state-of-the-art literature (can be found in section 2) and concluded that none of the recent existing studies consider additional criteria from user or expert perspectives to evaluate the accessibility of web platforms. Most of them focus only on usability and accessibility criteria from various platforms such as multiple guidelines or standards including aesthetic design, cognitive load, etc. but do not really focus on the user or expert perspective. User or expert perspectives play a vital role in identifying some additional criteria from their personal experience that could be helpful for accessibility evaluation [5,6]. Besides, to implement the selected criteria in real-life applications, an advanced and improved web accessibility testing tool is an emerging need as successfully implementing these criteria requires advanced and updated techniques. In the previous literature, several approaches have already implemented different methods and techniques to evaluate several accessibility criteria such as ontology modeling [7], agile methods [8], variable magnitude approach [9], etc. However, in some cases, these techniques are not efficient enough to improve the performance of the developed tool. Some recent studies addressed this issue and enhanced the importance of incorporating several Artificial Intelligence (AI) techniques in implementing and evaluating the accessibility criteria that could bring some great and significant outcomes to contribute

DOI: 10.36244/ICJ.2025.5.3

to the accessibility domain of web platforms [10-12]. Addressing this manner, in this paper, we have considered expert opinion or perspectives as an important factor, and selected 15 major attributes for evaluating webpage accessibility through our proposed framework where the proposed framework is developed considering several AI techniques, specifically NLP methods and some auxiliary functions.

The main aim of this paper is to contribute to the accessibility perspective of digital platforms by proposing an automated web accessibility evaluation framework incorporating AI techniques to determine webpage accessibility according to the additional evaluation criteria from expert perspectives. The proposed system is dynamic in nature, it can be integrated or implemented for any webpage evaluation considering the webpage URL as input to process and generate the result. The prime contributions of this research work are listed as follows:

- In our proposed framework, fifteen key attributes are considered that are beyond web content accessibility guidelines related to the web page's arbitrary information and content information.
- To validate the proposed framework, an experimental evaluation has been performed considering 15 healthcare webpages from Hungary.
- Along with this, we have conducted a questionnaire-based evaluation to evaluate their accessibility considering the selected 15 attributes for the same webpages that we used to evaluate through our proposed framework.
- Finally, we conclude the accessibility status of the evaluated webpages with identified issues that require additional consideration in the future to improve the accessibility of the tested webpages.

This paper is organized as follows. Section 2 provides a brief review of related studies that contributed recently to this field. Section 3 discusses the methodology of the proposed framework by demonstrating the system architecture, design, development, implementation, and validation strategies in detail. Section 4 provides a detailed discussion. Finally, Section 5 concludes the paper with some recommendations and future directions.

II. RELATED STUDIES

Web Content Accessibility Guideline (WCAG) was initiated for the advancement of public and private sector practitioners to direct them about some extensive criteria related to the design and development of web platforms in order to motivate designers and developers to implement such criteria to ensure complete access opportunities for the people with disabilities [13,14]. However, mainly, the focus of WCAG is on the technical artifact of the web, but not on users' and experts' perspectives. This means that the conformance of WCAG is technical-oriented rather than evaluating user experience, more particularly, the needs of people with specific needs. Therefore, we argue that this could be a contributing factor to reducing accessibility opportunities from real-life facts.

Recently, there have been several studies conducted by considering accessibility as the prime resource to assess issues with web navigation. For example, Bigham et al. [15] concluded several crowdsourced methods that contribute to web accessing barriers. Unfortunately, they concluded that though these methods have effectiveness in accessibility issues evaluation, most of them were a particular guideline-specific approach. Abhirup Sinha [16] evaluated several Indian web pages regarding web content accessibility guidelines incorporating several automated accessibility testing tools. They concluded their findings by highlighting the importance of accessibility improvements. In another study, a similar approach was also conducted by Parmanto and Zeng [17] by proposing their custom accessibility evaluation metrics (called WAB score). Also, Miranda and Araujo [18] proposed a framework to support in analysis, development, and validation of accessibility requirements following the agile routine. Their prime objective is to improve the specification and demonstrate accessibility requirements through a goal-oriented model according to the WCAG. In another study, Alzahrani, and Al-Aama [19] proposed a framework, namely, the social media accessibility framework (SMAF), to evaluate the accessibility of social media platforms specifically for people with hearing and visual disabilities. Their evaluation showed that the ratio of accessibility was noticeable when the guidelines were followed during the development. Besides, few studies focused on guidelines from the Americans with Disabilities Act (ADA). Fichtner and Strader [20] stated that to make a website accessible according to ADA compliance, developers need to incorporate advanced tools and at the same time, they need adequate training to design accessible websites effectively.

However, considering user requirements or expert perspectives, a limited number of studies have been noticed in the literature. In one of the studies, Koutsabasis et al. [21] proposed a web accessibility evaluation framework considering user requirements related to colors, style sheets, and images. Another study conducted by Akgül et al. [22] evaluated web platforms considering the user requirements related to accessibility, usability, readability, and security issues.

These user-centric approaches have a limited number of user criteria in consideration as issues with accessibility tend to evaluate every aspect of web objects including arbitrary information, structural objects, and visual aspects. Besides, along with user criteria, regarding the technological aspects, most of the related works developed their framework considering some traditional methods and techniques such as ontological model, heuristic model, agile techniques, etc. which is also a crucial issue that needs to be focused to improve the performance of the developed model or generated reports.

Addressing all of these issues, first, we have conducted an extensive study involving experts to identify the additional criteria that are beyond WCAG criteria and crucial to improving the accessibility of the web platform. Later, we validated webpages using the selected criteria through our

proposed framework where we considered several AI techniques that have a great contribution to identifying the accessibility status of the tested webpages in terms of the selected criteria and improving the performance of the evaluation method.

III. Material and Methods

This section is structured by demonstrating the selected evaluation criteria, and the proposed framework with clarifying its design and development process. Also, the proposed framework has been validated through an experimental process where we experimented and evaluated a bunch of samples of webpages.

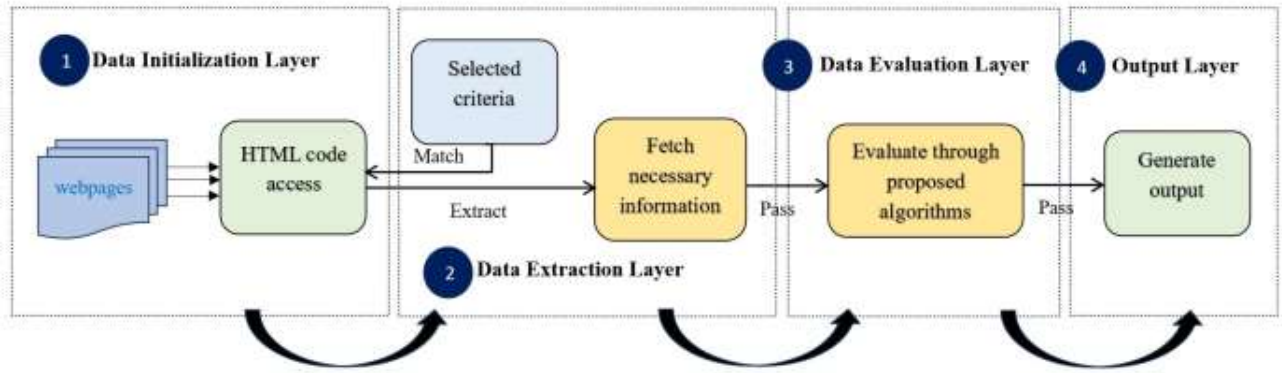


Fig. 1: The System Architecture of the proposed model

A. Evaluation Criteria Selection

As literature supported that web content accessibility guidelines could not support every aspect related to accessibility, thus our prime focus in this work is to identify what are the additional criteria that could be effective in facilitating the evaluation process along with WCAG. Addressing this issue, we conducted an expert study where we interviewed five experts and asked their suggestions about the possible potential additional criteria that weren't mentioned in the WCAG and might be valuable to incorporate into the evaluation process. All of the experts were from the Department of Electrical Engineering and Information Systems, University of Pannonia, Veszprem, Hungary. Three experts have more than 20 years of experience in the accessibility of digital platforms and others have more than 5 years of experience in this field. Based on their feedback, we identified 15 criteria that could be used as additional criteria and might be effective for facilitating the website accessibility evaluation. The selected 15 criteria are related to two distinctive aspects such as arbitrary information {sever status; webpage loading time; and webpage length}, and the content information {paragraph length; the ratio of Hyperlinks; webpage default language; user information; CAPTCHA; multiple language options; image ratio; text font family; text font size; text pattern; content type; audio/video content ratio}. All of these aspects have been analyzed through several criteria using three separate algorithms. The whole automated evaluation process has been described in detail in the following sections.

B. System Architecture, Design, and Development

The proposed framework has four distinctive layers as shown in Figure 1. The first layer is responsible for data initialization, the second layer is responsible for data extraction, the third layer is for extracted data evaluation through algorithmic observation and the final layer is responsible for output representation. All of these layers are described in detail in the following subsections:

a) Data Initialization Layer: The data initialization layer performs the tasks of accessing the HTML code of the tested webpage via the URL of the page. We used sublime text editor as a development framework and Python programming language to write the script. For HTML code access, we used an HTML parser which parses the HTML code and facilitates the data extraction process. We used an HTML parser as HTML source code represents objects referring to several tags, elements, and attributes which are considered as unstructured or semi-structured elements or information. Considering this large number of unstructured information, it is quite difficult to perform the evaluation process effectively. Thus, to make this unstructured information into a structured format, we used an HTML parser which extracts information from HTML source code in a tree view format with a structural manner. As an HTML parser, we used [1]Beautiful Soup which is a Python package that allows us to access any HTML or XML documents. After initializing the data, it redirects its output into the data extraction layer.

[1] https://en.wikipedia.org/wiki/Beautiful_Soup_(HTML_parser)

b) Data Extraction Layer: The data extraction layer performs the extracting process of all the necessary data or information from the HTML tree view or HTML source code of the tested website. We extracted information according to our selected criteria that have been selected according to the expert opinion. Under the selected criteria, 15 key attributes have been decided to be included in this study. To determine the selected criteria in the HTML code and extract the related information, we perform a simple matching function that matches all the criteria in terms of tags, elements, and attributes in a hierarchical manner. Upon matching the criteria, we extracted or fetched their corresponding information and passed the information to the next layer to evaluate through an algorithmic observation.

c) Data Evaluation Layer: In general, the data evaluation layer is responsible for conducting the algorithmic evaluation process incorporating several auxiliary methods to determine the accessibility issues of the tested webpage. The whole evaluation process is performed using three different algorithms stated in Algorithms 1–3 where algorithm 1 is for evaluating webpage arbitrary information, algorithm 2 is for evaluating webpage content information, and algorithm 3 is for overall score computation and accessibility status specification.

Algorithm 1: Algorithm for webpage arbitrary information.

Input: Webpage URL
Output: **A**rbitrary information score

1. counter = 0;
2. initialize the webpage through URL;
3. load the webpage through urllib.request.urlopen (URL);
4. parse the HTML code using BeautifulSoup parser

//validating webpage activation status

5. read the responses through requests.get(URL) function;
6. if (response.status_code == 200),
7. *webpage_activation_score*=counter++;

//calculate the webpage loading time

8. calculate the start time = time.time ();
9. read the responses through requests.get (URL) function;
10. calculate the end time = time.time ();
11. calculate the loading time = end time – start time;
12. if (loading time <=0.3 sec),
13. *webpage_loadingTime_score*=counter++;

//calculate page length

14. calculate page length in byte using (len(urllib.request.urlopen (URL).read())) function;
15. convert page length into byte to KB by dividing 1024;
16. if (page length <=14 kb),
17. *webpage_length_score*=counter++;
18. calculate *score_of_arbitrary_information* = {*webpage_activation_score* + *webpage_loadingTime_score* + *webpage_length_score*}

Algorithm 1 evaluates arbitrary information on the tested webpage by calculating the webpage's active status, loading time, and webpage length. Initially, it performs by loading the webpage and extracting the HTML code of a given URL using the BeautifulSoup python library (lines: 1-4). Upon accessing the webpage resources, it evaluates the arbitrary information. To evaluate the active status of the tested webpage, we tracked the responses of the loaded webpage and according to the response status code, we determined their activation/deactivation status (lines: 5-7). To evaluate webpage loading time, we calculated loading time by tracking the start and end times. After tracking the start time, it continues to read all the responses that are redirected from the page, as well as the end time, and uses it to calculate the overall loading time (lines: 8-13). The loading time is calculated using the difference between the start and end times. Finally, we calculated the page length into bytes and then converted it into kilobytes to evaluate their preferable length (lines: 14-17). Finally, it calculates the score of arbitrary information using the arbitrary information score calculation formula in line 18.

Algorithm 2: Algorithm for webpage content information.

Input: Webpage URL
Output: Content information score

1. counter = 0;
2. initialize the webpage through URL;
3. load the webpage by sending request through urllib.request.urlopen (URL);
4. parse the HTML code using BeautifulSoup parser

//validating webpage text length

5. calculate the length of the textual content by words;
6. if word count <=1500,
7. *webpage_textLenght_score*=counter++;

//validating webpage hyperlinks ratio

8. count all the hyperlinks;
9. if hyperlinks count <=50,
10. *webpage_hyperlinks_score*=counter++;

//validating webpage default language

11. identify the default language;
12. if language is ("en" or "en-US" or "en-GB"),
13. *webpage_language_score*=counter++;

//validating webpage required user information

14. check the required login information;
15. if no ('Username' and 'Password') is required,
16. *webpage_userInformation_score*=counter++;

//validating webpage CAPTCHA

17. identify the 'captchaBlock' in div element;
18. if (‘id’! = 'captchaBlock'),
19. *webpage_CAPTCHA_score*= counter++;

//validating webpage language changing option

20. check language option through (‘nav’, ‘ul’, ‘li’, ‘a’) elements;
21. if ('onclick' is active),
22. *webpage_languageOption_score*=counter++;

//validating webpage image ratio

23. count all the images;
24. if (image count >10),
25. *webpage_image_score*=counter++;

//validating webpage audio/video ratio

26. count all the audio and video content;
27. if (audio/video count is between >=1 to <=2),

28. *webpage_audioVideo_score*=counter++;
//validating webpage font type
29. check the font family using style element,
30. if font family is (Tahoma/Calibri/Helvetica/Arial/Verdana/Times New Roman),
31. *webpage_fontType_score*=counter++;
//validating webpage font size
32. check the font size in pixel using style element;
33. if font size is (16px/17px/18px/19px/20px),
34. *webpage_fontSize_score*=counter++;
//validating webpage text pattern
35. check the text pattern;
36. if (b, strong, i, em, mark, sub, sup) pattern is not in text,
37. *webpage_textPattern_score*=counter++;
//validating webpage content type
38. check the content type;
39. if (text/image/video) content is identified,
40. *webpage_contentType_score*=counter++;
41. calculate *score_of_content_information* = {*webpage_textLenght_score* + *webpage_hyperlinks_score* + *webpage_language_score*+ *webpage_userInformation_score* +*webpage_CAPTCHA_score*+*webpage_languageOption_score*+*webpage_image_score*+*webpage_audioVideo_score*+*webpage_fontType_score*+*webpage_fontSize_score*+*webpage_textPattern_score*+*webpage_contentType_score*}

Algorithm 2 demonstrates the evaluation of webpage content information where we considered twelve attributes such as texts, hyperlinks, language, required user information, CAPTCHA, language option, images, audio/video, font style, font size, text pattern, and content type. Similar to algorithm 1, at first it loads the webpage through the given URL to parse the HTML code through BeautifulSoup python Library (lines: 1-4). Upon accessing the HTML code, the algorithm checks each selected attribute and evaluates them according to the determined criteria to validate their status in terms of accessibility considering several auxiliary functions and Natural Language Processing (NLP) techniques. In lines: 5-7, the webpage text length is evaluated by counting words and if the counted number is under the determined condition, the accessibility status is marked as successful, and increase the counter number. Also, a similar approach has been used for other selected attributes to evaluate their accessibility status (lines: 8-40). Finally, it calculates the accessibility score of content information by summing the score of each evaluated attribute, in line 41.

Algorithm 3: Algorithm for accessibility score computation.

Input: Webpage URL
Output: Overall accessibility score

1. retrieve the 15 types of attributes selected for accessibility evaluation;
2. calculate score of each attribute;
3. calculate accessibility score (*score_arbitrary_informan*+*score_content_information)/N*;

Algorithm 3 calculates the accessibility score of the given webpage URL. First, it considers 15 types of attributes and calculates the score of each attribute incorporating Algorithm 1 and Algorithm 2 (lines 1-2). After calculating all the attributes scores, the accessibility score per webpage is calculated by summing their score using the accessibility score calculation formula, in line 3.

d) Output Layer: The output layer provides the analysis results considering each attribute with their evaluation status (Passed, Failed, Not Tested, and Not Detected), identified issues, and future improvement suggestions. Besides it provides the computed overall accessibility score and overall accessibility status based on the overall accessibility score where we consider several ranges such as if accessibility score is >=90 then Completely Accessible; if accessibility score is <90 to >=75 then Comparatively Accessible; if accessibility score is <75 to >=55 then Partially Accessible; and if accessibility score is <55 then Slightly Accessible.

C. Implementation and Validation

In this section, we presented our implementation by experimenting with 15 selected web pages and validated the evaluation result performing a questionnaire-based user study to represent the effectiveness of the proposed framework for accessibility evaluation of web contents.

TABLE I
TESTED WEBPAGES WITH THEIR EVALUATED SCORE AND ACCESSIBILITY STATUS

Web ID	Webpage URLs	Accessibility score	Accessibility status
Web1	https://klinikaikozpont.unideb.hu/en/node	53.33 %	Slightly Accessible
Web2	https://szkt.hu/en/	40.0%	Slightly Accessible
Web3	https://eegeszsegugy.gov.hu/web/eeszt-information-portal/home	33.33%	Slightly Accessible
Web4	https://www.bazmkorhaz.hu/	41.52%	Slightly Accessible
Web5	http://www.sopronkorhaz.hu/	35.33%	Slightly Accessible
Web6	https://petz.gyor.hu/	40.08%	Slightly Accessible
Web7	https://csfk.hu/	30.02%	Slightly Accessible
Web8	https://www.mfkh.hu/	43.08%	Slightly Accessible
Web9	https://onkol.hu/	46.66%	Slightly Accessible
Web10	https://www.uzsoki.hu/	39.05%	Slightly Accessible
Web11	https://bhc.hu/en/kedvezmenyek	33.33%	Slightly Accessible
Web12	https://wmc.hu/en/	35.59%	Slightly Accessible
Web13	http://heimpalkorhaz.hu/	49.66%	Slightly Accessible
Web14	https://delpestikorhaz.hu/	30.36%	Slightly Accessible
Web15	https://kk.pte.hu/klinikak-intezetek	32.71%	Slightly Accessible

a) Proposed framework implementation: To implement the proposed framework, we experimented by validating fifteen healthcare webpages (hospital and medical point) from Hungary that are listed in Table 1. Also, in Table 1, we presented the evaluation result in terms of their computed accessibility score (by applying three algorithms described in section B) with their accessibility status that has been classified according to the statistics described earlier (subsection *B (d)*).

Table 1 depicts that none of the tested web pages was found accessible in terms of the selected evaluation criteria. All the tested webpages found as slightly accessible that indicate none of the webpages followed all the selected criteria and, in this regard, all of the pages have serious issues with accessibility. Additionally, we found some issues that were frequently observed in the majority of the tested webpages such as issues with 'webpage loading time', 'hyperlink ratio', 'webpage length', 'webpage default language', 'language changing option', 'font type', 'font-size', and 'webpage content type' that need to be considered in future to improve accessibility.

b) Proposed framework validation: To validate the proposed framework, we incorporated end users to evaluate webpages and provide their feedback in terms of our asked questions where the questions asked were related to our identified 15 criteria. To perform the user study, we invited participants to attend online participation via Zoom meeting. All the participants were university bachelor's and master's students from the Electrical Engineering and Information Systems Department of the University of Pannonia, Hungary. The total number of participants was 20, including 8 female and 12 male students aged between 21 and 25. All of them have sufficient knowledge about 'web programming', and 'User Interface Design'. To make the evaluation process effective, first, we briefly explained the aim, and testing process and described each question to the participants which took around 10 minutes. All the questions were designed in such a way as to understand the user perspective properly. After explaining everything to the participants, we shared the resources with users including the Google questionnaire link, and the website information in a shared file that needs to be evaluated. On average, the experiment took 20 to 30 minutes. The questionnaire used for the user feedback is shown in the following.

Q1: Does the webpage's loading time satisfactory? (Yes/No), please clarify your answer.

Q2: Is the paragraph or textual content length of the webpage satisfactory? (Yes/No), please clarify your answer.

Q3: Is the webpage's ratio of hyperlinks satisfactory? (Yes/No), please clarify your answer.

Q4: Is there a default English version of the webpage? (Yes/No), if yes/no, do you think it's useful? (Yes/No), clarify your answer.

Q5: Is the length of the webpage satisfactory? (Yes/No), please clarify your answer.

Q6: Is the server of the webpage active? (Yes/No), if yes/no, do you think it's useful? (Yes/No), clarify your answer.

Q7: Does webpage require user information to access? (Yes/No), if yes/no, do you think it's useful? (Yes/No), clarify your answer.

Q8: Does webpage use CAPTCHA? (Yes/No), if yes/no, do you think it's useful? (Yes/No), clarify your answer.

Q9: Does webpage have a multiple-language option? (Yes/No), if yes/no, do you think it's useful? (Yes/No), clarify your answer.

Q10: Is the ratio of image content satisfactory? (Yes/No), please clarify your answer.

Q11: Is the font used on webpages understandable? (Yes/No); if not, please clarify your answer.

Q12: Is the font size on webpage satisfactory? (Yes/No); if not, please clarify your answer.

Q13: Does webpage use multiple text patterns? (Yes/No), if yes/no, do you think it's useful? (Yes/No), clarify your answer.

Q14: Does webpage contain multiple content types (e.g., audio/video/text/images)? (Yes/No), if yes/no, do you think it's useful? (Yes/No), clarify your answer.

Q15: Is the ratio of audio/video content on the webpage satisfactory? (Yes/No), please clarify your answer.

Each participant's responses to the asked questions are listed in Table 2. This table shows that from the overall feedback for each of the questions, the majority of the user answers were very poor in terms of positive responses. It depicts that the majority of the users were not satisfied while they navigated the web pages to respond to the asked questions. It directs the emerging need to consider the addressed criteria in this paper in the web accessibility evaluation process as these criteria have a great impact on improving accessibility.

TABLE II
ASSESSMENT QUESTIONNAIRE WITH PARTICIPANT'S RESPONSES

Questionnaire	Yes (%)	No (%)
Q1: Does the webpage's loading time satisfactory?	51.38	48.62
Is the paragraph or textual content length of the webpage satisfactory?	30.32	69.68
Q3: Is the webpage's ratio of hyperlinks satisfactory?	20.0	80.0
Q4: Is there a default English version of the webpage?	8.7	91.30
Q5: Is the length of the webpage satisfactory?	60.05	39.95
Q6: Is the server of the webpage active?	92.15	7.85
Q7: Does webpage require user information to access?	9.48	90.52
Q8: Does webpage use CAPTCHA?	5.0	95.0
Q9: Does webpage have a multiple-language option?	9.53	90.47
Q10: Is the ratio of image content satisfactory?	64.71	35.29
Q11: Is the font used on webpages understandable?	31.58	68.42

Q12: Is the font size on webpage satisfactory?	42.11	57.89
Q13: Does webpage use multiple text patterns?	20.3	79.7
Q14: Does webpage contain multiple content types (e.g., audio/video/text/images)?	10.65	89.35
Q15: Is the ratio of audio/video content on the webpage satisfactory?	76.17	23.83

However, the work presented in this paper is a part of our detailed research regarding web accessibility. From the analysis, we can conclude that there is a huge scope to improve the accessibility of web platforms. As the web platform act is an important medium to access a wide array of information, thus this platform should be well designed along with future improvements, considering improving webpage loading time, properly maintaining hyperlink ratio, reducing webpage length, providing webpage default language, and language changing option, and ensuring proper font type, font size, and webpage content type. We believe that this study might help website designers, developers, and future researchers to enhance their contribution to their developed sites to a large extent. Also, we believe more future studies regarding this context could bring some new perspectives and motivate the practitioners to put their attention broadly.

IV. Discussion

In general, web content accessibility guidelines, for example, Web Content Accessibility Guideline (WCAG) is one of the widely accepted standards but few special objects are not included in this standard that could raise accessibility issues associated with people with disabilities. For example, almost every webpage has no manual text size or color adjustment option which raises issues for people with vision disability or color disabilities in navigating the content. Sometimes webpages require user information for accessing web content, and few webpages ask to pass through CAPTCHA testing which is considered a difficult task for people with special needs. Some other issues related to the excessive number of internal/external links, images, and video and audio content also hinder access opportunities for people with cognitive disability. Unfortunately, the most advanced and standard guidelines normally do not consider these aspects during their guideline specification. Therefore, according to the expert's opinion, considering these aspects as additional criteria along with any standard guideline might facilitate the webpage accessibility evaluation process to reveal the true insights of webpage accessibility. Besides, the proposed approach is an AI-driven approach that facilitates the evaluation process in terms of time, semantic improvements, and matching the web feature with specific guidelines compared to other solutions such as ontological modeling, agile modeling, or goal-oriented modeling.

With this aim, this work presents an automated web content accessibility evaluation framework that performs the evaluation considering different algorithmic evaluation considering AI techniques. According to the expert suggestion, in our proposed framework, we considered fifteen criteria that are crucial to incorporate into the webpage accessibility evaluation. To evaluate the web content regarding the selected criteria, we conducted an algorithmic evaluation which directed that the majority of the tested webpage has serious accessibility issues regarding the selected aspects. Also, from the user-centric study, the same scenario is reflected similarly to the evaluated result. Also, during the user study, we encouraged the participants to share their additional suggestions that might help to understand the most frequent issues they have experienced in the tested webpages. After analyzing all the shared opinions or suggestions, we categorized their responses under six factors. Therefore, according to the respondents' opinions, the tested webpages should have adequate focus on the following factors:

- hyperlink ratio
- webpage length
- webpage default language
- language changing option
- font type
- font-size
- webpage content type

Figure 2, the pie chart depicts that 20% of suggestions were related to the issues with font type and font size adjustment option; 17% of responders reported issues related to the webpage's default language and demand for specifying the English language as a default language to improve the accessibility of webpage; 15% responder suggested to add manual language changing option; 11% responder suggestion was related to issue with hyperlink ratio; and 9% and 8% responder were concerned about webpage length and webpage content type, respectively.

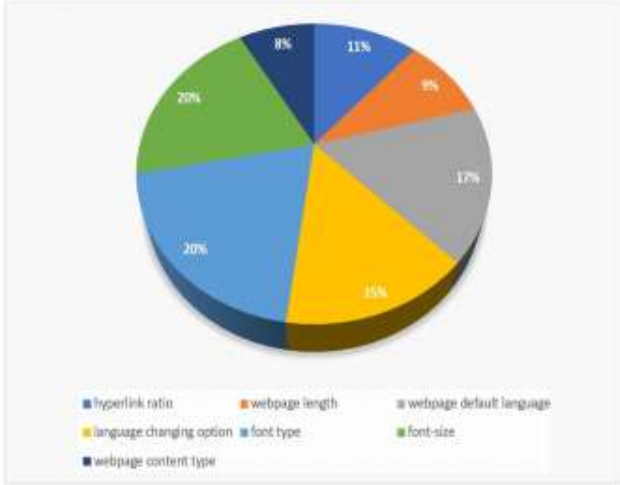


Fig. 2: Responders suggestion for evaluated criteria

V. Conclusion and Future Research

This paper presents an extensive study on the development of an accessibility evaluation framework to evaluate accessibility of the web content. As the ratio of the inaccessible web has increased dramatically, thus an updated and dynamic web evaluation tool is an emerging need.

Besides, to improve the performance of the evaluation tool, the importance of focusing on advanced techniques is significantly important. To address this concern, we proposed a framework that incorporates several AI techniques that make the development dynamic in nature and able to evaluate any webpage in terms of our selected 15 attributes. Our main challenge was implementing AI techniques to make it dynamic as different HTML structures have been used in different web page development. This proposed framework can act as a tool that can evaluate accessibility issues and generate accessibility scores for the tested webpage. Along with this, an experimental evaluation and questionnaire-based user validation have been performed which reveals that the proposed framework has significance in generating satisfactory results which also indicates the significance of AI-based web accessibility evaluation tools. However, our future work aligns with performing another extensive study involving the user to validate the outcome of this proposed approach and identify suitable criteria that need to be focused on in the web accessibility evaluation process further.

References

[1] 2023. WebAIM: The WebAIM Million - The 2023 Report on the Accessibility of the Top 1,000,000 Home Pages. https://webaim.org/projects/million/.

[2] Harper, S., & Chen, A. Q. (2012). Web accessibility guidelines: A lesson from the evolving Web. *World Wide Web*, 15, 61-88.

[3] Cooper, M., Sloan, D., Kelly, B., & Lewthwaite, S. (2012, April). A challenge to web accessibility metrics and guidelines: putting people and processes first. In *Proceedings of the international cross-disciplinary conference on Web accessibility* (pp. 1–4).

[4] Carlbring, J. (2020). Inclusive Design for Mobile Devices with WCAG and Attentional Resources in Mind: An investigation of the sufficiency of the Web Content Accessibility Guidelines when designing inclusively and the effects of limited attentional resources.

[5] Brajnik, G., Yesilada, Y., & Harper, S. (2011). The expertise effect on web accessibility evaluation methods. *Human–Computer Interaction*, 26(3), 246-283.

[6] Ara, J., Sik-Lanyi, C., & Kelemen, A. (2023). Accessibility engineering in web evaluation process: a systematic literature review. *Universal Access in the Information Society*, 1-34.

[7] Sapna, P. G., & Mohanty, H. (2011). An ontology based approach for test scenario management. In I*nformation Intelligence, Systems, Technology and Management: 5th International Conference, ICISTM* 2011, Gurgaon, India, March 10-12, 2011. Proceedings 5 (pp. 91–100). Springer Berlin Heidelberg.

[8] Sanchez-Gordon, S., & Luján-Mora, S. (2017, March). A method for accessibility testing of web applications in agile environments. In *Proceedings of the 7th World Congress for Software Quality (WCSQ)*. En proceso de publicación. (citado en la página 13, 15, 85) (p. 144).

[9] Ara, J., Sik-Lanyi, C., & Kelemen, A. (2023). An Integrated Variable-Magnitude Approach for Accessibility Evaluation of Healthcare Institute Web Pages. *Applied Sciences*, 13(2), 932.

[10] Abou-Zahra, S., Brewer, J., & Cooper, M. (2018, April). Artificial intelligence (AI) for web accessibility: Is conformance evaluation a way forward?. In *Proceedings of the 15th international web for all conference* (pp. 1–4).

[11] Draffan, E. A., Ding, C., Wald, M., Everett, H., Barrett, J., Sasikant, A., ... & Newman, R. (2020). Can a Web Accessibility Checker Be Enhanced by the Use of AI?. In Computers Helping People with Special Needs: 17th International Conference, ICCHP 2020, Lecco, Italy, September 9–11, 2020, Proceedings, Part I 17 (pp. 67–73). Springer International Publishing.

[12] ARAa, J., & Cecilia, S. L. (2022). Artificial intelligence in web accessibility: potentials and possible challenges. *Proceedings of IAC*, 173.

[13] Patvardhan, N., Ranade, M., & Patvardhan, K. (2023, February). Web Accessibility Supporting Diversity Inclusion and Effective Internet Communication in e-commerce, for Sustainable Social Development. In *2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON)* (pp. 1–5). IEEE.

[14] Ishaq, M. (2023). Improved Web Accessibility Evaluation of Open Learning Contents for Individuals with Learning Disabilities. *arXiv preprint arXiv:2306.10039*.

[15] Bigham, J., Ladner, R. and Borodin, Y. (2011). The design of Human-Powered Access Technology. *Proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility (ASSETS '11)*. New York: ACM Press.

[16] Sinha, A. (2020, September). Web accessibility analysis on government of India websites based on WCAG. In *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)* (pp. 1–7). IEEE.

[17] Parmanto, B., & Zeng, X. (2005). Metric for web accessibility evaluation. *Journal of the American Society for Information Science and Technology*, 56(13), 1394-1404.

[18] Miranda, D., & Araújo, J. (2020). A framework for integrating web accessibility requirements in agile methodologies. In *13th International Conference on the Quality of Information and Communications Technology (QUATIC)*.

[19] Alzahrani, R. K., & Al-Aama, A. Y. (2024). A framework to improve the accessibility of social media account content. *Universal Access in the Information Society*, 1-15.

[20] Fichtner, J. R., & Strader, T. J. (2024). An analysis of US website accessibility court cases: are standalone websites subject to ADA requirements?. *Universal Access in the Information Society*, 1-17.

[21] Koutsabasis, P., Vlachogiannis, E., & Darzentas, J. S. (2010). Beyond specifications: Towards a practical methodology for evaluating web accessibility. *Journal of Usability Studies*, 5(4), 157-171.

[22] Akgül, Y., Uymaz, A. O., & Baba, A. (2023). Accessibility, usability, readability, and security analysis of company websites.

Jinat Ara graduated from Southeast University, Bangladesh in 2018 with a bachelor's degree in Computer Science and Engineering (CSE). Following that she obtained her master's degree in 2020 from Jahangirnagar University, Bangladesh from the Department of Computer Science (CS). At the same time, she worked as a Software Engineer at Odyssey Apps, Dhaka, Bangladesh. Currently, she is pursuing her Ph.D. degree at the Faculty of Information Technology, University of Pannonia, Hungary. Her research area for her Ph.D. studies on the area of "accessibility investigation in web platforms or environments". Additionally, she has a keen interest in conducting cutting-edge research and currently, she is carrying around six years of research experience in several areas of computer science. Her area of research interest is 'Machine Learning (ML)', 'Virtual and Augmented Reality', 'Assistive Technologies', 'Web Technologies', 'Artificial Intelligent (AI)', 'Natural Language Processing (NLP)' and 'Automated Vehicles (AV)'.

Cecilia Sik-Lanyi is a Professor at University of Pannonia, Veszprem, Hungary. Sik-Lányi studied Mathematics and Computer Science. She obtained the degree of Dr. Univ. in Physical-chemistry (1993), and Ph.D. in Computer Science (2000), and became a Doctor of Hungarian Academy Sciences (D.Sc.) in 2022. Her research area is virtual reality, serious games, and user interface design, focusing on color science and accessibility. She received several awards, the most important ones are: "Diamond-Award from the Association for the Advancement of Assistive Technology in Europe" in 2015, and the "King Salman Award for Disability Research" in 2018. She has supervised more than 220 BSc and MSc theses since 1997. She has published more than 500 referred articles and conference papers and worked as a guest editor for many renowned journals and books.

Special Issue
of the Infocommunication Journal

Application of Process Discovery Methods for Learning Process Modeling

Erika Baksáné Varga, and Attila Baksa

Abstract—**Process mining encompasses a suite of techniques aimed at analyzing event data to gain insights and improve operational processes. One way of achieving this is to discover the driving process of the activities that occurred in a system. Technically, process discovery algorithms are used to transform an event log into a process model which is representative of the activities registered in the given system. This study explores the application of process discovery methods to better understand the learning processes in an introductory programming course for first-year Computer Science BSc students. A total of 52 practical problems were assigned as out-of-class activities via GitHub Classroom, resulting in 2789 commits from 59 students. These commits, along with the students' exam grades, were recorded in an object-centric event log, subsequently converted into a case-based log for analysis using the PM4Py program library.**

The study had two primary goals: first, to identify the characteristics of successful learning strategies by comparing process models of students who passed versus those who failed the programming exam; and second, to identify bottlenecks that hindered student progress. By employing the Heuristic Miner and Inductive Miner algorithms, we developed and contrasted learning process models, revealing significant patterns and obstacles within the educational process. The findings provide valuable insights into the factors that contribute to effective learning and suggest areas for enhancing our teaching methodologies.

Index Terms—**Process discovery, Learning process modeling, Application of PM4Py package**

I. Introduction

WITHIN the expansive domain of education, the significance of understanding how individuals learn and retain information cannot be overstated. This comprehension is critical for the development and implementation of instructional methods that are truly effective [1]. Historically, traditional teaching approaches have tended to adopt a one-size-fits-all methodology, which unfortunately overlooks the reality that each learner possesses unique preferences and cognitive processes. This oversight can result in learning experiences that are less than optimal, potentially hindering the development of individuals' skills and overall educational performance [2]. Recognizing the diversity in learning styles is thus essential for improving educational outcomes.

To enhance educational practices, it is imperative to identify and understand the processes that lead to successful learning. In this context, the field of process discovery emerges as a promising approach. Process discovery involves uncovering and analyzing the patterns and strategies employed by learners, providing insights that can be used to inform and improve teaching methodologies. By applying process discovery methods specifically to the study of learning procedures in programming, educators can gain a deeper understanding of student behavior. This approach allows instructors to identify and highlight problems that require additional iterations to be resolved, ultimately facilitating a more tailored and effective educational experience.

A. Motivation and Goals

The motivation behind this study lies in the need to better understand the learning behaviors of students in programming education. Programming presents unique challenges, such as mastering problem-solving and abstract thinking, which often leads to high variability in student performance. This study aims to explore how process discovery methods can reveal patterns of effective learning strategies and the obstacles faced by struggling students. By identifying these patterns, educators can improve their instructional methods, offering better support to students.

The study has two main goals: first, to compare the learning processes of students who passed versus those who failed the programming course, using process discovery techniques to reveal key differences in learning behaviors; and second, to identify bottlenecks that hinder student progress. These insights can guide future teaching practices by highlighting the areas where students struggle the most, enabling educators to adapt their teaching strategies accordingly.

B. Challenges and Novelty

Analyzing learning behaviors through event logs presents the challenge of handling diverse and noisy data, especially in educational settings where individual learning paths vary widely. This paper addresses these challenges by applying process discovery techniques, traditionally used in business process analysis, to model and analyze student learning in programming courses. The novelty lies in applying process mining tools – specifically *Heuristic Miner* and *Inductive Miner* – to educational data, which enables a more detailed and dynamic understanding of how students engage with programming tasks.

By capturing event log data from GitHub Classroom, this study provides a unique and fine-grained view of student behavior, going beyond traditional performance assessments. It is one of the first studies to apply process discovery techniques to programming education, thus offering new insights into student engagement and problem-solving strategies.

Authors are working at University of Miskolc, Miskolc-Egyetemváros, Hungary
(E-mail: erika.b.varga@uni-miskolc.hu, attila.baksa@uni-miskolc.hu)
Manuscript received May ..., 2024; revised ..., 2024.

DOI: 10.36244/ICJ.2025.5.4

C. Paper Structure

The paper is structured as follows: Section II outlines the research objectives. Section III reviews related work on the use of GitHub in education and the application of process mining in educational research. Section IV discusses process discovery algorithms, comparing them and selecting the most appropriate for the study. Section V focuses on data modeling and analysis, detailing the data collection process, the OCEL schema used for structuring the event log, and an analysis of the collected data, including key statistics. Section VI presents the results of learning process modeling, while Section VII identifies bottleneck problems that hinder student progress. Finally, Section VIII concludes the paper by summarizing the findings and discussing their implications for educational practices.

II. Research Objectives

Building on the challenges and motivations outlined in the introduction, this study seeks to achieve two main objectives:

1) Create a learning process model that is representative of individual cases to identify bottleneck problems that hinder students' progress.
2) Produce a learning process model of successful students to serve as a showcase for other students.

By focusing on these objectives, this research seeks to provide educators with actionable insights into the learning processes, enabling them to tailor their instructional strategies to better meet the diverse needs of their students. This, in turn, can lead to improved educational outcomes and more effective learning experiences.

III. Related Works

A. GitHub in Teaching Programming

There is a growing need for automated code assessment systems in computer science education due to the increasing number of learners and the limited availability of teaching staff. These systems aim to address the challenges of grading a large volume of code submissions. They help instructors save time, provide timely feedback to learners, and support the learning process. These systems target various types of errors in programming assignments, such as syntax, runtime, logic errors, and code quality issues, and may also address plagiarism concerns through similarity analysis [4]. Some tools use continuous integration for immediate feedback, while others perform symbolic executions and unit test assessments.

Within the array of tools, GitHub is a widely-used software development platform that originally supported version control, collaborative development, and project hosting. It is utilized by many businesses, organizations, and educators. In the context of education, GitHub has gained popularity in programming classrooms, with around $18,000$ educators incorporating it [5]. GitHub in education serves various purposes such as submitting assignments, collaborating on group projects, and receiving feedback. More recently, with GitHub Actions, teachers can automate the testing of code submissions.

The study by Hsing and Gennarelli [6] explores how the implementation of GitHub in programming classrooms affects students' learning outcomes and experiences. The researchers surveyed $7\,530$ students and 300 educators from classrooms using GitHub and classrooms not using GitHub. The findings indicate that incorporating GitHub in programming education yields several benefits, including enhancing students' familiarity with industry tools, facilitating collaboration and teamwork, boosting engagement, and fostering a sense of belonging in the classroom and within the field.

A more recent study [7] summarizes the key lessons learned when using GitHub in the classroom. First of all, the authors recommend providing proper instructions, compatible with students' prior experience, on how to use Git to gain the most benefits from system use. For undergraduate courses, GitHub Classroom is a better choice than GitHub, as GitHub Classroom simplifies the educational use of GitHub. They also found that custom offline automated systems are more effective for assessing students' assignments than using GitHub Actions.

In [8], the authors present two distinct approaches for automatic C code assessment in programming education: one is a custom-designed web-based tool, while the other involves using GitHub Actions and GitHub Classroom. The web-based tool offers a graphical user interface in Hungarian and assesses code based on various criteria, including syntax, behavior, and code quality metrics. The GitHub-based system employs repositories and automation to evaluate student assignments, providing immediate feedback and the ability to track students' progress and activities. The authors conducted experiments at the University of Miskolc to compare the effectiveness of both systems in engaging students and improving their coding skills. The results show that both systems were beneficial for student engagement and learning, with the GitHub-based system offering more comprehensive tracking capabilities and integration with the software engineering community's practices.

B. Mining Educational Data

The field of educational data mining (EDM) has received significant attention in recent years. Researchers and educators are increasingly seeking to utilize the extensive data generated within educational settings to gain valuable insights and drive substantial improvements. This goal can be achieved by applying data mining techniques, which offer a robust set of tools capable of uncovering hidden patterns, associations, and anomalies within educational data. By understanding these patterns, educators can develop targeted interventions and personalized learning strategies that address the specific needs of individual students [9], [10].

Building on the foundation established by educational data mining, the field of educational process mining has emerged as a specialized discipline dedicated to extracting meaningful insights from event logs. Process mining techniques [3] are particularly effective for analyzing the sequences of activities performed by students, which are captured in event logs generated from interactions with learning management systems, online courses, and other educational technologies. By applying process mining techniques to these event logs,

researchers can uncover the actual learning paths taken by students, identify critical points where students struggle or deviations from expected or "ideal" paths, and explore how these deviations impact learning outcomes [11], [12]. These valuable insights allow for educators to tailor their teaching strategies.

In summary, while educational data mining provides broad insights into various types of educational data, educational process mining focuses on the detailed analysis of event logs to uncover the actual learning paths taken by students. The integration of the two fields provides a powerful means to enhance the understanding and improvement of educational processes. This paper aims to contribute to this growing field by exploring how process mining algorithms can be applied to educational data, ultimately paving the way for more effective and personalized education.

C. Event Log Standards

Process mining algorithms work with event logs produced by various information systems. Consequently, these logs can appear in numerous formats and instantiations. Each system architecture with a logging mechanism has historically developed its own solution for recording events. The initial effort to standardize event logs was MXML (Mining eXtensible Markup Language), which organized timestamps, resources, and transactions in a uniform format. This was followed by the introduction of the XES (eXtensible Event Stream) standard, which was published in 2016 as the IEEE 1849-2016 Standard for eXtensible Event Stream.

Despite its comprehensive approach, the XES standard encounters difficulties when handling object-centric data (e.g., database tables) due to the complexities of one-to-many and many-to-many relationships. In response to these challenges, a new method was proposed to extract, transform, and store object-centric data, resulting in the Object-Centric Event Log (OCEL) Standard which was released in 2021 [13].

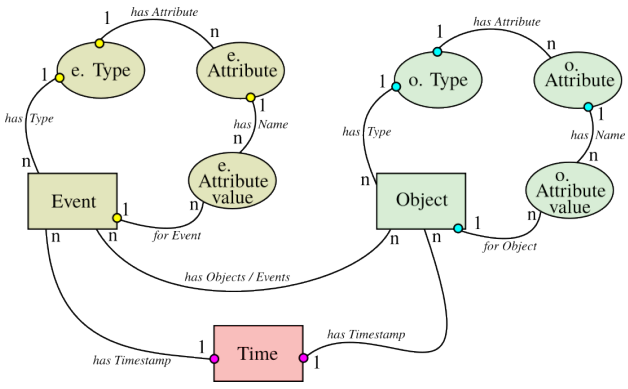


Fig. 1. OCEL schema with $n - n$ Events and Objects connections

OCEL 2.0 represents a modernized approach to capturing and analyzing event data in complex information systems [14]. Traditional event logs typically record sequences of events related to single entities, such as customer orders or individual transactions. However, many real-world processes involve multiple interacting objects, such as customers, orders, products, and payments. OCEL 2.0 addresses this complexity by enabling a richer and more holistic representation of event data (see Figure 1). The key features of OCEL 2.0 [14] are as follows.

- *Multi-Object Support*
  Multiple Objects Per Event: Unlike traditional event logs that often focus on a single case or entity, OCEL 2.0 allows each event to be associated with multiple objects of different types. This capability is crucial for accurately modeling complex processes where events involve interactions between several objects.
- *Enhanced Data Model*
  Object Types and Attributes: OCEL 2.0 defines various object types and allows each object type to have its attributes. This structured approach facilitates detailed analysis and better understanding of the relationships between different objects.
- *Improved Analysis Capabilities*
  Holistic Process Views: By considering the interactions between multiple objects, OCEL 2.0 enables more comprehensive process mining and analysis. This leads to deeper insights into process performance, bottlenecks, and areas for improvement. Multi-Perspective Analysis: Analysts can explore processes from different perspectives, such as the lifecycle of individual objects or the interactions between specific types of objects.
- *Interoperability and Standards*
  Standardized Format: OCEL 2.0 promotes the use of standardized formats for event logs, ensuring compatibility and interoperability between different tools and systems used in process mining and analysis.
- *Tool Support*
  Integration with Process Mining Tools: OCEL 2.0 is supported by various process mining tools, enabling users to leverage its advanced features for process discovery, conformance checking, and performance analysis.

Overall, OCEL 2.0 represents a significant advancement in the field of process mining and event log analysis, allowing for a more nuanced and complete understanding of the processes.

IV. Methods

In this research, we utilized the PM4Py Python program library [15], [16], a sophisticated tool developed by the Fraunhofer Institute, to implement a variety of algorithms and services associated with process mining. PM4Py stands out for its comprehensive suite of functionalities that facilitate the analysis, visualization, and discovery of process models from event logs, making it an ideal choice for our investigations.

Our primary focus was on process discovery algorithms, which are essential for revealing the underlying process models that govern the sequences of events captured in logs. An event log is defined as

$$\mathcal{L} = \{< A, B, C, D >, < A, C, D >, < A, B, D >\} \quad (1)$$

where A, B, C, and D are the events in the log, and $\mathcal{L}$ denotes the possible order of the events. To this end, we

conducted an in-depth examination of three prominent process discovery methods: the Alpha Miner, the Inductive Miner, and the Heuristic Miner [17]. Each of these methods brings distinct methodologies and advantages to the table, allowing us to explore different dimensions of process discovery.

The *Alpha Miner* algorithm [22], one of the earliest and most fundamental process discovery techniques, operates by identifying and interpreting patterns of event sequences to construct a process model. This algorithm works by examining the direct succession of events and determining causal relationships, which it then uses to build a Petri net representation of the process. In Eq. (1) these two cases are possible (which is a sequence pattern):

$$A \rightarrow B \quad \text{or} \quad A \rightarrow B \tag{2}$$

While the Alpha Miner is effective in providing a basic structure of the process, it has limitations when dealing with noise, infrequent paths, and complex dependencies within the event log.

In contrast, the *Inductive Miner* algorithm [19], [20] offers a more advanced and robust approach. It employs a recursive technique that divides the event log into smaller parts, constructs models for these parts, and then combines them to form a comprehensive process model. A directly follows graph is represented mathematically by

$$\mathcal{G}(\mathcal{L}) = (A_L, \rightarrow_L, A_L^s, A_L^e) \tag{3}$$

where A_L is an event in the log, $\rightarrow_L$ means an edge between two events (a directly follows relation), A_L^s is the start and A_L^e is the end event.

This method is particularly suitable for handling noisy and complex data, producing more precise and comprehensible models. Its ability to generate hierarchical models makes it especially useful for understanding complex processes with multiple layers of activities.

The *Heuristic Miner* algorithm [21] takes a different approach by focusing on the discovery of frequent patterns and significant relationships within the event log. This algorithm uses statistical measures to assess the frequency and significance of event connections, thereby identifying the most prominent pathways in the process. The Heuristic Miner is particularly valuable for analyzing real-world data that may contain deviations, exceptions, and variations, providing a realistic and practical view of the process.

In our investigations we used a real-world event log recorded by Github, which contains few frequent patterns and a multitude of unique cases. For this reason, the Alpha Miner was deemed insufficient for our needs due to its sensitivity to noise and difficulty in handling complex dependencies. Instead, we opted to use the Inductive Miner and Heuristics Miner algorithms.

V. Data Modeling and Analysis

Our experiment covered 52 practical problems propagated using GitHub Classroom within an introductory programming course. These assignments are designed for out-of-classroom practice to help novice programmers in developing their skills in C programming. The tasks are categorized into 8 groups according to the related topics as listed in Table I. The groups represent the level of the acquired skills as the tasks are designed to incrementally build upon one another. The 1st level is completed if the tasks from the first 3 groups are solved. The 2nd level is done after completing tasks from groups 4 and 5. The next level contains tasks from groups 6 and 7, and the top level is represented by the tasks of group 8. Students were asked to complete tasks in the order of these levels, and they were free to choose the tasks to solve and their order within the topic groups. We encouraged them to complete as many assignments as possible and try to deliver more than one acceptable solution for the same problem.

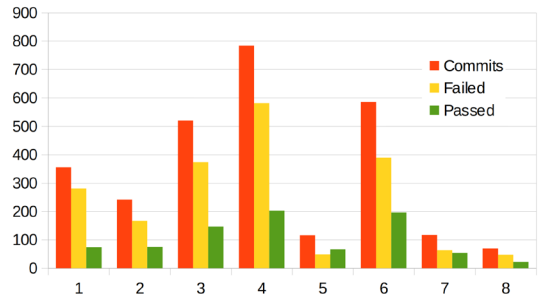


Fig. 2. Commit statistics for tasks groups

In the present study, 59 students registered for GitHub Classroom to access these assignments as extracurricular activities. They collectively produced 2,789 commit events, which were automatically tested by GitHub Actions to provide prompt evaluations as either failed (1,951 commits) or passed (838 commits). The statistics of the executed commits are shown in Figure 2. These data are recorded and stored in an event log, with each entry containing the GitHub identifier of the student (resource), the name of the assignment (activity), the time of the commit event (timestamp), and its result (conclusion). The details of the assignments include their topic group and level. Student data come from the university's learning management system, where their mid-semester performance (signature) and final grade (grade) have been recorded. Since all three sources are utilized to address our research objectives, the data model is described by the OCEL schema in Figure 5.

```
<object>
   <string key="id" value="student1"/>
   <string key="type" value="user"/>
   <list key="ovmap">
     <string key="signature" value="1"/>
     <string key="grade" value="2"/>
   </list>
 </object>
```

```
<object>
   <string key="id" value="1_06"/>
   <string key="type" value="task"/>
   <list key="ovmap">
     <string key="topic" value="1"/>
     <string key="level" value="1"/>
   </list>
 </object>
```

Fig. 3. Objects data in XML format

TABLE I
COMMIT STATISTICS FOR TASK GROUPS

Task group	Programming topic	Level	Num. of tasks	Commits	Failed	Passed	Success rate(%)
1	Use of variables, basic data types, arithmetic operators, and built-in mathematical and input/output functions	1	10	355	281	74	20.85
2	Control structures (selections and iterations), use of relational and logical operators	1	7	242	167	75	30.99
3	Basic algorithms (counting, summing)	1	8	520	373	147	28.27
4	Using one-dimensional numeric and character arrays	2	7	784	581	203	25.89
5	Basic algorithms with one-dimensional arrays (max/min selection, searching)	2	2	116	49	67	57.76
6	Defining functions	3	9	585	389	196	33.50
7	Working with strings and functions	3	4	117	63	54	46.15
8	Using struct data type	4	5	70	48	22	31.43
Total			**52**	**2 789**	**1 951**	**838**	**30.05**

```
<event>
  <string key="id" value="0"/>
  <date key="timestamp" value="2022-09-26T02:19:00"/>
  <string key="activity" value="1_06"/>
  <string key="resource" value="student1"/>
  <list key="omap">
    <string key="object-id" value="student1"/>
    <string key="object-id" value="1_06"/>
  </list>
  <list key="vmap">
    <string key="conclusion" value="failure"/>
  </list>
</event>
```

Fig. 4. Event data in XML format

The data are stored in an xmlocel file. In this format, events are represented as shown in Figure 4, capturing essential attributes such as timestamp, activity, and resource. The event type is not specified because all events in our data set are of the type GitHub commit. However, the result of the commit is important and is stored in the conclusion attribute of the event. Each event is connected to two objects: a student and a task. Student objects serve as the resources for the events and are described by the signature binary flag and exam grade attributes. Task objects represent the activities affected by the event and are described by the topic and level attributes, as shown in Figure 3.

Table I summarizes the commit data for the programming task groups, while Table II presents the same data categorized by the students' exam grades.

We can see from Table I that students achieved the lowest success rate in the first task group, although these tasks are the simplest ones. The reason behind this is that the use of Git was new to the students, and they used these simple tasks to get acquainted with the technique of taking an assignment and then committing and pushing the solution back to GitHub while producing a high rate of unsuccessful commits. In task groups 5 and 7 less number of commits occurred with the highest success rate, which means that most students quit the game after completing the 1st level and those who carried on were eager to solve the problems. The least number of commits were executed in task group 8. Here the success rate is low, which can be attributed to the fact that this group contains advanced-level tasks that are not included in the end-semester

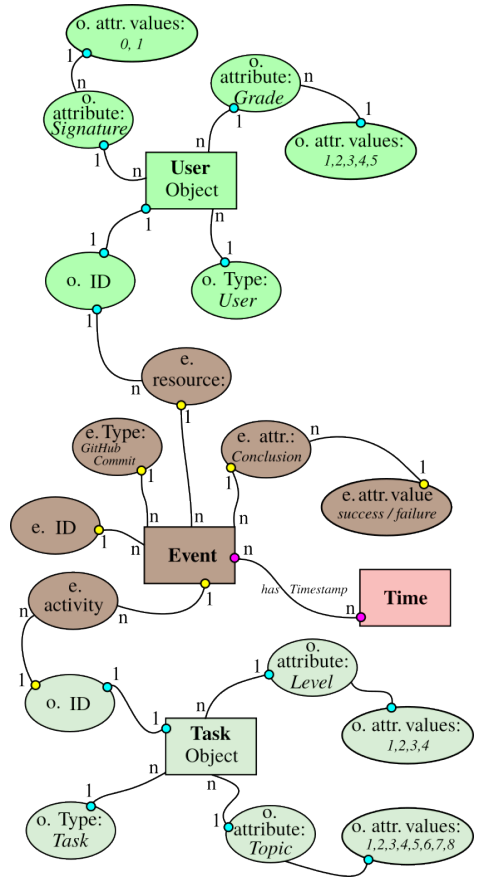


Fig. 5. OCEL schema for User-Task-Event log

exam. Therefore few students worked on these problems and they were not motivated to make more effort.

Table II summarizes the commit events from the students' aspect. The data show that the success rate of GitHub commits and the number of successful commits per student increase in line with the grade, so we can conclude that students who were more insistent in finding the right solution achieved better results at the end of the semester. It is also worth noting that

TABLE II
COMMIT STATISTICS FOR STUDENT GRADES

Exam grade	Num. of students	Commits		Failed		Passed		Success rate (%)
		Total	*Per student*	*Total*	*Per student*	*Total*	*Per student*	*Total passed / total commits*
1 – failed	14	489	34.93	391	27.93	98	7.00	20.04
2 – passed	17	766	45.06	564	33.18	202	11.88	26.37
3 – satisfactory	9	526	58.44	364	40.44	162	18.00	30.80
4 – good	6	307	51.17	197	32.83	110	18.33	35.83
5 – excellent	13	701	53.92	435	33.46	266	20.46	37.95
Total	59	2 789	47.27	1 951	33.07	838	14.20	30.05

the total number of trials is significantly higher in the case of students passing the exam than for those who failed. This number is also remarkable for excellent students compared to those who completed the exam with less success.

By applying process discovery methods to this event log, we first created and compared learning process models of successful and failing students. The second experiment focused on creating a learning process model that is representative of individual cases to identify bottleneck problems that hinder students' progress.

VI. Learning process modeling

Analyzing the event log, the aggregated values in Table III do not show a significant difference between the students who completed the C programming course with an exam grade > 1 and those who failed the exam in terms of the number of commits. We applied independent t-tests to investigate this hypothesis with 95% significance, and we obtained a p-value of 0.20 for the total number of commits and a p-value of 0.50 for the failed commits, so there is no evidence to reject the null hypothesis of equal means in these cases. On the other hand, there is a difference between the number of successful commits, as indicated by a p-value of 0.0056. Also, there is a significant difference between the means of the number of tasks solved, as indicated by a p-value of 0.0077.

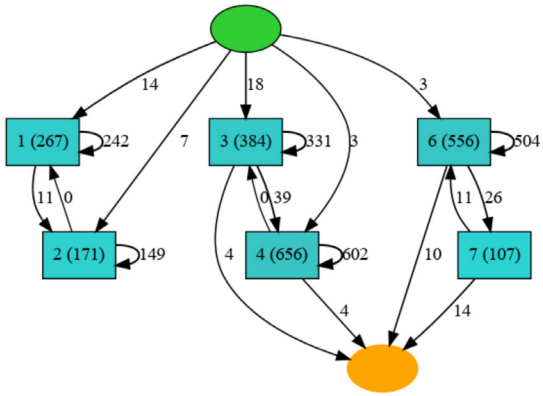


Fig. 6. Learning process model of successful students

For modeling the learning process of the two student groups, the event log was first filtered to include only those students who completed the C programming course. This resulted in an event log containing 2 300 committed events from 45 students. We applied the Heuristic Miner algorithm in the PM4Py package to produce the heuristics net in Figure 6, as opposed to Figure 7, which was generated from the event log containing the 489 commits of 14 students who failed the course exam. This heuristics net provides a comprehensive visualization of the processes, highlighting the critical paths, frequent events, and transitions.

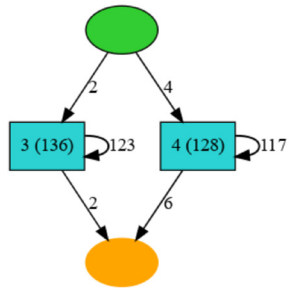


Fig. 7. Learing process modell of failing students

In Figure 6, we can see that in the group of successful students, 14 started the learning process with a task from topic No. 1, 7 started with topic No. 2, and 18 started with topic No. 3. Most frequently, they stopped after completing tasks in topics No. 6 or No. 7. Students typically solved the tasks in the order of the topics. The only exception is when some of them stepped back from topic No. 7 to topic No. 6. Tasks from topic No. 5 were less frequently solved, which is why this is not depicted in the figure. The numbers in the brackets after the topics indicate the related commit events. In this respect, topics No. 4 (Using one-dimensional arrays) and No. 6 (Defining functions) contain the tasks that required the most practice, as evidenced by the highest number of repetitions. This indicates a high number of unsuccessful commits, which motivated students to keep on practicing with another task from the same topic.

Comparing this learning process model to the one in Figure 7, we can conclude that failing students solved fewer tasks from a smaller number of topics, typically from No. 3 or No. 4, and then stopped practicing. This resulted in a smaller number of successful commits, which discouraged them from continuing this kind of practice. This explains why their learning model is less complex than the model of the other group.

TABLE III
AGGREGATED VALUES OF COMMIT EVENTS FOR STUDENT GROUPS

		Num. of commits	Successful commits	Failed commits	Num. of tasks
All students	mean	47.27	14.20	33.07	16.58
	max	221	36	185	42
	min	2	0	0	1
Successful students	mean	51.12	**16.45**	35.67	**18.76**
	max	137	35	110	42
	min	2	0	0	1
Failed students	mean	34.93	**7.00**	27.93	**9.57**
	max	221	36	185	35
	min	2	0	0	1
T-test p-value		0.20	**0.0056**	0.50	**0.0077**

VII. Detecting bottleneck problems

A bottleneck problem is defined as a task that is the last task attempted by a student during their learning process within any topic except for topics No. 6, No. 7, and No. 8, and concluded with failure. Identifying these problems is crucial for instructors when determining their teaching methods. If they recognize the most significant problem areas, they can focus on these during practice in class. We do not consider tasks from the last topics as problematic, even if most students stop practicing after completing them, because these topics represent the highest level we cover for novice programmers.

In this examination, we identified 18 tasks where students quit the practice process. From these, we filtered out the ones that meet our bottleneck problem definition and visualized the learning processes that most frequently failed at these tasks. In this case, the filtered event log contained a significantly reduced number of cases showing infrequent behavior, rendering the Heuristics Miner algorithm inapplicable. As an initial approach, we used the Directly Follows Graph (DFG), where the nodes represent the activities in the log, and directed edges exist between nodes if there is at least one trace in the log where the source task is directly followed by the target task. Frequency values are represented on these directed edges. Figures 8 and 9 highlight two examples of bottleneck problems for instructors of C programming fundamentals.

A. Topic No. 3: Basic algorithms

Task 3_02: Count leap years between two years that you have to read from the standard input while checking their validity (integer numbers between 1 900 and the current year).

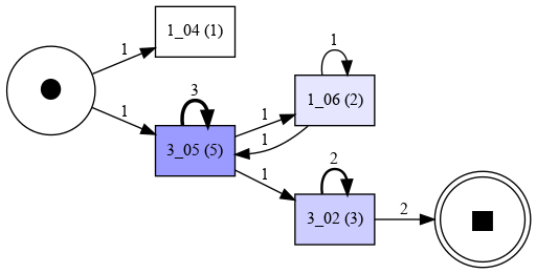


Fig. 8. DFG graph of learning processes ending with Task 3_02.

B. Topic No. 4: Using one-dimensional arrays

Task 4_03: Read float numbers from the standard input, store them in an array, and decide whether the values follow a monotonic-increasing order or not.

Since a DFG-based process map is created without generalization, it is typically much more complex than other process models. For this reason, we also experimented with the Inductive Miner algorithm of the PM4Py package, which is optimized for handling infrequent cases. This method generated the process tree on Figure 10 for the problem at hand, which was converted to a BPMN model for better comprehension.

We can conclude that students following the learning process in Figure 8 were not persistent and stopped practicing after a few attempts. On the other hand, students following the learning process in Figure 9 got stuck at Task 4_03 after solving several tasks.

VIII. Conclusions

This study aimed to enhance the understanding of successful learning processes in programming by applying process discovery methods to student commit data. In an introductory programming course for first-year Computer Science BSc students, 52 practical problems were distributed via GitHub Classroom as out-of-class assignments. GitHub recorded 2 789 commits from 59 students, whose exam grades were extracted from the learning management system of the university. All these data were incorporated into an object-centric event log, which was converted to a case-based log to execute the process discovery algorithms implemented in the PM4Py library. The primary objectives were to identify the distinguishing features of the learning processes of successful students and to detect bottlenecks that hinder student progress.

By applying the Heuristics Miner and Inductive Miner process discovery methods to the event log, we first created and compared learning process models of successful and failing students. This comparison revealed specific patterns and strategies that contributed to student success. Subsequently, we conducted a second experiment focusing on identifying bottleneck problems within the learning processes. These bottlenecks were found to be significant barriers to student progress, providing crucial insights for improving educational practices.

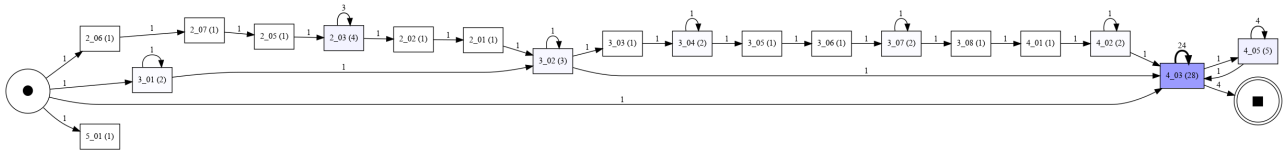


Fig. 9. DFG graph of learning processes ending with Task 4_03.

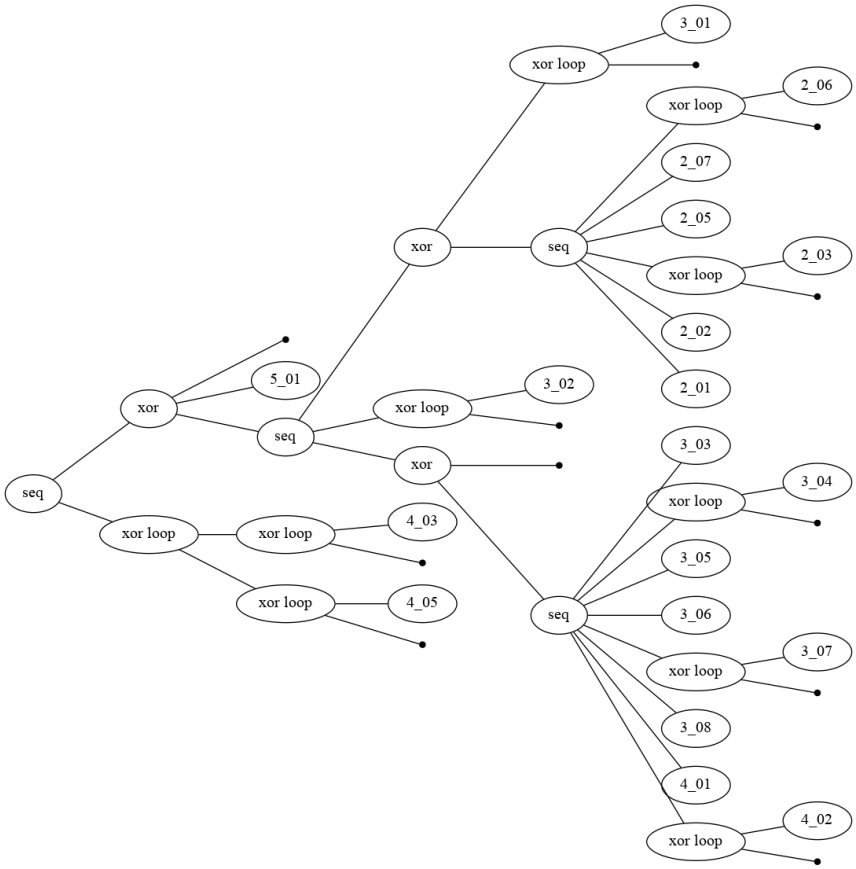


Fig. 10. Process tree of learning processes ending with Task 4_03.

Overall, the findings from this research contribute to a deeper understanding of how students interact with programming assignments. The results highlight the importance of tailored instructional methods that address individual learning challenges, ultimately enhancing educational outcomes.

References

[1] M. A. Ghazal, O. Ibrahim and M. A. Salama, "Educational Process Mining: A Systematic Literature Review." *2017 European Conference on Electrical Engineering and Computer Science (EECS)*, Bern, 2017, pp. 198–203, DOI: 10.1109/EECS.2017.45

[2] I. Y. Kazu, "The Effect of Learning Styles on Education and the Teaching Process.", *Journal of Social Sciences*, 5(2), pp. 85–94, 2009, DOI: 10.3844/jssp.2009.85.94

[3] W. van der Aalst, Process Mining – Data Science in Action, *Springer Berlin*, Heidelberg, 2016, DOI: 10.1007/978-3-662-49851-4

[4] S. Combéfis, "Automated Code Assessment for Education: Review, Classification and Perspectives on Techniques and Tools." Software 2022, 1, pp. 3–30. DOI: 10.3390/software1010002

[5] Empowering the next generation of developers, [Online], Available: https://github.com/edu

[6] C. Hsing, V. Gennarelli. "Using GitHub in the Classroom Predicts Student Learning Outcomes and Classroom Experiences: Findings from a Survey of Students and Teachers." In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, pp. 672–678. 2019, DOI: 10.1145/3287324.3287460

[7] Yu-Cheng Tu, V. Terragni, E. Tempero, A. Shakil, A. Meads, N. Giacaman, A. Fowler, K. Blincoe. "GitHub in the Classroom: Lessons Learnt." ACE '22: *Proceedings of the 24th Australasian Computing Education Conference*, pp. 163–172, 2022, DOI: 10.1145/3511861.3511879

[8] E. Baksáné Varga, K.A. Fekete. "Applications for Automatic C Code Assessment." In: *Proceedings of the 2023 24th International Carpathian Control Conference (ICCC)*, pp. 21–26, 2023, DOI: 10.1109/ICCC57093.2023.10178987

[9] C. Romero, S. Ventura. "Educational data mining and learning analytics: An updated survey." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery.

[10] Y. S. Mitrofanova, A. A. Sherstobitova, O. A. Filippova. "Modeling Smart Learning Processes Based on Educational Data Mining Tools." In: Uskov, V., Howlett, R., Jain, L. (eds) *Smart Education and e-Learning 2019. Smart Innovation, Systems and Technologies*, vol 144. Springer, Singapore. DOI: 10.1007/978-981-13-8260-4_49

[11] E. M. Real, E. Pinheiro Pimentel, L. V. de Oliveira, J. Cristina Braga, I. Stiubiener. "Educational Process Mining for Verifying Student Learning Paths in an Introductory Programming Course." *2020 IEEE Frontiers in Education Conference (FIE)*, Uppsala, Sweden, 2020, pp. 1–9, DOI: 10.1109/FIE44824.2020.9274125

[12] M. A. Ghazal, O. Ibrahim and M. A. Salama. "Educational Process Mining: A Systematic Literature Review." *2017 European Conference on Electrical Engineering and Computer Science (EECS)*, Bern, pp. 198–203, 2017, DOI: 10.1109/EECS.2017.45.

[13] A. Ghahfarokhi, G. Park, A. Berti, W. Aalst. "OCEL: A standard for object-centric event logs." ISBN 978-3-030-85081-4, 2021, pp. 169–175.

[14] OCEL 2.0 Specification, [Online] Available: https://www.ocel-standard.org/specification/overview/

[15] State-of-the-art-process mining in Python, Fraunhofer Institute for Applied Information Technology (FIT). [Online] Available: https://pm4py.fit.fraunhofer.de/

[16] A. Berti, S. van Zelst, D. Schuster, "PM4Py: A process mining library for Python", *Sowtware Impacts*, 17, 100556, 2023, DOI: j.simpa.2023.100556

[17] L. Numminen, "Process Mining Algorithms Simply Explained", 2023, [Online] Available: https://www.workfellow.ai/learn/process-mining-algorithms-simply-explained

[18] W. van der Aalst, A. J. Weijters, L. Maruster. "Workflow Mining: Discovering Process Models from Event Logs." *IEEE Trans. on Knowledge and Data Engineering*, 2004

[19] S.J. Leemans, D. Fahland, D., W. van der Aalst. "Discovering block-str uctured process models from event logs – a constructive approach." In: *Petri Nets. Lecture Notes in Computer Science,* vol. 7927, pp. 311–329. Springer, 2013

[20] S. J. Leemans et al. "Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour." *Business Process Management Workshops*, 2013

[21] A. J. Weijters, J. T. Ribeiro. "Flexible Heuristics Miner (FHM)", *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, Paris, France, 2011, pp. 310–317, DOI: 10.1109/CIDM.2011.5949453

[22] W. van der Aalst, A.J. Weijters, L. Maruster. "Workflow Mining: Discovering Process Models from Event Logs." *IEEE Trans. on Knowledge and Data Engineering*, 2004

Erika Baksáné Varga was born in 1976. Got MSc degree in Information Engineering in 2000 and MSc degree in Economics for Engineers in 2003 from the University of Miskolc. PhD student at József Hatvany Doctoral School for Computer Science and Engineering from 2003 till 2006. Holds PhD degree since 2011 for defending the thesis entitled "Ontology-based Semantic Annotation and Knowledge Representation in a Grammar Induction System". Currently associate professor at the Institute of Information Technology. Research interests: programming languages, natural language processing, data and knowledge engineering, process mining.

Attila Baksa was born in 1976. Received his MSc degree in Information Engineering from the University of Miskolc in 2000. Holds PhD degree since 2006 for defending the thesis entitled "Numerical Analysis of Mechanical Contact". Currently associate professor at the Institute of Applied Mechanics. Research interests: programming, numerical simulations, nonlinear problems, algorithms, finite element method.

Failure prediction with Weibull distribution

Anita Agárdi[1], and Károly Nehéz[1]

Abstract—**In the industrial environment, the reliability of machines and equipment is not only a matter of convenience but a key factor in terms of the productivity and competitiveness of companies. Unexpected breakdowns, shutdowns or malfunctions of machines can cause serious economic damage, not to mention potential workplace accidents or environmental damage. This article presents a failure prediction model, where the probability failure of machines and pieces of equipment is determined using the Weibull distribution. The model can predict the failure of a single machine and determine the failure of the entire system. After the introduction, the most important literature on the subject is presented, followed by a description of the Weibull distribution. The article describes the test datasets and their results. The tests were created for the following data sizes: 2 units, 5 units, 15 units, 40 units, 100 units.**

Index Terms—**failure prediction, machine health, Weibull distribution**

I. Introduction

The reliability and efficiency of machines and pieces of equipment are key factors for the smooth operation of production. Machine breakdowns and unexpected shutdowns can cause significant economic losses and also increase the risk of workplace accidents and environmental pollution. To increase the efficiency of industrial maintenance and operation, failure prediction has become increasingly important. Over the years, researchers have proposed mathematical models and distributions to determine the probability of machine failure, which can be used to optimize maintenance schedules and minimize unexpected downtime. The Weibull distribution is a powerful tool in failure prediction and reliability analyses. The Weibull distribution is a parametric probability distribution. It is well adapted to data that comes from long-lived systems and helps in the analysis of systems where failure changes over time. The rest of the article is structured as follows: Section 2 describes the importance and the scientific background of the topic based on search engine analyses and related publications. Section 3 describes the Weibull distribution. After that, Section 4 presents the test datasets and their results. The last section presents the conclusion and future research direction.

The main contribution of this paper is the application of the Weibull distribution to predict system failures in several datasets of different sizes (2, 5, 15, 40, and 100 units). The research presents two innovative types of diagrams for visual representation of failure probabilities: failure unit and fault tree probability diagrams. These tools enable a more detailed and efficient understanding of the failure behavior of industrial equipment, providing insights to optimize maintenance schedules and reduce unplanned downtime.

II. Literature review

Over the years, many researchers have investigated the topic of failure prediction. The article presents the search results of Google Scholar, where the following keywords were investigated: "Asset lifecycle management", "Condition monitoring" and, "Maintenance strategies". , "Machine health monitoring", "Reliability-centered maintenance", "Equipment reliability", "Lifecycle cost analysis". Figure 1 presents the number of publications per year from 2010 to 2023. Based on the figure it can be seen that the number of publications is increasing over the year.

The article then presents some of the featured articles that investigate the topic of failure probability determination system.

Çınar, Z.M. et al. al [1] investigated predictive maintenance. The paper describes the following types of maintenance: reactive (repairing if it is damaged), planned (scheduled maintenance), proactive (troubleshooting to improve performance), and predictive (reliability is predicted). The article reports on a system where smart sensors monitor machines. This data is transmitted over the network, then the data is monitored by staff to investigate if the machines are well maintained and whether they are in good condition. The software can predict future failures and machine health. The system infers this condition from past data with Machine Learning. The system automatically issues a maintenance ticket to the technician, which the technician approves and performs.

Karuppusamy, D. P. [2] investigates the predictive maintenance scheduling. This maintenance process is based on sensors and their measurements. In the article, the following algorithm was used for predictive maintenance scheduling: decision tree, and random forest.

Luo, M. et al. [3] presented a two-stage maintenance system for equipment prediction and maintenance schedule optimization. A neural network is used for the failure prediction.

Wan, J. et al. [4] implemented a cloud-based big data solution for active preventive maintenance in a production environment. It provides data processing, analysis and forecasting.

[1] Institute of Informatics, Faculty of Mechanical Engineering and Informatics, University of Miskolc, Hungary (E-mail: anita.agardi@uni-miskolc.hu, karoly.nehez@uni-miskolc.hu)

DOI: 10.36244/ICJ.2025.5.5

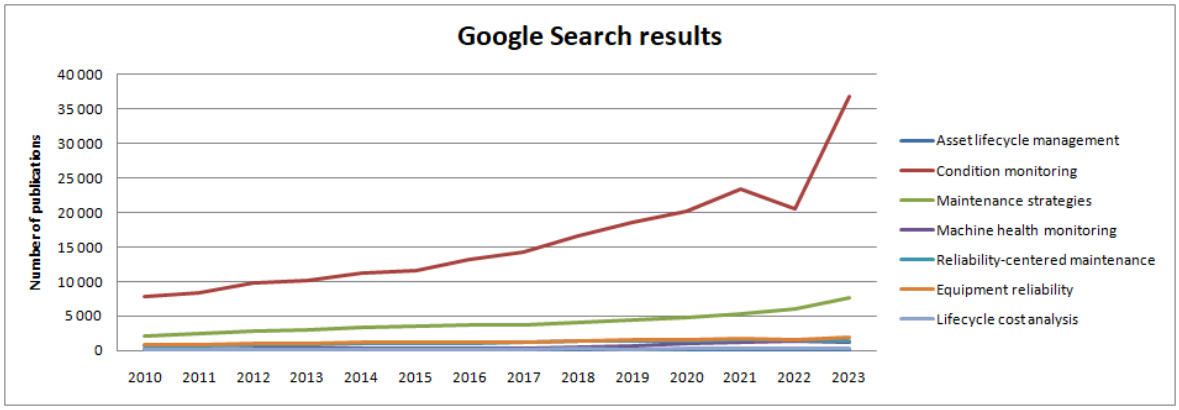


Fig. 1. Google Search result for "Condition monitoring" keyword

TABLE I
COMPARISON OF MAINTENANCE APPROACHES IN THE LITERATURE

Article	Objective	Methods Used	Data Source	System Examined
Our research	Failure prediction using Weibull distribution	Weibull distribution, tests for different equipment sizes	Simulated data (2, 5, 15, 40, 100 units)	Individual machines and full system
Çınar et al. [1]	Predictive maintenance	Machine Learning, analysis of data from smart sensors	Sensor data, transmitted over a network	General machines
Karuppusamy [2]	Predictive maintenance scheduling	Decision tree, Random Forest	Sensor data	Manufacturing equipment
Luo et al. [3]	Two-stage prediction and scheduling	Neural network	Sensor data	Industrial equipment
Wan et al. [4]	Big data analysis for preventive maintenance	Big data-based analysis and forecasting	Cloud-based system	Manufacturing environment
Bastos et al. [5]	Testing predictive algorithms	Rapid Miner, data mining algorithms (alarm, maintenance actions)	Single machine data	Single machine, multi-machine planned
Dangut et al. [6]	Aircraft maintenance and scheduling	Deep Learning, Markov Decision Process	Aircraft data	Aircraft maintenance
Irinyi & Cselkó [7]	Effectiveness of maintenance strategies	Testing constant and variable interval strategies	Artificially created faulty data	Simulated fault-repair system
Shimada & Sakajo [8]	Building maintenance	Time series analysis, statistical approach	Building condition data	Buildings
Wang et al. [9]	ATM error prediction	Classification algorithms (XGBoost, Random Forest, Ada Boost M1, LibSVM)	ATM data	Automated Teller Machines (ATMs)
Martins et al. [10]	Analyzing the condition of a paper industry press	K-Means, Hidden Markov Model (HMM)	Manufacturing data	Paper industry drying press

Bastos, P. et al. [5] investigated maintenance forecasting, during which data analysis algorithms are used. It utilizes the various data mining predictive algorithms found in Rapid Miner. The logical structure of the developed maintenance system consists of three parts: alarm, maintenance measures and predictive maintenance. The authors used the developed prototype system on a single machine, and they plan to apply it to several machines.

Dangut, M.D. et al. [6] investigated aircraft maintenance and maintenance scheduling. Deep learning techniques are used for the problem. In the system, the data is first subjected to a pre-processing step. Then the data is divided into two parts (training and test) and classification is used, and the classification algorithms are evaluated according to efficiency. The developed system was tested by the authors on datasets of real airplanes. Markov Decision Process (MDP) was used as failure prediction algorithm.

D. Irinyi and R. Cselkó [7] examined the effects of constant and variable interval maintenance strategies and compared their effectiveness. Tasks are examined through artificially created faulty data. In the first method, a certain number of errors are corrected, and in the second method, the authors prevent a certain number of errors.

Shimada, J., and Sakajo, S. [8] examine the maintenance of building facilities prone to collapse. The condition of the buildings is diagnosed with time series data. A statistical approach is used to determine when maintenance is required.

Wang, J. et al. [9] investigated predictive maintenance and present a classification-based error prediction method. The authors give an example of ATM maintenance in their article. The authors used the following classification algorithms in the system: XGBoost, Random Forest, Ada Boost M1, and LibSVM (which are available in Weka).

Martins, A. et al. [10] presented a maintenance task of a paper industry drying press as an example. The data was collected every minute over three years and ten months. Based on the data, the model classifies the status of the device into the following categories: "Proper operation", "Warning status" and "Device error". Data cleaning and normalization are

performed by the authors. Then the following methods are used to determine the states: K-Means and Hidden Markov Models (HMM).

III. Weibull-distribution

The Weibull distribution [11] is one of the probability distributions widely used in statistical modelling and data analysis. It is particularly popular in industrial reliability analysis and failure prediction. The Weibull distribution is often used to model events over time, such as machine failures, deaths, or product lifetimes. The advantage of the Weibull distribution is that it flexibly adapts to different data sets and reliably describes changes over time that often occur in real life. The following notations are used:

- η – life expectancy
- $F(x) = P(\eta < x)$ – distribution function

There are several common notations for the parameters of the Weibull distribution. The use of different notations is clearly explained by the fact that the Weibull distribution has been used very widely in a wide variety of scientific fields, as well as the fact that many different ways of determining the parameters are known, and the rewriting of the variables for each solution results in significant simplifications.

$$F_c(x) = \{1 - exp(-x^c), \ \ if\ x \geq 0,\ 0, \ \ if\ x < 0 \qquad (1)$$

The paper uses the above notation to denote the standard Weibull distribution. From this, the distribution of linear transforms results in the following formula:

$$F_c(\tfrac{x-a}{b}). \qquad (2)$$

This family of distributions is also a three-parameter, from which a is the so-called shape parameter (type parameter). However, it must be an asymmetric distribution.

1. In the case of the distribution, if $c = 1$ then the exponential distribution, if $c = 2$ the Rayleigh distribution, while $c = 3.57$ the distribution becomes nearly symmetrical and closely approximates the normal distribution. With a suitable parameter choice, it is also possible for the Weibull distribution to closely approximate the lognormal and Γ -distributions [12].

2. There are many methods for determining the parameters, but they are not robust on the one hand, and are difficult to handle on the other. A good example of difficult handling is parameter determination using the momemtum method. If ξ a, b, c are a random variable with parameter Weibull distribution, then

$$E\left((\xi - a)^k\right) = b^k\Gamma\left(1 + \tfrac{k}{c}\right), \qquad (3)$$

i.e.

$$\mu = E(\xi) = a + b\Gamma\left(1 + \frac{1}{c}\right) \sigma^2 = E(\xi - \mu)^2 =$$

$$b^2\left(\Gamma\left(1 + \frac{2}{c}\right) - \Gamma\left(1 + \frac{1}{c}\right)^2\right) \alpha_3 = E\left(\left(\frac{\xi-\mu}{\sigma}\right)^3\right) =$$

$$\frac{\Gamma\left(1+\frac{3}{c}\right)-3\Gamma\left(1+\frac{2}{c}\right)\Gamma\left(1+\frac{1}{c}\right)+2\Gamma\left(1+\frac{1}{c}\right)^3}{\left(\Gamma\left(1+\frac{2}{c}\right)-\Gamma\left(1+\frac{1}{c}\right)^2\right)^{3/2}} \qquad (4)$$

The μ, σ^2, α_3 can be easily determined from a given sample, the system of equations is not easy to solve, and it does not always have a solution [13].

Investigating with partial cases first, i.e. we assume that either the value of c or a is known.

1. If the value of c is known, then the value of a and b can be easily estimated from the previous system of equations, but the estimate will not be robust, because the average used to approach the expected value is not robust. On the other hand, the method of robust moments based on the distribution function gives a good estimate, since the distribution function is easy to handle and the type parameter is known. And so we already have a robust method, e.g. to the exponential distribution, to the Rayleigh distribution.

2. If a is known, then our sample or the distribution function can be transformed into another type of distribution with the following transformation [14,15]:

$$\eta = (\xi - a) \qquad (5)$$

then the distribution function is the following

$$G(y) = 1 - exp\left(-\ exp\left(\tfrac{y-T}{s}\right)\right), \qquad (6)$$

where $T = b$ is the location parameter and $s = \frac{1}{c}$ is the scale parameter. With this, we not only got a robust estimation option for the scale and shape parameters of the Weibull distribution, but also a simple option compared to the relatively complicated methods.

For the case when all three parameters are unknown, it was not possible to develop a truly robust method, because it was not possible to provide a really good robust estimate for the location parameter a or the shape parameter c separately. However, the following simple procedure can be used well.

The value of the parameter a from the sorted sample can be estimated with the following equation:

$$a_n^{(0)} = \xi_1^{\star} - 2(\xi_2^{\star} - \xi_1^{\star}) \qquad (7)$$

This can be further accelerated by not using the median and the median absolute deviation as a starting solution, but the

$$1 - e^{-1} \quad and \quad 1 - e^{-0.5} \qquad (8)$$

using quantiles (ordered sample elements) belonging to probabilities

$$T_n^{(0)} = (\xi_k^{\star} - a) \quad and \quad s_n^{(0)} = \frac{(\xi_k^{\star}-a)-(\xi_l^{\star}-a)}{(2)} \qquad (9)$$

using starting estimates where

$$k = \left[n\left(1 - e^{-1}\right) + 0.5\right]$$

$$and \qquad l = \left[n\left(1 - e^{-0.5}\right) + 0.5\right]. \qquad (10)$$

Computer experience shows that if the number of sample elements is greater than 100, then the values of the parameters can be easily recovered from the pseudo-random numbers. The statistic value of ω_n^2 is usually less than 0.2. This shows that the fit is acceptable at almost all significance levels.

For example, the computer runs to return the theoretical result that a sample with a normal distribution can be well approximated by a Weibull distribution.
If we use the suggested quantile estimates instead of the median and median absolute deviation, the number of iteration steps is approx. halved.

Inference Process

The Weibull distribution allows for flexible modeling of failure probabilities for individual components. During inference:

- Input Data: The model starts with time-to-failure data for each unit or system component. This data is either collected empirically or generated synthetically, as in your test cases.
- Parameter Estimation: The Weibull parameters (a, b, c) are estimated using robust methods when possible. Depending on the available information:
  - If c (shape parameter) is known, robust estimation techniques can calculate a (location) and b (scale) efficiently.
  - If a (location parameter) is known, transformations simplify the estimation of b and c.
  - If all parameters are unknown, initial estimates are derived using quantiles ($1 - e^{-1}$ and $1 - e^{-0.5}$) to accelerate computation and reduce iteration steps.

The outcome of this process is a probability distribution for each component's time-to-failure.

Aggregation Using Dependency Graphs

To determine the system-level failure probability, the model aggregates individual unit distributions based on their relationships within a dependency graph:
Graph Representation: Nodes represent components (units), and edges define dependencies (e.g., whether one component's failure directly impacts another).
System Failure Rules:

- Series Configuration: The system fails if any component fails.

$$P_{system} = 1 - \prod_{i=1}^{n}(1 - F_i(t)) \qquad (11)$$

where $F_i(t)$ is the failure probability of the i-th component

- Parallel Configuration: The system fails only if all components fail.

$$P_{system} = \prod_{i=1}^{n}(1 - F_i(t)) \prod_{i=1}^{n}(1 - F_i(t)) \qquad (12)$$

IV. Test tesults

This section presents the test results. During the research, test runs were conducted using the following datasets: 2 units, 5 units, 15 units, 40 units, and 100 units. The datasets consist of randomly generated test data, designed to include parameters relevant to building operations.
In this context, a 'unit' refers to individual components in the case of a single machine or, for a system, different machines that collectively form the system. Units represent elements that function together within a system. A failure in a single unit can potentially impact the entire system's operation.
The program is self-developed, with a Python backend and Angular frontend. In Python, we used the Reliability library for the operations in connection with distribution. The frontend was coded in Angular, and the Chart.js library was applied for the charts.
The test results is represented on two types of diagrams. The first is the failure-unit probabilities diagram, where the units are listed as individual objects when determining the failure probabilities, and the failure-tree probabilities diagram, where the failure probability of individual units is already determined by the failure probability of those units determines what it depends on.

A. 2 units

In this test case, the system contains 2 units. The start date of the measurements is: 28.01.2014, and the end date is: 28.01.2024. 10 years have passed between the start date and the end date of the measurements. And approx. 90-120 days passed between the two measurements.
The forecast starts from 28.01.2024 and does it every 10 days, and it makes a forecast for a total of 100 days from the start day.

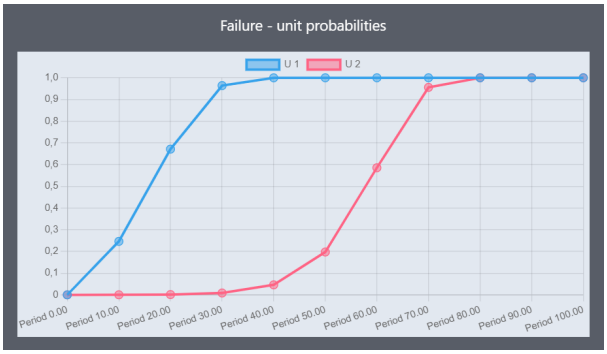


Fig. 3. Failure-unit probabilities diagram for 2 units

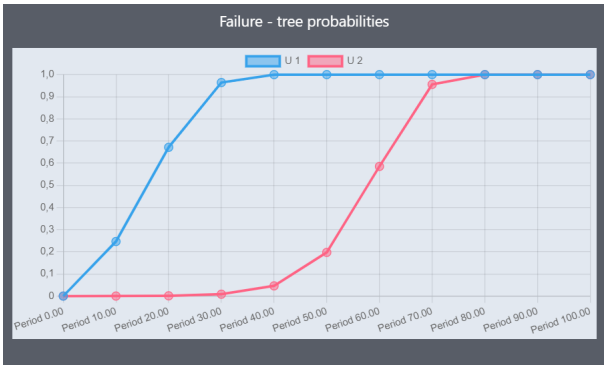


Fig. 4. Failure-tree probabilities diagram for 2 units

In the figure above, we can see that the failure-unit probabilities and the failure-tree probabilities diagrams are the same. Even at the starting time, both units have a 0 probability of failure. U1 fails at day 40 with probability 1, and U2 fails at day 80 with probability 1.

B. 5 units

The measurements for 5 units are also started on 28.01.2014. and lasted for 10 years. 90-120 days also passed between each measurement.

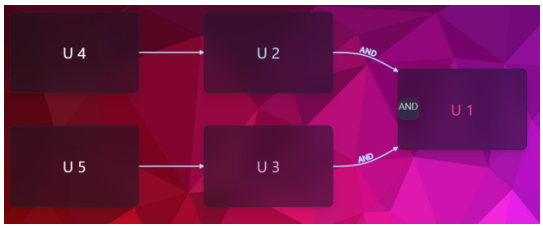


Fig. 5. Dependency graph for 5 units

It can be seen that U1 has an AND relationship with U2 and U3. This means that if U2 or U3 fails, then U1 will fail as well. U2 depends on U4, while U5 depends on U3, which means that if U4 fails, so does U2, and if U5 fails, U3 also fails.
The forecasts are begin from 28.01.2024.

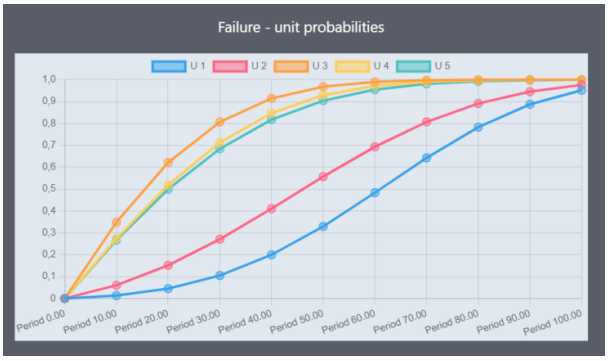


Fig. 6. Failure-unit probabilities diagram for 5 units

The Figure 6 is the figure of failure-unit probabilities, which shows that U3 has the earliest probability of failure, and U1 has the lowest probability of failure. As the period increases, the probability of failure increases. While the failure rate is 0 at the start time, this value is often 100 or close to 100 days after the start time, so it is almost certain that the individual units will break down.

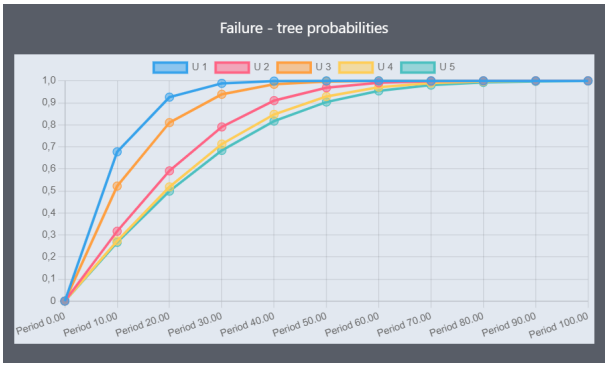


Fig. 7. Failure-tree probabilities diagram for 5 units

The failure-tree probabilities diagram already takes into account the dependence on other units for the failure probabilities of individual units. This type of diagram always gives greater or equal failure values than the failure-unit probability diagram, since it no longer examines the individual units by themselves, but also includes the dependence on other units in the failure probability.
The values of the failure-tree probabilities diagram are already different from the values above. Here, U1 has the highest probability of failure (it also depends on U2 and U3, which also depend on U4 and U5, so U1 actually depends on all other units). U4 and U5 have the lowest failure probabilities, since these units do not depend on the others. Here, U1 already fails with a probability of 1 in the 30th period, while the failure probability of each unit becomes 1 around the 80th period.

C. 15 units

The measurements started on 28.01.2014. The measurements were created every 90-120 days for 10 years.

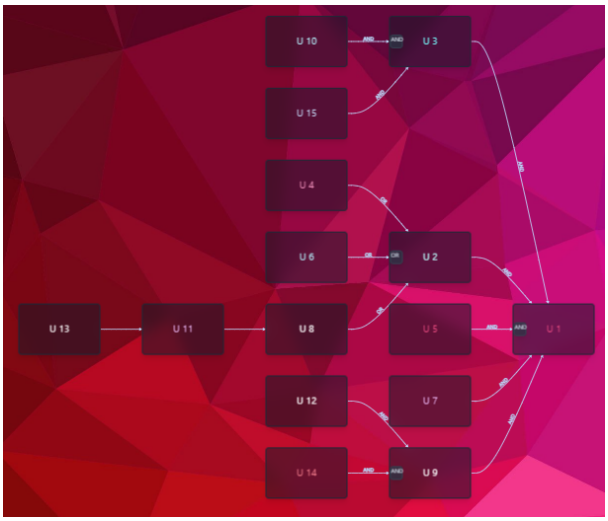


Fig. 8. Dependency graph for 15 units

The diagram above shows the graph representation of the system. The U1 directly or indirectly depends on each unit.

This unit is AND connection with U3, U2, U5, U7 and U9. U3 is also AND connection with U10 and U15. U2 is OR connection with U4, U6 and U8. U9 is AND connection with U12 and U14. In the case of the AND connection, if one unit fails, the given unit will also fail. In the case of an OR relationship, the given unit fails if all the units on which it depends are error.
The forecasts started from 10.01.2023 and forecasts were created every 10 days for 150 days.

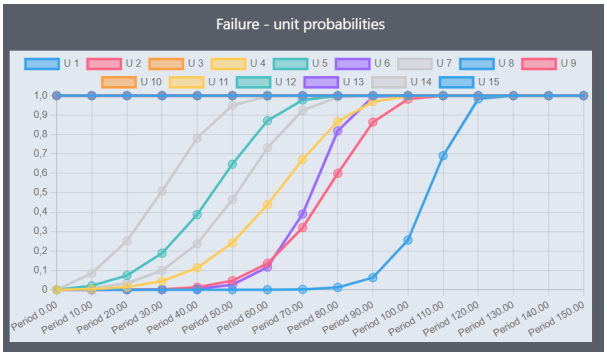


Fig. 9. Failure-unit probabilities diagram for 15 units

The failure-unit probabilities diagram is shown in the figure above. Accordingly, it can be seen that some units will fail from the start date of the forecast. While many units will certainly not fail even when the estimates begin. However, approximately 120 days from the start of the estimate, all units will be defective. It can be seen that some units break down later, while other units break down earlier (on their own, not as a dependency system).

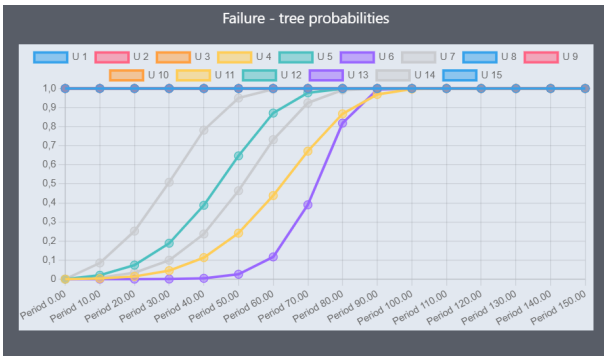


Fig. 10. Failure-tree probabilities diagram for 15 units

The failure-tree probabilities diagram is already narrower, because here we also take into account dependencies. So if a given unit would not be fail in itself at a given time, but the unit (or units) it depends on is fail, then the unit will also fail on this diagram. In the graph above, we can see that the U1 unit is the one that depends on all other units, so it is already fail at the beginning of the forecast. The diagram shows that certain units have a 0-probability failure at the beginning of the forecast, but after 90-100 days, all units will fail. It can also be seen that there are only 6 units that do not fail at the start of the forecast.

D. 40 unit

The previously presented systems were smaller, but the created software is also suitable for analyzing large systems. In this test run, we created a system with 40 units. The beginning of the measurements is 28.01.2014., and the measurements lasted for 10 years, the measurements were made every 90-120 days here as well.

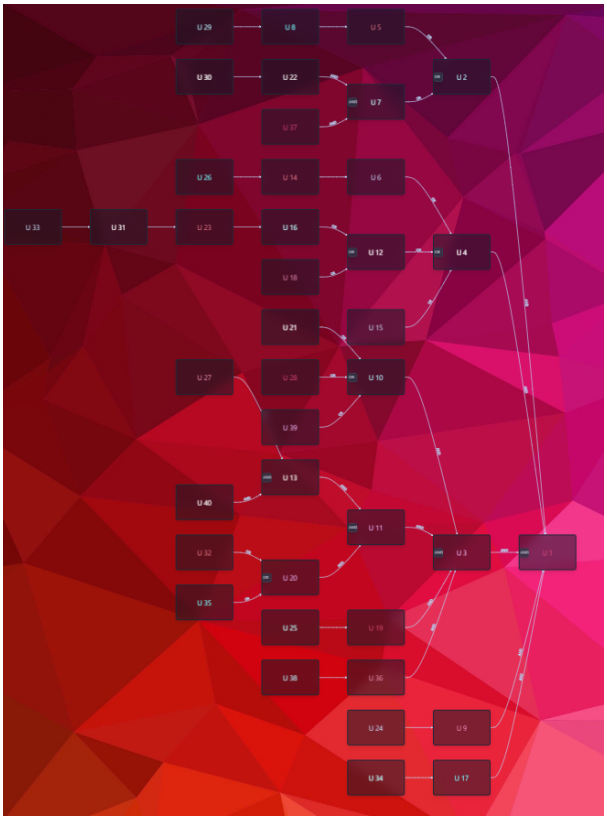


Fig. 11. Dependency graph for 40 units

The figure above presents the system in a graph structure. Unit U1 depends directly or indirectly on all other units.
Here too, the starting date was 10.01.2023, forecasts were created every 10 days for 150 days.

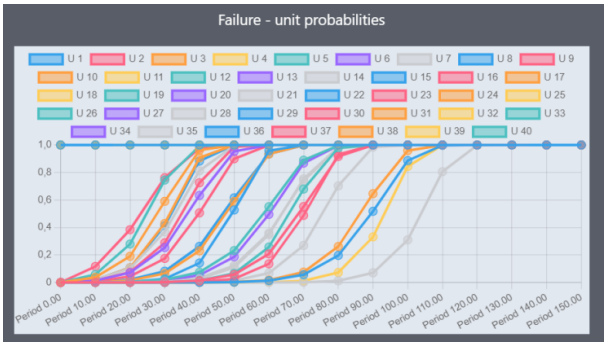


Fig. 12. Failure-unit probabilities diagram for 40 units

The figure above is the failure-unit probabilities diagram. Here we can see the individual failure probabilities of each unit. Some units fail already at the start of the estimate, while other units approx. errors occur only 120 days after the start of the estimate. We can see how different the failure probabilities of individual units are.

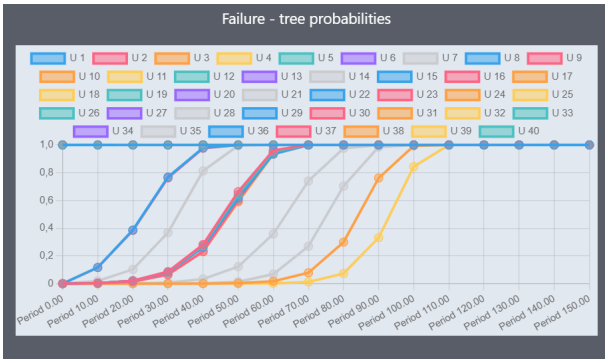


Fig. 13. Failure-tree probabilities diagram for 40 units

In the case of the failure-tree probabilities diagram, many units will already be fail at the start of the measurement, because here the system also includes the dependency relationships in the evaluation. It can be seen that individual units fail only 90-110 days from the start of the estimate. This is possible for units that do not depend on other units (they may already depend on units, but the number of dependencies is not large).

E. 100 unit

The last data line shows a system of 100 units. The measurements started on 28.01.2014, lasted for 10 years and the measurement was made every 90-120 days.
For the forecasting the 2023.10.01 was chosen as the starting date, and forecasts were created every 10 days for 150 days from the starting date.

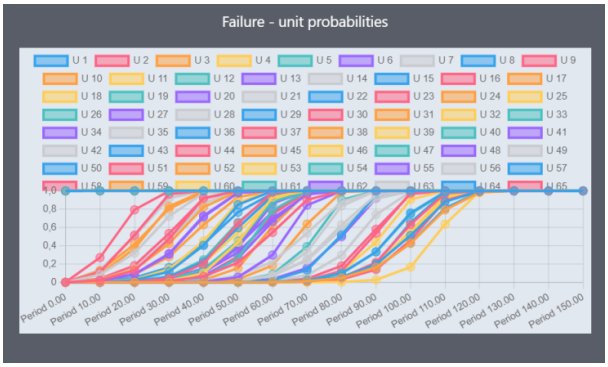


Fig. 14. Failure-unit probabilities diagram for 100 units

The figure above is the diagram of failure-unit probabilities, where the failure probabilities of the individual units were also illustrated. Here we can see that most of the units do not fail even at the start time of the forecast (0 probability of failure). However, by the 120th day from the forecast start time, almost all units will be faulty.

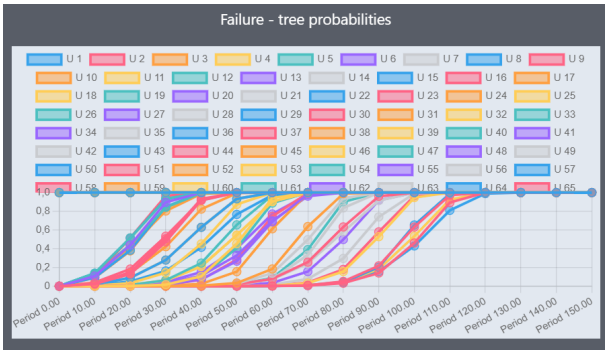


Fig. 15. Failure-tree probabilities diagram for 100 units

With the created software, we can create a diagram for a specific unit, then only the unit and its dependencies (direct and indirect dependencies) will be visible. This is also good because in practice it is possible that we only want to know about the failures of a few units. Furthermore, even in the case of this large example, it is easier to view the entire system in its small details, because the diagrams projected on a part of the system are much more readable than the above-mentioned complete system diagram.
For example, for the U60 we get the diagrams below.

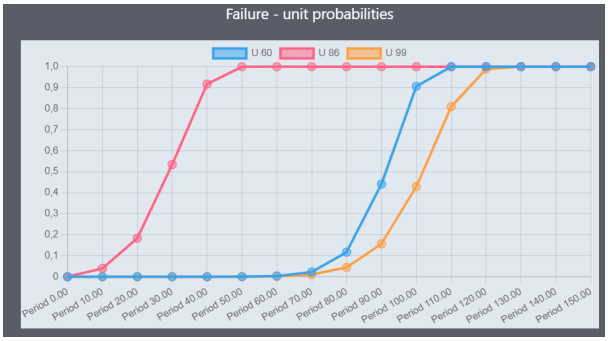


Fig. 16. Failure-unit probabilities diagram for U60

The diagram above shows the failure-unit probabilities diagram. U60, U86 and U99 are also represented because U60 depends on these units. It can be seen that U60 alone begins to fail drastically from the 70th day from the start of the forecast, and will definitely fail by the 110th day. U86 starts to fail drastically from the 10th day from the beginning of the forecast and the glaze deteriorates by the 50th day. U99 starts to fail from the 80th day and will definitely fail by the 120th day.

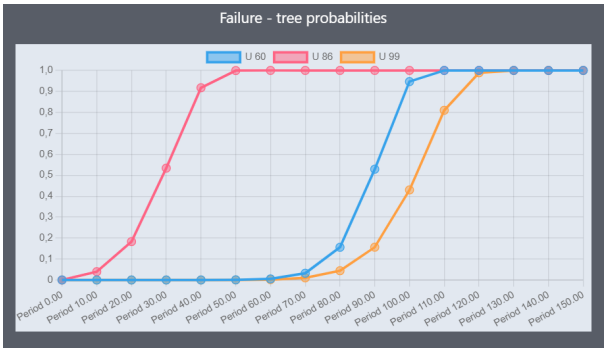


Fig. 17. Failure-tree probabilities diagram for U60

The failure-tree probability diagram is completely similar to the failure-unit diagram. This is possible because U60 is AND connected to the other two units, and while U86 will fail before U60, U99 will fail later, and U60 will still be functional (because of the OR connection) if the one of the units it depends on is still working.

After that, the paper presents another example: the following diagrams show the results for U6.

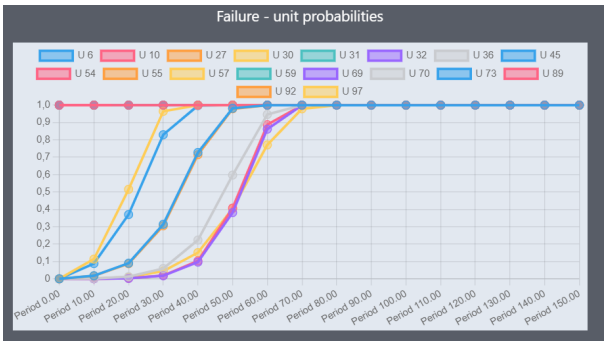


Fig. 18. Failure-unit probabilities diagram for U6

It can also be seen from the figure above that U6 has many dependencies If we compare it with the failure-tree probabilities diagram below, individual units fail sooner (due to dependencies) than in the diagram above.

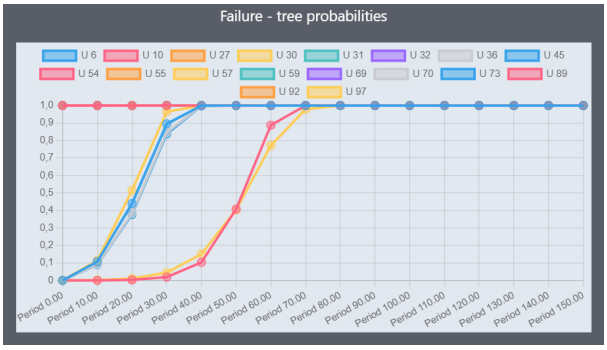


Fig. 19. Failure-tree probabilities diagram for U6

V. Conclusions

This paper presents the failure estimation of complex systems with Weibull distribution. During the analysis of the examined data series, it can be concluded that the Weibull distribution can be effectively used to determine the failure probabilities. In the data sets examined in the paper, it can be observed varying failure times for different units. This data allows us to determine the probability of machine failure over time and optimize maintenance schedules to ensure the uninterrupted operation of industrial processes. the article presented tests for the following data sizes: 2 units, 5 units, 15 units, 40 units, 100 units.

Overall, the Weibull distribution is efficient in industrial reliability analysis and failure prediction, which can help industrial companies optimize their operations and increase their competitiveness in the dynamic business environment.

Future research direction is the introduction of the developed system in a production environment and the comparison of the results given by the Weibull distribution with the results given by other algorithms.

Acknowledgement

This research was partially supported by National Research Development and Innovation Office, Hungary, grant number 2020-1.1.2-PIACI-KFI-2020-00147.

References

[1] Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, "Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0." *Sustainability*, vol. 12, no. 19, 8211, 2020, DOI: 10.3390/su12198211

[2] D. P. Karuppusamy, "Machine learning approach to predictive maintenance in manufacturing industry-a comparative study." *Journal of Soft Computing Paradigm (JSCP)*, 2(04), pp. 246–255, 2020, DOI: 10.36548/jscp.2020.4.006

[3] M. Luo, H. C. Yan, B. Hu, J. H. Zhou, and C. K. Pang, "A data-driven two-stage maintenance framework for degradation prediction in semiconductor manufacturing industries." *Computers & Industrial Engineering*, vol. 85, pp. 414–422, 2015, DOI: 10.1016/j.cie.2015.04.008

[4] J. Wan, S. Tang, D. Li, S. Wang, C. Liu, H. Abbas, and A. V. Vasilakos, "A manufacturing big data solution for active preventive maintenance." *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2039–2047, 2017, DOI: 10.1109/TII.2017.2670505

[5] P. Bastos, I. Lopes, and L. Pires, "Application of data mining in a maintenance system for failure prediction." *Safety, Reliability and Risk Analysis: Beyond the Horizon: 22nd European Safety and Reliability*, pp. 933–940, 2014

[6] M. D. Dangut, I. K. Jennions, S. King, and Z. Skaf, "Application of deep reinforcement learning for extremely rare failure prediction in aircraft maintenance." *Mechanical Systems and Signal Processing*, 171, 108873, 2022, DOI: 10.1016/j.ymssp.2022.108873

[7] D. Irinyi and R. Cselkó, "Analysis of failure prediction methods," *6th International Youth Conference on Energy (IYCE)*, Budapest, Hungary, 2017, pp. 1–5, 2017, DOI: 10.1109/IYCE.2017.8003692.

[8] J. Shimada and S. Sakajo, "A statistical approach to reduce failure facilities based on predictive maintenance." In *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 5156–5160, 2016, DOI: 10.1109/IJCNN.2016.7727880

[9] J. Wang, C. Li, S. Han, S. Sarkar, and X. Zhou, "Predictive maintenance based on event-log analysis: A case study." *IBM Journal of Research and Development*, vol. 61, no. 1, 11–121, 2017, DOI: 10.1147/JRD.2017.2648298

[10] A. Martins, I. Fonseca, J. T. Farinha, J. Reis, and A. J. M. Cardoso, "Maintenance prediction through sensing using hidden markov models—A case study." *Applied Sciences*, vol. 11, no. 16, 7685, 2021, DOI: 10.3390/app11167685

[11] J. Wang, and H. Yin, "Failure rate prediction model of substation equipment based on Weibull distribution and time series analysis." *IEEE access*, 7, 85 298–85 309, 2019, DOI: 10.1109/ACCESS.2019.2926159

[12] E. Dékány, J. Gedeon, L. Gillemot, "A Weibull-eloszlás néhány műszaki alkalmazása I.", *GÉP,* pp. 12-15, 1981

[13] I. Deák, "Véletlenszámgenerátorok és alkalmazásaik," *Akadémiai Kiadó,* Budapest, 1986.

[14] S. S. Shapiro, C. W. Brain, "Some new tests for the Weibull and extreme value distributions" *Coll. Math. Soc. J. Bólyai, Goodness-of-fit, Debrecen*, pp. 511–527, 1984

[15] V. M. Zolotarev, "Független valószínűségi változók összegeinek modern elmélete" *Nauka, Moszkva*, 1986

Anita Agárdi is senior lecturer at the Institute of Informatics, Faculty of Mechanical Engineering and Informatics, University of Miskolc, Hungary. She received a Ph.D. degree in Computer Science from the University of Miskolc in 2023. Her research interest include optimization algorithms, data mining and logistics.

Károly Nehéz got his MSc in mechanical engineering at the University of Miskolc, Hungary, in 1997 and PhD degree in software engineering in 2003. He currently works as an associate professor at the Institute of Computer Science, head of the institute since 2019. His primary research interest is Software Engineering, and AI algorithms.

Special Issue of the Infocommunication Journal

Using issue tracking as a groupwork facilitator in education

Melinda Magyar*, and David Burka*

Abstract—**In higher education, collaborative work is a prevalent method for skill development and assessment. This approach enables learners to use available time and resources efficiently, supported by peers, for deeper understanding and practical application of learned concepts. However, group task execution reduces individual work transparency. This aspect can be im-proved with the help of IT support, measuring not only out-comes but also individual contributions through task fragmen-tation, responsibility assignment, and performance tracking. In the business world, ticket management systems are commonly used for issue tracking, but less so in education. In this study, we describe integrating issue tracking for project management and communication in a new experimental course. We propose leveraging system data for evaluation. The presented empirical data and experiences could aid stakeholders in similar projects to benefit from issue tracking systems.**

Index Terms—**active learning methods, computer-supported collab-orative learning, education, issue tracking**

I. Introduction

Project-based courses are widely recognized and favored in universities globally, especially in engineering, computer science, business, and design, as they foster critical thinking and lifelong learning often through Problem-based Learning (PBL). In these courses, students collaborate to solve real or simulated challenges, applying theoretical knowledge practically, and developing teamwork and problem-solving skills. They take on significant responsibility for their learning, often working in groups to address real-world issues pertinent to their future careers, thereby gaining a comprehensive and interdisciplinary understanding [1].

The management and passing of tacit knowledge have many challenges, especially in an educational environment, where students have a far more diverse degree of motivation than employees in an actual workplace. Working on a project as a team with the support of a supervisor is among the best ways for students to develop skills that require experience; however, the equitable performance assessment of the individual student proves to be difficult [2].

An experimental project course was launched last year at Corvinus University of Budapest, designed to simulate real-world scenarios and enhance team-based project skills. As part of evaluating this course, the instructors sought to integrate formative evaluation methods to better address the common challenges inherent in such team-based projects. These challenges include the time and resource-intensive nature of the assessment, the establishment of fair assessment conditions and criteria, and the multitude of assessment milestones [3]. Faced with the substantial data need to fairly assess such project contributions, instructors recognized the potential benefits of adopting tools typically used in the corporate world. Consequently, an issue tracking system commonly employed in business environments was introduced, adapted to the academic setting to ensure complete transparency in grading. These are often referred to as ticketing systems, and the rest of the article uses the latter to avoid confusion when discussing the issues solved by the students.

Students were required to track their progress through a ticketing system, which aided in evaluating both the overall project and individual contributions. By making the quality of the ticketing system management a part of the grading process, the students were engaged as active partners in data collection, critical for accurately evaluating their individual contributions and teamwork. This integration seeks to refine the assessment of project-based learning by ensuring fairness and motivating continuous engagement.

Our goal is to develop courses which can prepare students for real life scenarios while addressing the typical issues associated with PBL. Traditional methods lack tools and data for individual assessments, leading to uneven work distribution among team members and the demotivation of students. The introduction of the ticketing system is a novel approach which can improve the engagement and efficiency of students while also providing a basis for fair assessment.

This study aims to verify the positive impact of the ticketing system and to further enhance course quality. We examine individual performance in comparison to other courses along with student feedback and the quality of final projects to assess the added value of our approach. By analyzing data collected from the ticketing system, we refine the evaluation method and propose changes aimed at improving the engagement of students.

II. Background

Active learning distinguishes itself from traditional educational methods by emphasizing student engagement and participation directly in the classroom setting. Unlike conventional lecture-based instruction, where students typically receive information passively, active learning involves students through meaningful tasks that necessitate critical thinking and reflection about their actions [4].

Problem-based learning is a frequently employed active learning method where the process of knowledge transfer begins with a problem definition, which then acts as a moti-

* Corvinus University of Budapest/ Institute of Data Analytics and Information Systems, Budapest, Hungary
(E-mail: melinda.magyar@uni-corvinus.hu, david.burka@uni-corvinus.hu)

DOI: 10.36244/ICJ.2025.5.6

vational cornerstone for all subsequent learning activities. PBL is first proposed by Neufeld and Barrows in the 1970s [5]. This approach deviates from the traditional passive student role, as students are expected to demonstrate a high level of autonomy based on their previously acquired knowledge and experiences. Within PBL, the typical sequence involves problem identification, group work, research, learning, solution development, and periodic presentations by student groups. The role of the instructor undergoes significant transformation during PBL. While in conventional teaching, instructors primarily convey their knowledge to students, in PBL, instructors adopt the roles of facilitators and mentors [6].

Team Based Learning (TBL) actively engages students in small collaborative groups, focusing on understanding and applying information. Its goal is to develop critical thinking, problem-solving, teamwork, and communication skills, deepening their understanding of the subject matter. [7]. The duration of tasks to be solved can vary based on the employed teaching strategy, ranging from a single class session to a complex project spanning an entire semester. In the university setting, TBL plays a pivotal role in enhancing students' skills by solving complex problems in teams [8].

When teams are assigned the responsibility of solving problems independently, it is easy to conclude that the instructional methodology should be combined with the use of specific roles. Role-Based Learning (RBL) is an educational method where students assume specific roles within a structured activity or scenario, often mimicking real-world professional environments. This approach aims to enhance understanding and skill development by placing students in contexts where they must apply knowledge, make decisions, and collaborate based on their assigned roles [9], [10].

Teamwork utilized in education often faces criticism for potentially leading to uneven distribution of workload and value creation among participating students (free-rider problem), especially when group members possess varying abilities or levels of commitment [11]–[13]. In his comprehensive review article, Davies (2009) systematically addresses critiques and recommendations concerning teamwork, encompassing the nature of assigned tasks, strategies for addressing motivational issues, and other pertinent considerations. Especially in the case of long-term teamwork projects, Kłeczek et al. [14] identified that they often lead to students feeling overwhelmed due to poorly managed workload distribution.

From the instructor's perspective, assessing the outcomes of teamwork, especially when the task involves creating a project workpiece, can be exceedingly resource-intensive [15], [16]. Evaluation may be constrained by the final product, but the need for fair judgment might necessitate accounting for individual performance and active participation within the team. This type of individual performance assessment demands a significant amount of data.

In computer-supported collaborative learning, students ideally engage with tools that prepare them conceptually and practically for real-life challenges. In certain cases, the tools themselves can be considered subjects of the training [17]. The corporate world has long utilized widely adopted solutions for tracking and managing problems to be solved either in teams or individually; these are the ticketing systems, also known as issue tracking systems.

Ticketing systems are pivotal tools in both the IT and customer service domains, designed to efficiently manage, track, and document tasks and issues [18], [19]. Such systems typically record incoming requests, problems, or tasks in a structured manner as tickets. These tickets are then allocated to the responsible party for task completion, who subsequently logs the action, optionally including the time spent.

A ticketing system can prove to be an effective tool for supporting and evaluating student work. By logging and documenting task performance through tickets, students enable precise assessment criteria during evaluations, minimizing conflicts within groups. This leads to more accurate individual and team performance assessments. Additionally, ticketing systems provide insights into student activities, preparing them for future workplace tools.

The widespread use of these systems can enhance academic assessment precision and also sets the stage for broader applications. Captured data, including causal relationships for tasks and solutions, can be used to train cognitive systems like chatbots [20]. The increasing use of generative AI in educational settings expands personalized and adaptive learning solutions [21], highlighting the potential to refine AI's effectiveness in academia.

Our inference is that the implementation of a project course should integrate the best practices of Problem-Based, Team-Based, and Role-Based learning with robust computer support. This setup enables students to manage their tasks (the quality of task management should also be included among the evaluation factors), and it offers instructors complete transparency. This approach shifts the focus of assessment from evaluating the final product to include the process of execution in the evaluation as well. Additionally, extensive data collection paves the way for data mining, which can provide valuable insights into both student and instructor behaviors.

III. Related work

There is a relative scarcity of publications concerning data collected by ticketing systems and the analyzes based on them, especially in a university setting. However, structured information made available by these systems can enable a wide array of data mining tools, including network analysis for uncovering collaboration patterns [22], application of text mining tools [23], [24], and even data-driven predictions [25], extending beyond obvious performance assessment analyzes [26]. The primary reason for the lack of findings is the limited adoption of ticketing systems in education and the business realm. In these areas, their usage is not widespread, and the data they store is often not extensively accessible for analysis due to the proprietary nature of business-related information.

Perera and colleagues conducted research based on ticketing systems and team-based learning [27]. Throughout the

study, they monitored the work of 43 students divided into seven distinct groups over a semester. The students collaborated on a software development project across three integrated platforms: a ticketing system, a version control tool, and a documentation wiki. The student groups in this research were homogeneous, meaning every member worked on identical tasks without designated leaders or coordinators. It was pointed out that if a team does not have a designated leader, but a leader stands out from the team in terms of behavioural patterns, this has a positive impact on the performance of the group.

Version control systems, which are commonly used in the business world to track software development activities, are also suitable for measuring individual contributions. In their article, Fernandez-Gauna et al. describes a sophisticated approach by using Git for the automated assessment of team-coding assignments in a university setting [28]. Version control capabilities are leveraged to gather detailed metrics on both team and individual student contributions. Team Performance Metrics (TPM) assess the overall health of the project, such as the percentage of time the code builds successfully and passes tests. Individual Performance Metrics (IPM) evaluate personal contributions, including the regularity of commits, adherence to coding standards, and the effectiveness of each student's code in passing automated tests. These metrics are periodically compiled into reports, offering continuous, detailed feedback to students and instructors. This process helps identify both collective and individual performance issues, facilitating targeted improvements.

It is important to differentiate in the case of team task execution in relation to team composition. In university settings, teams are often homogeneously structured: each student's role is identical, and they participate in solving the problem according to their own motivational and skill levels. In real life, a project team rarely consists of members with identical roles. It is more likely that each member has a designated role, which may rarely or never change during the course of the project. This role-based learning is evident in Sancho and his colleagues' study that focuses on distributed problem-based learning [9]. The learning process described in the article takes place in a virtual world, where instructors assign missions to students working in various roles. The learning process is realized by completing these missions. The study primarily examines students' effectiveness and the architecture of a custom-developed system, emphasizing the distributed problem-based learning through three case studies.

Role-based learning can also be seen as an element of gamification, and in the case of long-running projects, education can be enriched with a number of role-playing games taken from corporate life. Gamification appears in student assessment in the article by Udeozor and co-authors [10]. They propose a Game-Based Assessment Framework and discuss how immersive learning technologies can enhance education but require new assessment methods. The authors propose a Game-Based Assessment Framework (GBAF) that leverages the Evidence-Centered Design (ECD) framework and Constructive The framework collects data through gameplay, providing immediate feedback and aligning game tasks with learning outcomes. The study showed that students' performance improved with these immersive assessments, suggesting that the GBAF is a practical tool for integrating immersive technologies in education.

Overall, there are many studies implementing techniques like TBL, RBL or GBAF to improve the transfer of tacit knowledge while keeping the students engaged. The use of a ticketing system can support these solutions while also addressing the usual issues with objective and fair evaluation of the individuals. However, to our knowledge, the studies addressed in this chapter are the only ones which delved into the possibilities of ticketing systems in education, but they all narrowed their focus on software development related tasks. Our approach aims to be more generalized and flexible in the topic as well as the heterogeneity of individual tasks.

IV. Context and data

We collected and analyzed data over a semester from the activities of 90 second-year students majoring in Business Informatics. During the classes students were able to gain essential practical experience in the fields of software development and IT project management, building upon their previous studies. The course was launched for the first time, making its execution akin to a pilot project. The main project deliverable is creating a corporate website with e-commerce functionality using a Content Management System (CMS), ideally capitalizing on the abilities they had acquired in their earlier studies. CMS is a software that helps users to easily manage, edit, and publish websites and digital content. It is essentially a web-based application installed on a web server that allows users to create, edit, organize, and share content. The problem was chosen to be non-trivial, not easy to solve even with external help, and to better reflect students' attitudes to poorly structured problems and difficult tasks. During the project, groups of 4-5 students were responsible for the installation and customization of a CMS system, the design of a webshop, and the development of a custom-built module.

For the course, DotNetNuke (DNN) has been chosen, which is a specific example of a CMS based on the Microsoft .NET framework When selecting the CMS to be used for the project work, our specific goal was to choose a relatively well-documented, open-source CMS, but one with low popularity. This reduced the likelihood of students applying ready-made solutions available on the internet without adding substantial value. In the CMS market, WordPress is the most widespread, while according to available statistics, DNN ranks 34th with a 0.16% market share [29]. Despite its low popularity, its development is ongoing [30]. By selecting DNN, we presented student groups with difficult but solvable problems typical of the implementation of CMS systems.

Reflecting on the free-rider problem commonly associated with TBL, we aimed to reduce homogeneity in student activity within teams using RBL. We designated roles within each team based on students' personal commitments and interests,

striving to create heterogeneous knowledge spectrums within teams. Role changes were not permitted during the semester. The project roles include 20 students each as administrators, developers, data managers, and content owners, managing team organization, development, data collection, and design respectively, with an additional 10 students focused on online marketing. The de facto team leader was the administrator, but we didn't formalize this to leave events unfolding organically, which as stated by Perrera could have a positive impact [27], [31]. The instructors were also involved in the role-play as stakeholders: they played the role of the future owner of the CMS system developed.

The semester, spanning 13 teaching weeks, was divided into four phases. At the end of each phase, which can be considered as project milestones, we provided feedback to students regarding their progress within the phase. Each phase had a specific main objective for both the team and individual roles. However, it was the responsibility of the students to define their detailed tasks, which also were subject to evaluation.

Although the instructors have access to the ticketing systems, and relied on the content of it, the scoring was done manually, based on subjective expert judgement. Students were required to use the ticketing system and its proper management was part of the grade. Team scores accounted for 25% of the points earned, which were the same for all members, while individual performance scores accounted for 75%. Each student prepared a personal report for each phase. The individual points were determined by their activity in the ticketing system, the completion and quality of their assignments, and the contents of their reports. The assignment point is subject to expert evaluation. The ticketing point should be more or less clearly derivable from the ticketing system data. Normally, the report point is closely related to both the ticketing and assignment points. Lastly, the team point should be the result of both individual performance and collaboration among the participants.

The project used the open-source ticketing system MantisBT. The relational database of the system provided a high degree of flexibility in data extraction. This system uses the usual structure of ticketing systems: the basic unit is the ticket, to which many fields suitable for classification, text comments and files can be attached. The time spent can be added as numerical information to the comments. A ticket, although containing a single specific person in charge, allows the work of different team members to be filed. The system also allows for the hierarchical linking of tickets and the definition and control of sub- and stage deadlines. Fig. 1 shows a simplified ticket management process through the state changes of a task. After formulating the problem to be solved, a new ticket is added. Once a responsible person has been assigned, the state changes to 'assigned'. The responsible person then indicates any comments and records the time spent in a note before closing the ticket. If there is a lack of information needed to solve the ticket, supplementary information can be queried from the reporter using the feedback state. If the available information is or becomes sufficient, the ticket can be resolved by the assigned person and returned to the submitter, who can then close or reopen it after testing. The tester, who can be the original reporter or another assignee, can also attach comments and time-tracking information during the process.
A detailed event log can be extracted from the ticket management system database, which can be analyzed to get a comprehensive picture of the activity of each team member. Over the semester, 90 users recorded around 20,000 events. The system distinguishes a total of 37 event types, which were grouped into six categories according to practical reasons, summarized in Table 11.

TABLE 1
EVENT CATEGORIES USED FOR GROUPING EVENT LOG RECORDS

Event Type	Description
NEW_BUG	Add a new ticket
BUG_ASSIGNED	Assigning a ticket to the assignee
BUG_RESOLVED	Set the ticket to solved status; wait for the test
BUG_CLOSED	Close ticket
BUGNOTE_ADDED	Adding a comment to the ticket. Comments also store the time spent, so it is possible that several participants are working on the same ticket assigned to a user. Bugnotes can also hold file attachments.
BUG_EDITED	Modify any of the ticket data in addition to the above

We have created a project for each student team in the system. The teams were not allowed to see each other's projects, they could only work on their own. The chosen system allows for a hierarchical grouping of projects, so each project was categorised under a project representing the seminar group. This allows for instructor-dependent analyzes and analyzes based on the characteristics of the seminar group, such as the number of teams in the course.

We calculated the Cumulative Grade Point Average (CGPA) for the participating students at the beginning of the semester. The CGPA is the overall average of a student's grades throughout their academic program. It is calculated by taking the grade points earned in all courses, multiplying each by the credit hours of the course, summing these values, and

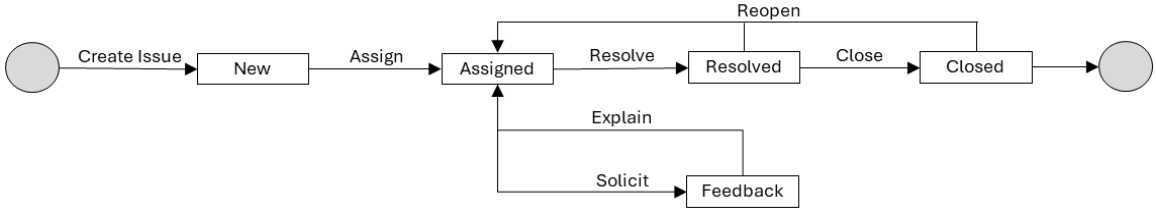


Fig. 1. Visual representation of simplified ticket handling process through state changes of a ticket.

dividing by the total number of credit hours completed. This measure provides a comprehensive overview of a student's academic performance. For engineering students, the CGPA may be a reliable predictor of final graduation performance, based on Adekitan and Salau's recent study [32]. We used this value to verify the adequacy of scores given through subjective evaluation.

V. Research Methods

The purpose of our research is to substantiate the empirical experiences of the course, implementing new solutions, and analyze the collected data to enhance the course content and evaluation system. The data collected from the ticketing system, along with the final grades and scores earned during the semester, form an event log, enabling not only exploratory statistical examination but also facilitating the analysis of temporal and sequential data.

After anonymizing and reducing the event categories (Table 1), the event log is prepared for examination. By using SQL queries, records from the event log can be extracted from the relational database of the ticketing system. We explored the collected data along three dimensions and derived further explanatory variables from the obtained data:

Temporal: Analyzing temporal aspects helps us understand how frequently students used the ticketing system, the evenness of activity distribution, and whether there are signs of expected real-time usage and indications of deviations from that pattern. To achieve this, we measure daily events and event types by role and collectively. Additionally, we conduct time-series analysis to uncover short-term and long-term seasonal effects.

Content: We examine how many times representatives of each role initiated events, what kind of content they recorded in the system, and how effectively they used the system for communication. The content dimension includes the received points, evaluations, and submitted reports. Our goal is to understand the behavioural patterns characteristic of each role and the students' attitudes towards tasks. For this purpose, beyond investigating averages and dispersion measures, we conduct dictionary-based sentiment analysis [33] on student reports using the PrecoSenti lexicons [34], [35], after lemmatizing the texts using Hunspell [36]. Primarily for methodological verification, we also perform sentiment analysis on the textual evaluations provided by the instructors, comparing the results with assigned scores. If we observe the expected correlation, it suggests that our analysis is correct, and the results obtained from the sentiment analysis of the students' reports are relevant as well.

Network: Collaborative learning is based on cooperation. We are interested in understanding, based on the data from the ticketing system, which group members were able to collaborate effectively. We aim to investigate whether weaker connections align with the experiences reported in the evaluations, or if they were simply characterized by different communication channels in those cases. To examine this, we conducted a network analysis following these considerations: two team members are connected if they worked on the same ticket. The relationships between two students were weighted by the frequency of their connections.

Our goal is to leverage the insights provided by the analyzes to better support the work of struggling students in the next iteration of the course, and ideally, to design a more motivating yet fair scoring system. To achieve this, we will compare the results collected from the temporal, content, and network aspects with the scoring outcomes using a correlation matrix. This approach will allow us to identify significant relationships and patterns, thereby informing improvements in our pedagogical strategies and assessment methods.

VI. Results

The semester lasted for 14 weeks with a holiday week in the middle (week 7) and was divided into four phases. Each of the four project phases concluded with a phase-ending week, during which students presented their results and received the tasks for the next phase. The number of daily recorded events throughout the semester can be seen on Fig. 2. During the semester, there was a one-week break starting on April 3, during which the activity level dropped to zero. While this is not inherently concerning, this low activity level persisted until the end of the following week. This phenomenon is likely attributed to the half-semester exam period.

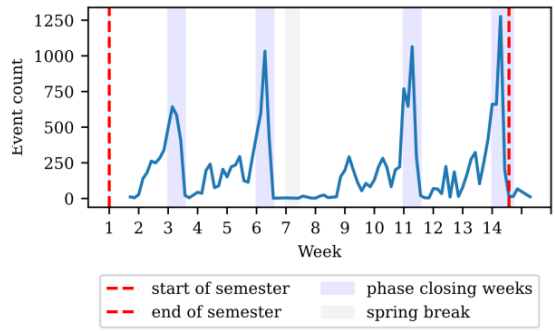


Fig. 2. The representation of the daily aggregated number of events. Four major milestones were designated during the semester, with the phase closing weeks indicated on the diagram (the closing presentations happened around the middle of the week). An increase in student activity can be observed as each phase closing week approaches, with activity peaking during these weeks.

We expected students to log their activities in real time. As deadlines approach in any project, motivation tends to increase, leading to a rise in the frequency of activities. Therefore, it was anticipated that activity levels would surge during phase-ending weeks. However, this should not overshadow the possibility that activities were entered into the system retroactively, solely to consider them during the scoring process. The event frequency curve shown in Fig. 2 suggests that this may have indeed occurred.

The daily activity divided by roles is depicted in Fig. 3. Remarkably, the Administrator role stands out in terms of activity, as these students recorded their activities in the ticketing system and closed them. Naturally, these two activities were more prominent during the phase-ending weeks. For the other

roles, such a trend effect is less acceptable, ideally, their workload should be distributed more proportionally. Nevertheless, for every role, there are four major peaks during the phase-ending weeks, which undoubtedly put the Administrator responsible for finalization in an uncomfortable position.

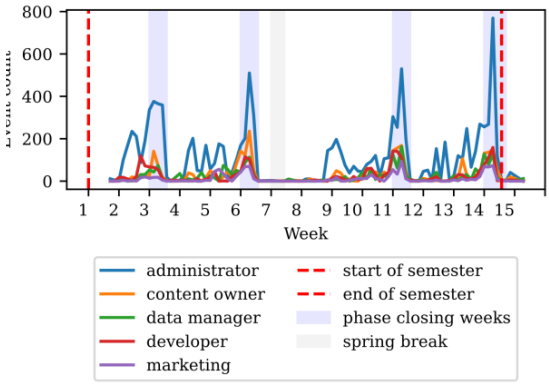


Fig. 3. Daily event count by roles, which accurately reflects the end-of-phase weeks.

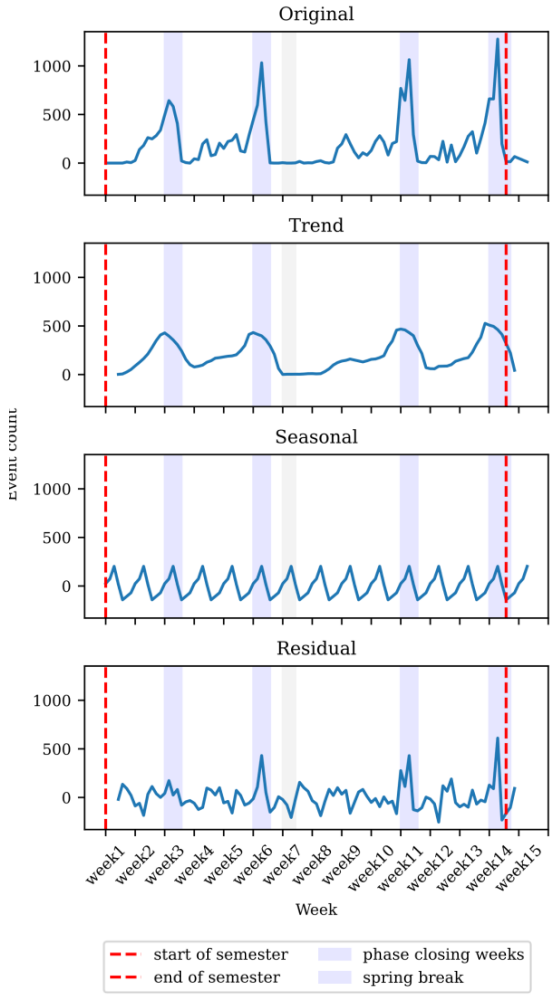


Fig. 4. Time series decomposition of daily event counts. The trend component shows the phases, while the seasonal component reflects the weekly repetition of classes.

To investigate the long- and short-term seasonal effects, the data set was analyzed using time series decomposition, the results of which are shown in Fig. 4. On the one hand, among the long-term trends, the aforementioned effect related to phase-locked weeks is clearly visible. On the other hand, among the short-term effects, it is also clear that a significant part of the activity is specifically linked to the weekly classes, although in fact there is no reason for this due to the organization of the training since the weekly exercises were weekly meetings without any task completion. The residual data series, stripped of trends and seasonality, also shows a spike in activity in the phase-out weeks.

The continuity of ticketing system usage is illustrated in Fig. 5, which provides information about active days, meaning how many students performed any activity in the system on a given day. The semester spanned 13 instructional weeks, covering 91 calendar days. The majority of students were active on fewer than a quarter of the possible days. This also suggests retroactive entry of activities. The figure also displays the distribution of active days by role. The average activity level of non-administrator role students is similar, though significant variations can be observed.

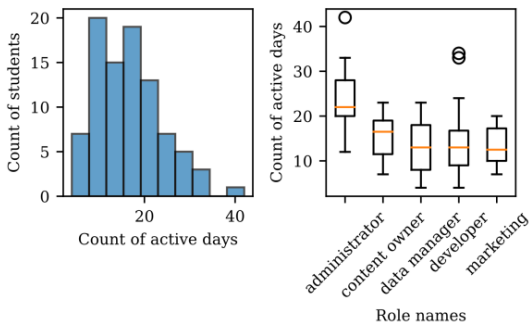


Fig. 5. Active days by students and roles. The majority of students were active at most twice a week during the semester. The most active role was that of the administrators, followed by the content owners, with the other roles having similar activity levels.

The low number of active days observed in Fig. 5 could also be attributed to the granularity of the tasks in the assignments. There is no consensus in the literature and practice regarding the ideal time requirements for tasks within an assignment. However, during the course, students were advised to work with relatively fine granularity, and ideally, the time requirement for a single assignment should not exceed one day. For instance, a task with a time requirement of 16 hours (calculated as four hours of work per day) could result in a perceived inactivity of 3 days. To determine this, we examine the reported time commitments by students, as shown in Fig. 6. The highest recorded time is approximately 100 hours, with an average of 44.3 hours per student and a high standard deviation of 22.1 hours. Since the total tracked time was not considered when determining the scores, the instructors did not focus on giving feedback regarding the lack of complete time recording. Thus, it is plausible that many students simply forgot to submit their time recordings, which is a common issue in real work environments as well.

Regarding the content created, apart from analyzing the temporal aspects, we also need to examine what was generated. Any ticketing system should be capable of reconstructing and presenting the completed work in a comprehensible manner, primarily through textual descriptions. This fulfilment is assessed through the analysis of the generated content in addition to the investigated events.

In summary, during the semester, students created 1954 tickets for their projects, to which they attached 4326 comments. In the specialized project designed for seeking support, they requested help in 156 cases, appending a total of 148 comments.

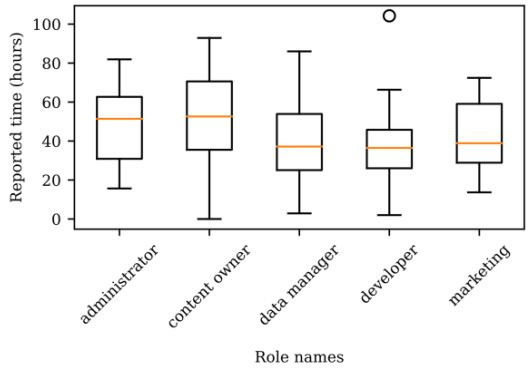


Fig. 6. Time reported by roles. According to the students' self-reported time logs, they spent a maximum of 60 hours on the course during the semester, which amounts to approximately 4 hours per

For the analysis of added textual content, we examined the length of content formulated in tickets and their associated comments. If the text is short, it can be assumed that it doesn't carry valuable information, and merely reviewing tickets and comments wouldn't yield more comprehensive insights into the workflow. The characteristics of log entries with content can be observed in Fig. 7.

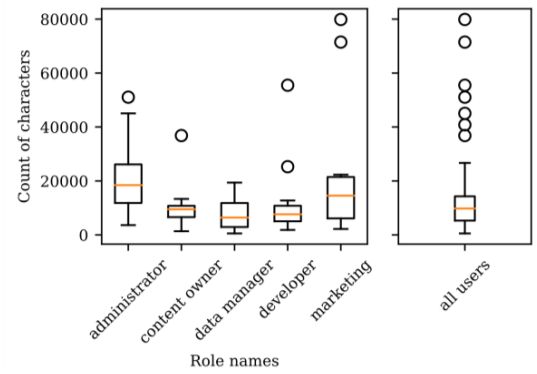


Fig. 7. Textual content added by roles and overall, including ticket descriptions, comments, and text content from attached files.

Throughout the semester, students provided insights into their achieved results in four reports corresponding to each phase. The submitted texts underwent dictionary-based sentiment analysis. In the first phase, students exhibited a positive attitude towards the task, with relatively high variability. In the second phase, the average sentiment measured in the reports continued in the third and fourth phases. Given the challenging nature of the tasks in the fourth phase, one might expect a further decrease in the observed sentiment in the reports.

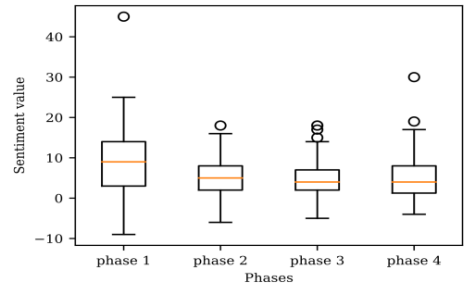


Fig. 8. Results of sentiment analysis of performance reports by students for each project phase.

However, the analysis did not indicate this, although there was a slight increase in variability. The results are summarized in Fig. 8.

The ticketing system is fundamentally designed to facilitate collaboration, making it suitable for successful application of network analysis tools. Since the five-member teams only worked within their own projects and the support project, we cannot speak of the emergence of a larger network. In terms of the entire ticketing system, the relationship graph is not connected. We consider two users to have a connection if they collaborated related to the same ticket. Within the project, the maximum achievable degree is equal to the team's size. As team sizes vary, the achieved degree is divided by the team size, resulting in a corrected degree. As shown in Fig. 9, half of the students worked together with all their teammates, but more than 10% left no trace of collaboration within the ticketing system. This could either have taken place on an external platform or not at all.

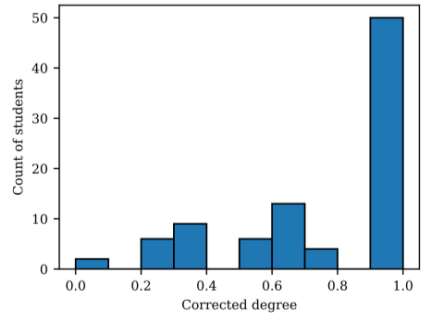


Fig. 9. Frequency of students with degrees corrected for team size. Nearly half of the students did not communicate with every teammate.

Network analysis can provide an additional dimension to characterize the roles. Consider Fig. 10, where we can observe the average weighted degrees calculated based on the connections established between the different roles. The figure shows which role collaborations were prominent: marketers typically collaborated with content owner role users, while developers collaborated with data managers.

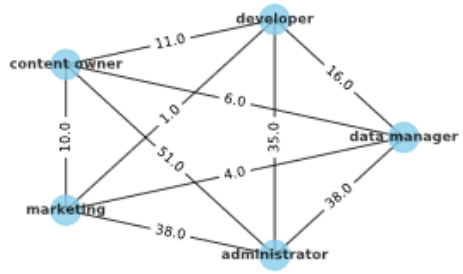
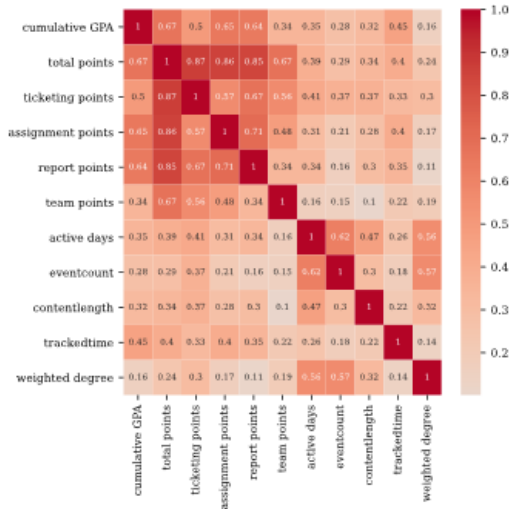


Fig. 10. Median of weighted degree between roles, based on the frequency of communication and collaboration.

One of the primary goals of this research is to propose a method for objectively grading the work of the students. This does not simply consist of evaluating the results as grades should reflect the management of the ticketing system and the whole process as well. During the examined semester, the metrics of the ticketing system played a marginal role in determining the points for this aspect. Due to the lack of benchmarking data, instructors primarily evaluated students' performance subjectively. Therefore, the grades assigned during the semester are not directly suitable for building a grading model. The distribution of points for the ticketing system management is skewed to the right (Fig. 11). This distribution shape can be attributed to the pilot nature of the course evaluation and the lack of previous experience (as it was the first year of the course). Evaluators might have leaned towards assigning more favourable scores to the students. In terms of the other components of the points, it can be said that while the distribution of points for assignments is closer to normal, the points for reports and particularly for teamwork are significantly high.

Improving the grading system requires an understanding of the details of the current subjective scoring. Fig. 12 shows the correlation matrix of our score variables and some of the measurements of the ticketing system. The total project score (total points) of the students has rather strong correlation with the cumulative grade point average (CGPA) from other courses, thus it is most likely that the subjective scoring managed to match the skill and effort of the students.

The correlation between the partial scores (ticketing points, assignment points, report points, team points) is relatively high as well, with the sole exception of the team points, since the other three focused on the individual. This means that even though these parts were graded separately both the quality of the work and the evaluation of the teachers stayed consistent.



Fig, 12. The correlation between the C GPA, total points received during the project course, the components of these points, and the extracted indicators.

The ticketing measurements are definitely not independent. The strongest connection is between the number of days a student was active on in the ticketing system (active days) and the events they generated in the system during this time (eventcount). These also have relatively strong correlations with the strength of the connection between the team members (weighted_degree). This variable was calculated based on the number of events which could be connected to other team members (e.g.: direct mentions, posts in the same task etc.) and the value was weighted based on the size of the team. It is important to note that this variable could be misleading as of course the ticketing system was not the only platform of communication between team members. The total length of text content created by a user (contentlength) and total time they reported spending on their tasks (trackedtime) on the other hand are only weakly related to the other variables. This seems reasonable as these are the only two variables which do not increase naturally the more granulated the activity is.

Comparing the scores with the measurements, it is natural that the ticketing point has the strongest correlation with each. The assignment point is close second, thus the better someone managed to keep track of their work, the more likely they managed to finish their tasks in time. The reports are similar as the more actual effort was put into the project the easier it is to write a good report about it. The measurements have very little connection with the team points, since these variables

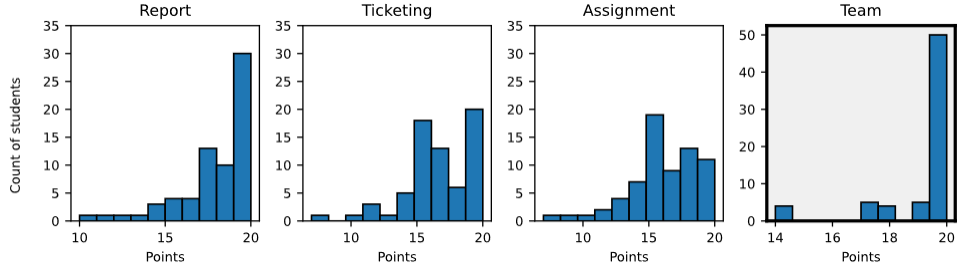


Fig. 11. Frequency of points for individual and team performance, detailing points given for report, ticketing, assignment, and team contributions

measure the continuous activity, while team points were only awarded for the completion of the tasks.

VII. Discussion

We showed that the scoring system led to results which correlate with the students' grades from other courses. Thus, it is safe to assume that the results of the subjective scoring method represent the skill level and effort of the individuals. So, the decision-making process of the teachers is correct, and an objective approach can be implemented by finding the driving factors behind their decisions.

Based on the correlation matrix of the scores and the measured variables we can conclude that the teachers granted higher scores for more granulated activities and were less influenced by the total length of the content or the total time tracked.

The overall activity patterns of the students seem to be similar along the roles (with the exception of the administrator), even though they have vastly different tasks. This indicates that the team members tend to work together, and they are also influenced by each other's habits. However, the activity and behaviour of the individual roles tends to differ significantly, but the current measurements cannot show these differences. The GitHub activity of the developers was not included, nor did we consider the attached images. For example, the content owners usually work with screenshots and design plans, while the marketing tends to create documents in which they collect their findings. Thus, appropriate, automated scoring requires either the inclusion of different types of measurements or the targets for each role must be set separately. The data from version control systems may contain valuable additions for understanding the behavior of the developer and data manager roles, as highlighted by related studies [27], [28].

The event count and granularity along with the stored content allows us to examine the work process and team dynamic of the students. It is clearly visible what steps the students in different roles took to produce results which satisfy the requirements set by their tasks. The medians of these role-specific measurements can be used as a benchmark for the objective scoring limits in the next iteration of the course.

The course aims to prepare students for real-life scenarios where continuous work is expected. However, even the most prominent students tend to leave tasks to the last minute. This behaviour has a negative effect in the long term: tasks tend to pile up, overtime is often needed to meet deadlines, stress and lack of regularity results in health issues and usually leads to burnout. In terms of the course the teammates (especially the administrator) can end up in situations which are difficult to handle, since they might need someone else's work finished to start their own. Even though teaching students how to handle these kinds of conflicts is part of the course, reducing their numbers would still be beneficial. Thus, the new formative grading system should include aspects which reward continuous work and sanction the procrastination mentality.

To address the procrastination behaviour of the students, the ticketing system points can be spread out into weekly scores. The timestamps would already support this approach, but continuous feedback on whether the weekly input of someone was satisfactory or not could help with the motivation as well. Smaller blocks of scoring are also easier to automate, as students have a better understanding of their results, and it is easier for them to argue about it. This can either lead to the students' better understanding of their mistakes and overall improvement of their performance, or we could realize a potential mistake in the way the scores are calculated. Thanks to the steady workload, the activity of the students will be naturally granulated. Thus, they have more time to reflect on their decisions or to improve on a good idea. So, they will automatically focus on aspects which were rewarded by the subjective scoring of the teachers.

The use of a ticketing system is beneficial for trainers because it can provide a good picture of the actual value added by students working in a team, but this is only a side benefit: its main purpose is to facilitate productivity and communication. Thus, network characteristics are explicitly recommended to be considered and rewarded. Working in a group can have a positive influence on the performance of the individual, as people tend to push each other further. Based on the final projects and reports we assume that most teams worked together well, and members used different methods to communicate with each other constantly. However, students who had a higher weighted degree, that is those who communicated through the ticketing system as well, achieved higher scores. It is likely that having the communication documented exerted pressure on people to do a good job with their tasks, since having multiple unanswered comments and notices from other team members in a phase made it clear who was slacking. So, the more access instructors have to project related communication, the less likely it is that someone holds back the team.

So simply making the related communication visible for a figure of authority could improve overall performance. To address this issue in the next iteration of the course, we intend to provide a communication platform – preferably with transcript enabled VoIP options – which the teachers could access as well. By explaining to the students that this can help them in case of conflicts or if they forget to document important information in the ticketing system, we hope to turn this platform into their preferred tool without making it a requirement.

The success of data collection can be crucial for conducting other studies aimed at studying behavior. For example, if the system is completely transparent, it opens up the possibility for the application of peer grading [2], as well as for studying the confirmed interaction between emerging de facto leadership roles and team performance [31]. As observed, the ticketing system enables tracking of who most frequently delegates tasks, communicates with others, or resolves problems themselves, which can lead to insights about the leadership style of the administrator role and its impact on team dynamics.

The success of data collection can be crucial for conducting other studies aimed at studying behavior. For example, if the system is completely transparent, it opens up the possibility for the application of peer grading [2], as well as for studying the confirmed interaction between emerging de facto leadership roles and team performance [31]. As observed, the ticketing system enables tracking of who most frequently delegates tasks, communicates with others, or resolves problems themselves, which can lead to insights about the leadership style of the administrator role and its impact on team dynamics.

It is important to reduce the chance of conflict in the teams, since this kind of project requires creativity, and it works best when participants get into a flow state while working, which is significantly easier if they like the tasks. The reports showed a relatively positive attitude from the students throughout the semester. In a real work environment, the enthusiasm tends to drop by the end of the project, since by that time the creative processes are usually overtaken by the tedious final tasks. However, a real "passion project" brings satisfaction with every single step. We aim to create a learning environment in which students can experience this kind of feeling of success.

A flexible system is necessary as students' creativity should not be hindered by it in such a project course. Gamification and GBAF are great tools to include limitations while not reducing the enthusiasm of the participants [10]. The kind of role-playing aspect of the course can already be considered a type of gamification, but including challenges and competitions between teams could further improve the experience. The current course wraps up with the team presentations at the end of the last phase. Next time, we will give the teams an opportunity to vote for the best projects after everyone finished. Not only will we provide extra points for the teams with the most votes, we will also honor those who managed to vote for the best teams, since judging the quality of someone else's project is an important skill to develop as well. This feature could be implemented after the closure of the former phases as well.

VIII. Conclusion

We examined the data collected in the ticketing system throughout the semester of our experimental, group project course with the aim of rethinking the scoring system and overall improvement of the course material. It can be concluded that the ticketing system database can be a good starting point for the development of an objective scoring system, but further steps are needed to measure real activity. Objective scores have a better chance of being suitable for automated evaluation with the help of custom developed plugins. So not only are they fairer, but they can also reduce the workload of the teachers, who can use this additional capacity to personalize support for the teams.

Based on our findings, we have made the following recommendations to improve the course:

- The median of event count and granularity can be used as a benchmark for objective scoring limits, but the ticketing system points should also be granted separately for every week to discourage procrastination.
- Including a built-in platform for project related communication and emphasizing the importance of documenting thought processes can further improve transparency and allows better feedback from the instructors.
- Introducing competition through gamification can significantly improve engagement while also providing an incentive for students to familiarize themselves with the work and ideas of other teams.

Overall, after the next iteration of the course data from both approaches will be available, and it will be possible to compare the results. However, right now the positive influence of the above changes is yet to be verified, and they might introduce loopholes. Since the current subjective scoring produced acceptable results, we only plan to base about half of the point on objective – hopefully automatically calculated – criteria. The rest will allow us to be flexible and compensate for any unintended effect. The development process of this course will most likely see many more iterations, but the more data is collected, the better, statistically verified improvements can be made.

References

[1] D. Kokotsaki, V. Menzies, and A. Wiggins, "Project-based learning: A review of the literature," *Improv. Sch.*, vol. 19, no. 3, pp. 267–277, 2016, DOI: 10.1177/1365480216659733.

[2] H. B. Shishavan and M. Jalili, "Responding to student feedback: Individualising teamwork scores based on peer assessment," *Int. J. Educ. Res. Open*, vol. 1, no. November, p. 100 019, 2020, DOI: 10.1016/j.ijedro.2020.100019.

[3] P. Black and D. Wiliam, "Classroom assessment and pedagogy," *Assess. Educ. Princ. Policy Pract.*, vol. 25, no. 6, pp. 551–575, 2018, DOI: 10.1080/0969594X.2018.1441807.

[4] M. Prince, "Does active learning work? A review of the research," *J. Eng. Educ.*, vol. 93, no. 3, pp. 223–231, 2004, DOI: 10.1002/j.2168-9830.2004.tb00809.x.

[5] V. R. . Neufeld and H. S. Barrows, "The 'McMaster Philosophy': an approach to medical education," *Acad. Med.*, vol. 49, no. 11, pp. 1040–50, 1974.

[6] N. M. Webb, "The teacher's role in promoting collaborative dialogue in the classroom," *Br. J. Educ. Psychol.*, vol. 79, no. 1, pp. 1–28, 2009, DOI: 10.1348/000709908X380772.

[7] M. Sweet and L. K. Michaelsen, *Team-based learning in the social sciences and humanities: Group work that works to generate critical thinking and engagement*. Stylus Publishing, LLC., 2012.

[8] P. Haidet, K. Kubitz, and W. McCormack, "Analysis of the Team-Based Learning Literature: TBL Comes of Age," *J. Excell. Coll. Teach.*, vol. 25, pp. 303–333, 2015, DOI: 10.1053/j.gastro.2016.08.014.CagY.

[9] P. Sancho, R. Fuentes-Fernández, P. P. Gómez-Martín, and B. Fernández-Manjón, "Applying multiplayer role-based learning in engineering education: Three case studies to analyze the impact on students' performance," *Int. J. Eng. Educ.*, vol. 25, no. 4, pp. 665–679, 2009.

[10] C. Udeozor, P. Chan, F. Russo Abegão, and J. Glassey, "Game-based assessment framework for virtual reality, augmented reality and digital game-based learning," *Int. J. Educ. Technol. High. Educ.*, vol. 20, no. 1, p. 36, Jun. 2023, DOI: 10.1186/s41239-023-00405-6.

[11] W. M. Davies, "Groupwork as a form of assessment: Common problems and recommended solutions," *High. Educ.*, vol. 58, no. 4, pp. 563–584, 2009, DOI: 10.1007/s10734-009-9216-y.

[12] A. Roskosa and D. Rupniece, "Advantages and Drawbacks of Using Group Work in Translator Training," *Procedia - Soc. Behav. Sci.*, vol. 231, no. May, pp. 244–250, 2016, DOI: 10.1016/j.sbspro.2016.09.098.

[13] Y. Chang and P. Brickman, "When group work doesn't work: Insights from students," *CBE Life Sci. Educ.*, vol. 17, no. 3, pp. 1–17, 2018, DOI: 10.1187/cbe.17-09-0199.

[14] R. Kłeczek, M. Hajdas, and S. Wrona, "Wicked problems and project-based learning: Value-in-use approach," *Int. J. Manag. Educ.*, vol. 18, no. 1, p. 100 324, 2020, DOI: 10.1016/j.ijme.2019.100324.

[15] G. N. Vik, "Doing More to Teach Teamwork Than Telling Students to Sink or Swim," *Bus. Commun. Q.*, vol. 64, no. 4, pp. 112–117, 2001.

[16] M. Freeman, "To adopt or not to adopt innovation: A case study of team-based learning," *Int. J. Manag. Educ.*, vol. 10, no. 3, pp. 155–168, 2012, DOI: 10.1016/j.ijme.2012.06.002.

[17] J. Chen, M. Wang, P. A. Kirschner, and C. C. Tsai, "The Role of Collaboration, Computer Use, Learning Environments, and Supporting Strategies in CSCL: A Meta-Analysis," *Rev. Educ. Res.*, vol. 88, no. 6, pp. 799–843, 2018, DOI: 10.3102/0034654318791584.

[18] K. D. M. Aglibar, G. C. T. Alegre, G. I. Del Mundo, and K. F. O. Duro, "Ticketing System : A Descriptive Research on the Use of Ticketing System for Project Management and Issue Tracking in IT Companies," *Int. J. Comput.* Sci. Res., no. 7, pp. 1–10, 2022, DOI: 10.25147/ijcsr.2017.001.1.90.

[19] F. Gohil and V. Kumar, "Ticketing System," *Int. J. Trend Sci. Res. Dev.*, vol. 3, no. 4, pp. 155–156, 2019.

[20] P. Dhoolia et al., "A cognitive system for business and technical support: A case study," *IBM J. Res. Dev.*, vol. 61, no. 1, pp. 74–85, 2017, DOI: 10.1147/JRD.2016.2631398.

[21] R. Al-Shabandar, A. Jaddoa, T. A. Elwi, A. H. Mohammed, and A. J. Hussain, "A Systematic Review for the Implication of Generative AI in Higher Education," *Infocommunications J.*, vol. 16, no. 3, pp. 31–42, 2024, DOI: 10.36244/ICJ.2024.3.3.

[22] M. Saqr and A. Alamro, "The role of social network analysis as a learning analytics tool in online problem based learning," *BMC Med. Educ.*, vol. 19, no. 160, pp. 1–11, 2019.

[23] M. Ruppel, "Automatic Extraction of Design Decision Relationships from a Task Management System Automatische Extraktion Design von Entscheidungsbeziehungen aus einem Task Management System Automatic Extraction of Design Decision Relationships from a Task Management S," Der Technischen Universität München, 2017.

[24] C. (2016). Andrade, A., Saleh, A., Rehack, A.M., Gomoll, A.S., Danish, J. A., & Hmelo-Silver and C. Paper, "Exploring a Text-Mining Approach for the Analysis of Computer Collaborative Data from a Design-Based-Research Project," *Am. Educ. Res. Assoc.*, no. December, 2016.

[25] T. Takeuchi, K. Suwa, H. Tamura, T. Narumi, T. Tanikawa, and M. Hirose, "A Task-Management System using Future Prediction Based on Personal Lifelogs and Plans," in *UbiComp 2013 Adjunct - Adjunct Publication of the 2013 ACM Conference on Ubiquitous Computing*, 2013, pp. 235–238, DOI: 10.1145/2494091.2494166.

[26] A. Seedahmed, A. Osman, A. Seedahmed, and A. Osman, "Evaluating Employee Performance using Automated Task Management System in Higher Educational Institutions," *Indian J. Sci. Technol.*, vol. 12, no. 9, 2019, DOI: 10.17485/ijst/2019/v12i9/142149.

[27] D. Perera, J. Kay, I. Koprinska, K. Yacef, and O. R. Zaane, "Clustering and sequential pattern mining of online collaborative learning data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 6, pp. 759–772, 2009, DOI: 10.1109/TKDE.2008.138.

[28] B. Fernandez-Gauna, N. Rojo, and M. Graña, "Automatic feedback and assessment of team-coding assignments in a DevOps context," *International Journal of Educational Technology in Higher Education*, vol. 20, no. 1. 2023, DOI: 10.1186/s41239-023-00386-6.

[29] Web Technology Surveys, "DNN technology usage statistics." https://webtechsurvey.com/technology/dnn (accessed Jun. 15, 2024).

[30] DNN Software, "DNN.Platform," 2024. https://github.com/dnnsoftware/Dnn.Platform.

[31] L. Kovács, E. Lénárd, and D. Losonci, "A vezetői magatartás hatása a teljesítményjavulásra a feldolgozóiparban," *Vez. - Budapest Manag. Rev.*, vol. 50, no. 2, pp. 41–58, 2020.

[32] A. I. Adekitan and O. Salau, "The impact of engineering students' performance in the first three years on their graduation result using educational data mining," *Heliyon*, vol. 5, no. 2, p. e01250, 2019, DOI: /10.1016/j.heliyon.2019.e01250.

[33] T. Hardeniya and D. A. Borikar, "Dictionary Based Approach to Sentiment Analysis - A Review," *Int. J. Adv. Eng. Manag. Sci.*, vol. 2, no. 5, pp. 317–321, 2016.

[34] M. K. Szabó, "Experiences of creation of a Hungarian sentiment lexicon," in *Nyelv, kultúra, társadalom*, 2014, pp. 219–226.

[35] Precognox, "Our hungarian sentiment lexicon is available on opendata.hu," 2016. https://www.precognox.com/blog/our-hungarian-sentiment-lexicon-is-available-on-opendata-hu/.

[36] J. Ooms, "hunspell," *R package version*, 3. 2018.

Melinda Magyar was born in Budapest, Hungary, in 1980. She received her B.Sc. and M.Sc. degrees in Business Informatics from Corvinus University of Budapest, Hungary, in 2018 and 2020, respectively, and is currently pursuing a Ph.D. degree in Business Informatics at the same institution.

She is a Senior Lecturer in the Department of Computer Science at Corvinus University of Budapest, where she applies her professional experience in the field of information systems. Her current research interests include database design, programming, data mining, and online communities.

Dávid Burka received his Ph.D. in Business Informatics at Corvinus University of Budapest, Hungary, in 2019, and has been teaching there ever since as an Assistant Professor in the Department of Computer Science.

His research focuses on simulation modelling and data analysis in multiple fields like neuroscience, forecasting of demographic processes or pension systems. He is currently working on developing new teaching methods for the Business Informatics faculty.

Performance Evaluation of MOSM Method on Resource-Constrained Multi-Objective Multi-Project Scheduling Problems

Krisztián Mihály, Gyula Kulcsár, and Mónika Kulcsárné-Forrai

***Abstract*—This paper presents an extended model to address complicated project scheduling challenges, which deals with diverse resources, projects, and tasks characterized by unique attributes, interdependencies, and constraints. Moreover, the model can accommodate a variable system of various objective functions, whose elements can include customized optimization direction and priority value. To tackle these extended problems, we define an optimization model and apply a novel decision- making framework, which integrates metaheuristic search strategies, constructive algorithms, and multi-objective relative comparison models. Measurements were executed on created multi-objective resource-constrained multi-project scheduling to evaluate the performance of the proposed method. A new test problem containing 20 projects has been developed. The J30 series from the PSLIB benchmark set were utilized and we defined deadlines for selected projects. During the tests, 6 objective functions were investigated. The priorities of these objective functions were adjusted prior to each examination. In this paper the results of these performance tests are summarized. The obtained results demonstrate that the proposed method is effectively capable of solving multi-objective, multi-project scheduling problems.**

***Index Terms*—Resource Constrained Project Scheduling (RCPSP), Many Objective Search Method (MOSM)**

I. Introduction

PROJECT SCHEDULING is a crucial aspect of managing various systems and environments, as evidenced by numerous review papers available on the subject. Based on these reviews, we identified the resource-constrained project scheduling problem (RCPSP) as a suitable starting model for our investigation. The first optimization model for RCPSP was introduced by Pritsker in [1]. RCPSP problem proved as an NP-hard problem was done by Blazewicz at all. [2].

In the literature, several survey papers on RCPSP have been published, for example [3],[4], and [5]. Even though the original RCPSP model has been known for a relatively long time and it is sufficiently powerful for many cases, the practical applications require further extensions. An updated overview of various extensions of RCPSP model [6]categorized the model variants using the following main aspects: generalization of the activity, alternative precedence constraints and network characteristics, and consideration of multiple projects [6].

Our model simultaneously addresses two primary extension categories: multiple objectives and multiple projects. When optimization problems involve more than three objectives, they are often termed many-objective optimization problems by researchers [7]. The literature includes methods for addressing multiple objective functions concurrently [8].

To solve multi-objective or many-objective project scheduling problems three main patterns have been widely used by researchers: application of a weighted combination of objective functions, Pareto-efficiency and ordered objectives. Combination of minimization of makespan and costs is applied by Dai et al. [9] and by Schnabel et al. [10]. Considering optimization on individual project level and combined project portfolio level has been presented [11]. Pareto-efficiency considers a few objective functions as the main driver for the key performance indicators. Tabrizi defined two objectives; first is the combination of project completion time and due date, and the second one is called the ecological impact of orders [12]. Dridi et al. consider makespan and cost minimization related to renewable resources [13].

Many researchers propose new models and algorithms to solve different industrial optimization problems. This category includes, for example, vehicle routing [14], workload control [15], process control [16], worker assignment [17], project progress evaluation [18], layout optimization [19], supply chain optimization [20], indoor access point networks [21], encryption [22], process mining [23], production planning [24], production scheduling [25], and many other problems.

Several heuristics have been proposed to solve the RCPSP, classified into three main groups: single-pass heuristics, multi-pass heuristics, and metaheuristics. Single-pass and multi-pass heuristics quickly generate feasible schedules using serial and parallel schedule generation schemes (GS) with various priority rules. Metaheuristics, such as genetic algorithms, simulated annealing, tabu search, artificial immune algorithms, particle swarm optimization, bee colony optimization, and ant colony optimization, employ more complex algorithms to thoroughly explore the solution space. While metaheuristics can find high-quality solutions, they typically require more computing time.

Submitted on 01/06/2024
Krisztián Mihály, Gyula Kulcsár, and Mónika Kulcsárné-Forrai, Department of Information Engineering, University of Miskolc, Miskolc, Hungary (E-mail: krisztian.mihaly@uni-miskolc.hu – correspondence author, gyula.kulcsar@uni-miskolc.hu, monika.kulcsarne@uni-miskolc.hu)

DOI: 10.36244/ICJ.2025.5.7

The primary objective of our research was to develop an extended optimization model and a decision-making framework for effectively solving diverse multi-objective, multi-project scheduling problems. In this paper, we focus on examining and evaluating the performance of the proposed solution method. To demonstrate the flexibility and effectiveness of the method, we present numerical results.

The remaining part of the paper is structured as follows: Chapter II formulates the mathematical model, providing the theoretical framework for our study. Chapter III details the solution approach, outlining the methodologies and algorithms employed. Chapters IV and V present some numerical results, illustrating the application and validation of the models and methods. Finally, Chapter VI forms the conclusion, summarizing the key findings and addressing potential directions for further work.

II. Mathematical Model

The multi-project, multi-objective resource constrained scheduling problem is an extension of the classical Resource Constrained Project Scheduling problem and it is formulated as following: (A) Input data, (B) Indirect values calculated from input data, (C) Constraints for input data, (D) Primary decision variables, (E) Auxiliary values calculated from the primary decision variables, (F) Constraints for the solutions, (G) Objective functions.

A. Input data

The multi project scheduling problem is described with the input data presented in Table I.

TABLE I
Input data

Symbol	Description	Definition
$\boldsymbol{P}$	set of projects to be executed	$\boldsymbol{P} = \{p_1, p_2, \dots, p_i, \dots, p_{NP}\}$
$\boldsymbol{T}$	set of tasks to be executed	$\boldsymbol{T} = \{t_1, t_2, \dots, t_j, \dots, t_{NT}\}$
$\boldsymbol{R}$	set of available resource types	$\boldsymbol{R} = \{r_1, r_2, \dots, r_k, \dots, r_{NR}\}$
RC_k	the predefined maximal capacity of resource type r_k	$RC_k \in \mathbb{Z}_0^+$
$rr_{j,k}$	the amount of resource units required by task t_j from resource type r_k	$rr_{j,k} \in \mathbb{Z}_0^+$
$pt_{j,k}$	the processing time of task t_j on the resource type r_k	$pt_{j,k} \in \mathbb{Z}_0^+$
$\boldsymbol{TA}$	set of assigned pairs of projects and tasks	$\boldsymbol{TA} = \{(p_i, t_j) \mid \forall t_j \in \boldsymbol{T}\}$
$\boldsymbol{PRE_j}$	set of predecessor tasks of task t_j	$\boldsymbol{PRE_j} \subset \boldsymbol{T} \mid \forall t_j \in \boldsymbol{T}$
$\boldsymbol{TPROP}$	set of available task arguments	$\boldsymbol{TPROP} = \{\boldsymbol{T_{rTime}}, \boldsymbol{T_{dTime}}, \boldsymbol{T_{prio}}, \dots, \boldsymbol{T_{prop}}\}$
$\boldsymbol{TP_j}$	set of maintained properties of task t_j	$\boldsymbol{TP_j} = \{T_{rTime,j}, T_{dTime,j}, T_{prio,j}, \dots, T_{prop,j}\}$
$\boldsymbol{PPROP}$	set of available project arguments	$\boldsymbol{PPROP} = \{P_{rTime}, P_{dTime}, P_{tCost}, P_{prio}, \dots, P_{prop}\}$
$\boldsymbol{PP_i}$	set of maintained properties of task p_i	$\boldsymbol{PP_i} = \{P_{rTime,i}, P_{dTime,i}, P_{tCost,i}, P_{prio,i}, \dots, P_{prop,i}\}$

B. Indirect values calculated from input data

From input data additional data can be calculated, which helps to formulate constraints and objective functions in a condensed way. Symbols and definition of the indirect values are presented in Table II.

TABLE II
Input data

Symbol	Description	Definition
$\boldsymbol{PT_i}$	set of tasks assigned to project p_i	$\boldsymbol{PT}_i = \{t_j \mid (p_i, t_j) \in \boldsymbol{TA}\}$
$\boldsymbol{A}$	set of ordered pairs of tasks by prerequisite definition	$\boldsymbol{A} = \{(t_j, t_{pre}) \mid t_j \in \boldsymbol{T}, t_{pre} \in \boldsymbol{PRE}_j\}$
G	directed graph of tasks	$G = (\boldsymbol{T}, \boldsymbol{A})$
$TP(t_j, T_{prop})$	function to get T_{prop} property of task t_j	
$PP(p_i, P_{prop})$	function to get P_{prop} property of project p_i	

C. Constraints for input data

Table III lists the constraints to be fulfilled by the input data.

TABLE III
Constraints for input data

Description	Definition
Negative capacity of a resource type is not allowed.	$RC_k \geq 0 \mid \forall r_k \in \mathrm{R}$
Negative processing time is not allowed.	$pt_{j,k} \geq 0 \mid \forall t_j \in \boldsymbol{T}$
Resource requirement cannot exceed the available maximum resource type capacity.	$0 \leq rr_{j,k} \leq RC_k \mid \forall t_j \in \boldsymbol{T}, \forall r_k \in \boldsymbol{R}$
Task cannot exist without project assignment.	$(p_i, t_j) \in \boldsymbol{TA} \mid \forall t_j \in \boldsymbol{T}$
Task cannot be a predecessor for itself.	$\boldsymbol{PRE_j} \subseteq (\boldsymbol{T} \setminus \{t_j\})$
No circle is allowed in the task dependencies.	Graph G shall fulfil the requirement of directed acyclic graph.

D. Primary decision variables

The main decision variables are the start time of the tasks. Symbols of decision variables are defined in Table IV. These values are non-negative, integer values that describes a solution for the problem. The application area can define the mapping between the integer values to applicable time units such as seconds, days, weeks, and tasks.

TABLE IV
Primary decision variables

Symbol	Description	Definition
S_j	start time of task t_j	$S_j \in \mathbb{Z}_0^+$
SCH	start vector representing start time of each task	$SCH = (S_1, S_2, \dots, S_j, \dots, S_{NT})$

E. Auxiliary values calculated from the primary decision variables

Additional auciliary values are defined in Table V., which can be calculated from the primary decision variables and the input data, allowing the constraints and objective functions in the mathematical model of the RCMOMPSP problem to be more simply formulated.

TABLE V
AUXILIARY VALUES CALCULATED FROM THE PRIMARY DECISION VARIABLES

Symbol	Description	Definition
C_j	completion time of task t_j	$C_j \in \mathbb{Z}_0^+$
C_{max}	maximum completion time of the tasks	$C_{max} = max_{j \in \boldsymbol{T}} (C_j)$
$\boldsymbol{RT}(\tau)$	set of running tasks at time τ	$\boldsymbol{RT}(\tau) = \{t_j \in \boldsymbol{T} \mid S_j \leq \tau \leq C_j\}$
$\lambda_{j,k}(\tau)$	indicates that task t_j is processed on resource r_k at time τ	$\lambda_{j,k}(\tau) = \begin{cases} 1, if\ S_j \leq \tau \leq S_j + pt_{j,k} \\ 0, otherwise \end{cases}$
$RL_k(\tau)$	resource load at r_k at time τ	$RL_k(\tau) = \sum_{j \in \boldsymbol{RT}(\tau)} \lambda_{j,k}(\tau) * rr_{j,k}$
$\lambda_{tdelayed}(i)$	indicates that task t_j is delayed	$\lambda_{tdelayed}(j) = \begin{cases} 1, if\ TP(t_j, T_{dTime}) > 0 \wedge \\ \quad C_j > TP(t_j, T_{dTime}) \\ 0, otherwise \end{cases}$
$TL(j)$	task latency	$TL(j) = \lambda_{tdelayed}(j) * (C_j - TP(t_j, T_{dTime}))$
$\lambda_{pdelayed}(i)$	indicates that project p_i is delayed	$\lambda_{pdelayed}(i) = \begin{cases} 1, if\ PP(p_i, P_{dTime}) > 0 \wedge \\ \max_{j \in PT_i}(\{C_j\}) > PP(p_i, P_{dTime}) \\ 0, otherwise \end{cases}$

F. Constraints for the solutions

A schedule is a feasible solution if the constraints defined for the solution are not violated. The constraints are represented in Table VI.

TABLE VI
CONSTRAINTS FOR THE SOLUTION

Description	Definition
A task cannot be started until all predecessor tasks have been completed.	$S_j \geq \max_{pre \in \boldsymbol{PRE}_j}(\{C_{pre}\})$
If task has predefined release time it cannot be started earlier.	$S_j \geq TP(t_j, T_{rTime})$
If project p_i has predefined release time, then any of the assigned tasks cannot be started earlier.	$S_j \geq max_{(p_i,t_j) \in \boldsymbol{TA}} (PP(p_i, P_{rTime}))$
Load of any resource type cannot exceed the maximum capacity of the resource type at any time.	$RL_k(\tau) \leq RC_k \mid \forall r_k \in \boldsymbol{R}, 0 \leq \tau \leq C_{max}$

G. Objective functions

We have designed the optimization model to find the best solution by simultaneously considering multiple objective functions. The main goal of our approach is to ensure that the solving method does not rely on specific information about the objective functions; we aim to use a general model that is completely independent of the actual set of objective functions. However, in this paper, we present six objective functions in Table VII. that were used in the performance measurement.

TABLE VII
OBJECTIVE FUNCTIONS

Symbol	Description	Definition
$L_{max} = f_{PC_{max}}$	maximum lateness	$f_{PC_{max}} = max_{j \in \boldsymbol{T}} (TL(j))$
$T_{max} = f_{TT_{max}}$	maximum tardiness	$f_{TT_{max}} = \max(0, max_{j \in \boldsymbol{T}}(TL(j)))$
$T_{sum} = f_{TT_{sum}}$	sum of tardiness of all tasks	$f_{TT_{sum}} = \sum_{j \in \boldsymbol{T}}(TL(j))$
$U_{sum} = f_{TDC}$	sum of tardy tasks	$f_{TDC} = \sum_{j \in \boldsymbol{T}} \lambda_{tdelayed}(j)$
C_{max}	maximum completion time	$C_{max} = max_{j \in \boldsymbol{T}} (C_j)$
C_{sum}	sum of completion time of all tasks	$C_{sum} = \sum_{j \in \boldsymbol{T}}(C_j)$

III. SOLUTION APPROACH

In a previous work, we proposed a hybrid method to solve extended scheduling problems [23]. The approach to solve scheduling problems consists of two main components: a search module and a scheduling construction module. The process contains three primary decision-making phases: (1) Establishing a priority sequence for tasks to be scheduled in the search module, (2) Incorporating the highest priority task into the schedule while accounting for resource availability and task requirements in the schedule generation, (3) Selecting the optimal schedule from a set of feasible solutions by evaluating each objective function concurrently. The proposed method was named as multi objective search method (MOSM).

A. Search algorithm to construct priority sequence of tasks

Inspired by the Jaya algorithm [27] and considering local search principles, we developed a hybrid approach to use an evolutionary algorithm together with local search techniques. This algorithm features three novelties:

(1) A search strategy that facilitates exploration of the search space. (2) An enhanced procedure for generating new candidate members for the next generation. (3) Utilize multiple objective functions simultaneously with individual priorities and optimization directions. The pseudo-code of the developed MOSM algorithm is presented in Algorithm 1. Notations used in the pseudo-code is listed in the Table VIII.

TABLE VIII
MOSM ALGORITHM NOTATION

Notation	Description
$M_{i,g}$	Control vector of the i^{th} candidate member of the g^{th} generation containing the priority sequence of tasks.
$SC(M_{i,g})$	The feasible schedule generated by the scheduling construction algorithm (SC) considering $M_{i,g}$ control vector.
NM	Number of members in the population
NG	Number of generations

Algorithm 1: Many-Objective Scheduling Method (MOSM)

Input: RCMPMOSP Problem definition, search parameters
Output: Scheduling vector SCH
Begin
1.1 i = 1; g = 1;
1.2 **while** (i <= NM)
1.3 Create the $M_{i,g}$ vector by the parallel schedule generation scheme with the earliest starting task selection rule
1.4 Construct the SC(Mi,g)
1.5 Evaluate the SC($M_{i,g}$)
1.6 Add the $M_{i,g}$ vector as the i^{th} member to the 1^{th} generation;
1.7 i = i + 1
1.8 **end while**
1.9 Select the best member of 1^{th} generation
1.10 g = g + 1;
1.11 **while** (g <= NG)
1.12 i = 1
1.13 limit = 1 – g / NG;
1.14 **while** (i <= NM)
1.15 number = Generate a pseudo random number from the interval [0, 1] with uniform probability;
1.16 **if** (number < limit)
1.17 Create a new candidate $M_{i,g}$ vector by mutating the i^{th} member of the previous generation;
1.18 **else**
1.19 Create a new candidate $M_{i,g}$ vector by mutating the best member of the previous generation;
1.20 Construct the SC($M_{i,g}$) schedule based on the $M_{i,g}$ vector;
1.21 Evaluate the SC($M_{i,g}$) schedule;
1.22 Add the better version of $M_{i,g}$ and $M_{i,g-1}$ as the i^{th} new member to the g^{th} generation;
1.23 i = i + 1
1.24 **end while**
1.25 Select the best member of the g^{th} generation;
1.26 g = g + 1;
1.27 **end while**
1.28 Return best member of the latest generation;
end

B. Construction algorithm to generate a feasible schedule considering task selection control parameters

MOSM consists of a constructive algorithm to generate a feasible schedule. The schedule construction algorithm (SC) begins with an empty schedule. In each iteration, a new, feasible partial schedule is generated by adding one selected task to the existing partial schedule. This method continues until all tasks are scheduled. Algorithm 2 provides the pseudo code for SC.

Algorithm 2: Schedule Construction Algorithm (SC)

Input: RCMPMOSP Problem definition, task selection rules and priority values
Output: Feasible schedule SC($M_{i,g}$)
Begin
2.1 Create an empty schedule ;
2.2 i = 1;
2.3 **while** (i <= number of tasks)
2.4 Choose the t_c task from the i^{th} position of the vector $M_{i,g}$;
2.5 Insert the chosen t_c task into the schedule with the earliest applicable start time;
2.6 i = i + 1;
2.7 **end while**
2.8 Return schedule SC($M_{i,g}$)
End

The MOSM algorithm was compared to the published PSLIB benchmark results using the C_{max} objective function. The experimental result showed a good performance of the applied solution method. Based on this experience, we examined the efficiency of the procedure when applied to multiple projects and multiple objective functions simultaneously.

C. Qualification Method for Comparing Schedules

MOSM can incorporate numerous objective functions. We assume that the set of objective functions is unlimited and can include various elements with different priorities and optimization directions. One scheduling always have one predefined set of applied objective functions. To compare two feasible schedules, represented by two members of a generation the following approach is applied.

Let s_x and s_y be two candidate feasible schedules. The quality of a given schedule is represented by a given vector containing K real numbers, each vector coordinate represents the actual value of the corresponding objective function.

TABLE IX
RELATIVE QUALIFICATION NOTATION

Notation	Description
u	$u = (u_1, u_2, \dots, u_k, \dots, u_K), u_k \in \mathbb{R}$; u denotes the vector containing the values of the objective functions considering the given schedule to be compared.
z	$z = (z_1, z_2, \dots, z_k, \dots, z_K), z_k \in \{-1, 1\}$; z denotes the vector containing the optimization directions of the objective functions. The value of z_k is 1 if we want to minimize the k^{th} objective function. The z_k is -1 if the k^{th} objective function must be maximized.
w	$w = (w_1, w_2, \dots, w_k, \dots, w_K), w_k \in \mathbb{Z}_0^+$; w denotes the vector containing priorities for the objective functions. Each w_k is a non-negative integer value ($w_k \geq 0$) that expresses the importance of the u_k value of the k^{th} objective function.

A distance function D is defined as follows.

$$D : \mathbb{R}^2 \to \mathbb{R},\ D(a,b) := \begin{cases} 0, & if\ \max(|a|,|b|) = 0 \\ \frac{b-a}{\max(|a|,|b|)}, & otherwise \end{cases} \quad (1)$$

The relative qualification uses notations defined in Table IX. Let x and y be two vectors with type u. These vectors contain the values of objective functions, and they represent the absolute quality of candidate schedules s_x and s_y to be compared. We define the F function to express the relative quality of y compared to x as a real number.

$$F : u^2 \to \mathbb{R},\ F(x, y) := \sum_{k=1}^{K} (w_k \cdot z_k \cdot D(x_k, y_k)) \quad (2)$$

IV. MULTI-OBJECTIVE, MULTI-PROJECT BENCHMARK SET

A necessary element for conducting the tests of the solution approach is to have multi-project, multi-objective scheduling problems available. For this, the J30 benchmark scheduling problems documented in the PSLIB benchmark set have been

used as a basis [26]. The instances j301_1, j301_2, j301_3, ..., j302_10 of the PSPLIB J30 series together was considered as a project portfolio. The project portfolio created with this procedure thus contains 20 projects, each with 30 tasks. The operation times, prerequisite constraints, and resource requirements given in the benchmark tasks were unchanged.

To ensure the comparability of each test, all tests use the same initial parameters and state space. During the search, the initial task priority vector is determined using the parallel generation scheme. The generation scheme used the earliest finish time priority rule.

V. Experimental Result

The executed tests are presented in grouped, sequentially organized subsections. Besides detailing the main input parameters of the tests, the test results, and their evaluative descriptions, the step-by-step presentation aims to provide insight into the background of additional conclusions identified during the testing.

A. Test results on individual objective functions

In the first test group, we used the first 20 projects of the PSLIB RCPSP J30 benchmark set. An individual due date (P_{dTime}) has been defined for every fifth project (5th, 10th, 15th, and 20th). Due to the characteristics of the PSLIB benchmark tasks, specifying the completion deadline at the project level can also be modeled by the completion deadline given for the last, virtual closing task of the selected projects. For the selected projects the project completion due date was determined based on the best C_{max} values documented in the PSLIB benchmark results, such that the completion due date of the selected project equals the sum of the C_{max} values of the fewer selected projects, as follows:

$$P_{dTime,i} = \sum_{j \le i} C_{max,j} \mid i,j \in \{5,10,15,20\} \quad (3)$$

The primary aim of the executed tests is to verify the controllability of the search by assigning a non-negative priority to a single designated objective function. The secondary aim of the tests is to use the objective function values obtained from the individual objectives as reference values for evaluating the results of subsequent tests.

Test results are presented in the Table X.

TABLE X
Individual Objective Function Priorities and Calculated Objective Function Values

	Objective function priority						Objective function value					
	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}
1	1	0	0	0	0	0	-8	0	0	0	321	79707
2	0	1	0	0	0	0	0	0	0	0	302	79557
3	0	0	1	0	0	0	0	0	0	0	311	80470
4	0	0	0	1	0	0	256	256	436	2	323	84605
5	0	0	0	0	1	0	237	237	637	4	276	79837
6	0	0	0	0	0	1	227	227	548	4	289	72393

The search characteristics of the test #1 execution are depicted in Fig 1. The points on the diagram show the improvements in the values of the selected objective function. The horizontal axis represents the number of search steps. It can be observed that within the first 500 iterations, the search module frequently finds better schedules than the previously known solution. Subsequently, the number of iterations required to find better solutions increases. It is noticeable that within the range of 2000-2500 iterations, it can once again make several improvements. The presented figure confirms that the search module follows an expected characteristic.

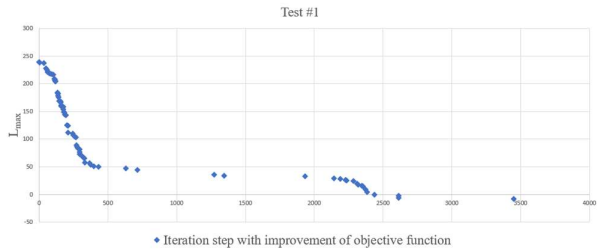


Fig. 1. Magnetization as a function of applied field. Note that "Fig." is abbreviated. There is a period after the figure number, followed by two spaces. It is good practice to explain the significance of the figure in the caption.

B. Test results on multiple objective functions considered simultaneously

In the second test group, I examined the simultaneous use of multiple objective functions. The scheduling task remained unchanged compared to the first test group. I grouped the objective functions according to three criteria and tested the values of different priorities accordingly. The considered groups are as follows:

- Minimizing the delays (L_{max}, T_{max}, T_{sum}, U_{sum})
- Minimizing the completion time of the last task (C_{max})
- Minimizing the sum of task completion times (C_{sum})

Table XI. presents the results of the defined objective function priorities and the calculated objective function values.

TABLE XI
Simultaneous Objective Function Priorities and Calculated Objective Function Values

	Objective function priority						Objective function value					
	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}
7	1	1	1	1	1	1	-7	0	0	0	287	73837
8	0	10	20	8	1	1	0	0	0	0	284	74760
9	0	2	2	2	1	0	0	0	0	0	283	79365
10	5	0	0	0	1	0	-7	0	0	0	280	79193
11	0	0	2	0	0	1	-2	0	0	0	297	73819

During the 7th and 8th tests all three criteria have been considered simultaneously. The aim of these two tests was to determine how close we could approach the minimum of C_{sum} and C_{max} compared to 5th and 6th test. The parameters set for

the 7th test favored C_{sum}, while those for the 8th test favored C_{max}. In the 9th and 10th tests the C_{sum} objective function has not been considered, that is why the priority has been set to zero. It can be observed that this favored the computed results of C_{max} (first 287 and 284, then 283 and 280). In the 10th test series, the best value for C_{max} was influenced by the L_{max} objective function. This generated earliness and resulted in an improvement in C_{max}. In the 11th test, only C_{sum} was considered while adhering to the deadlines. Here, T_{sum} alone managed to keep the deadline overrun at zero. The value of C_{sum} decreased quite well. A better value than this was only achieved in previous tests when the C_{sum} objective function was used independently (6th test).

C. Alternative way to model due date of a project

The due-date modelling of a project was modified in the next test group. In the alternative modelling the project completion due date was set as task due date for every task associated with the given project. The correctness of this modification is not affected because if task-level due dates were not otherwise specified, the project completion deadline can be considered as the maximum task deadline for all tasks within the given project.

Table XII. presents the considered objective function priorities and the calculated objective function values.

TABLE XII
SIMULTANEOUS OBJECTIVE FUNCTION PRIORITIES AND CALCULATED OBJECTIVE FUNCTION VALUES WITH ALTERNATIVE DUE-DATE MODELLING

	Objective function priority						Objective function value					
	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}
12	0	0	0	1	0	0	-2	0	0	0	304	79814
13	1	0	0	0	0	0	-8	0	0	0	321	79707
14	0	1	0	0	0	0	0	0	0	0	302	79557
15	0	0	1	0	0	0	-1	0	0	0	331	80912
16	0	0	10	0	1	0	213	0	0	0	279	79377
17	0	0	10	0	0	1	0	0	0	0	292	73797

The 12th test confirmed the correctness of the assumption made in the second test group. A different formulation of the same problem with the same model and scheduling method leads to better solutions because the search module can make more effective improvements based on feedback on due date violations.

The result of the 13th test completely matches the result of the 1st test. This was expected since, in terms of L_{max} optimization, the individual deadlines of tasks are not significant.

The result of the 14th test completely matches the result of the 2nd test because, in terms of T_{max} optimization, the individual deadlines of tasks are relevant.

In the 16th test, it was possible to further reduce the value of C_{max} by one unit while adhering to the deadlines. In the first test series, the best value was 280, while here the method achieved 279.

In the 17th test, it was possible to decrease the value of C_{sum} while adhering to the due dates. In the first test, the best value was 73819, while in this test it decreased to 73797.

The achieved results confirm that the formulation of the task greatly influences the quality of the solution, thus justifying the use of sophisticated optimization models.

D. Evaluation of tardy projects

In this test group, the investigation focused on how the algorithm handles the scenario where delays are unavoidable. To evaluate this circumstance, in this test group the initial scheduling problem was enhanced with additional data. The due dates of tasks were modified compared to the task used in the second test group. A single common due date was set for all tasks of the four projects with due date. The new due date was 39 time units. This value corresponds to the earliest due date of projects used in the previous test group.

If not all tasks can be completed by due date, then the T_{max}, T_{sum}, and U_{sum} objective functions can become competitors of each other or even reinforce each other's effect. In such cases, the following questions can be formed: If due date overrun is unavoidable, would it be more acceptable to have several small delays or fewer large delays? Alternatively, would it be advisable to minimize the total number of tardy delays?

Table XIII. shows the tested objective function priority considerations and the measured objective function values.

TABLE XIII
MEASUREMENT RESULT OF CERTAINLY DELAYED PROJECTS UNDER DIFFERENT OBJECTIVE FUNCTION PRIORITIES

	Objective function priority						Objective function value					
	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}	L_{max}	T_{max}	T_{sum}	U_{sum}	C_{max}	C_{sum}
18	1	0	0	0	0	0	25	25	553	38	309	80690
19	0	1	0	0	0	0	25	25	553	38	309	80690
20	0	0	1	0	0	0	33	33	492	35	344	82231
21	0	0	0	1	0	0	258	258	2848	28	325	81729
22	0	1	1	1	0	0	30	30	555	37	313	78621
23	0	1	1	10	0	0	35	35	575	29	313	81613
24	0	10	1	1	0	0	26	26	594	41	317	81451
25	0	1	10	1	0	0	29	29	522	39	317	80272

The results of tests 18 and 19 are completely identical. This was expected because if there is a certain delay, then the value of L_{max} is greater than zero and equals the value of T_{max}.

In tests 20, 21, and 22, T_{max}, T_{sum}, and U_{sum} individually provided better results from their own perspectives than the solutions achieved with the other two. When used together in the system, these objective functions can produce compromise solutions. Test 22 provides an example of this situation.

In tests 23, 24, and 25, three selected objective functions were set in such a way that one function has a priority value of 10, while the priority of the other two was set to 1. The results show that the higher priority had an impact on the search, which is reflected in the results as well.

From tests 24 and 25, an important conclusion can be drawn

that appropriately chosen priority values can achieve better results in the given scenario than the values achievable by individually applied objective functions.

VI. Conclusion and Future Work

This paper presented an extended model to solve complicated project scheduling problems. An optimization model is proposed to define the mathematical model of the investigated problem class, which involves diverse resources, projects, and tasks characterized by unique attributes, interdependencies, and constraints. The proposed model can also accommodate a flexible system of various objective functions, whose elements can include customized optimization direction and priority value. A novel decision-making framework is also presented. The solving approach integrates metaheuristic search strategies, constructive algorithms, and multi-objective relative comparison models.

In this paper, the focus was set to performance tests that were executed on a new multi- objective resource-constrained multi-project scheduling problem. To evaluate the performance of the proposed method, six objective functions were investigated. The priorities of these objective functions were adjusted, and the numerical result of the tests were evaluated. The summarized numerical results proved that the presented hybrid method performs very well. The calibrated priorities of the objective functions effectively control the searching of the best values for the active objective functions. The obtained results of the performance evaluation tests executed on the created RCMOMPSP problem express that MOSM method is efficient, robust, and flexible. Therefore, our work has achieved the research objectives. The obtained running results demonstrate that the proposed method is effectively capable of solving multi-objective, multi-project scheduling problems.

The proposed model flexibly adapts to changing optimization objectives. The applied many-objective optimization model based on relative qualification is able to solve effectively the many-aspect decision-making problems that focuses on selecting the next candidates in the search method based on previous evaluated results.

The concrete algorithms used in the decision-making phases can be easily exchanged. An excellently important feature of the investigated approach is that the searching process relies only on problem-independent information. A direct consequence of this fact is that the approach can be used in the case of any scheduling problem. In general, it can be stated that any effective permutation-oriented search metaheuristic algorithm is able to iteratively drives the schedule generation scheme by modifying the control sequence of tasks. The generation scheme has a very important role in this approach. Constructing the complete schedule means a reactive simulation that is encapsulates the constraint of the concrete problem type by making the search algorithm independent of the problem-specific constraints.

In practice, software developers can utilize the advantages of this separated and independent resolution of the necessary decision-making phases that support making up flexible software systems and cyber-physical systems. The wide applicability of the proposed optimization model for the extended many-project scheduling problem class is ensured by its formulation in such a way that the individual or system-oriented objective functions, constraints and attributes can also be taken into account. This paper offers a validated solving method for developers to realize practical scheduling applications for solving a wide variety of scheduling problems.

References

[1] A. A. B. Pritsker, W. J. Lawrence and P. M. Wolfe, "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach," *Management Science*, vol. 16, no. 1, pp. 93–108, Sept. 1969.

[2] J. Blazewicz, J. K. Lenstra and A. H. Kan, "Scheduling subject to resource constraints: classification and complexity," *Discrete Applied Mathematics*, vol. 5, no. 1, pp. 11–24, Jan. 1983, DOI: 10.1016/0166-218X(83)90012-4.

[3] Kolisch, R., Hartmann, S.: "Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis", Project scheduling, Project Scheduling: Recent Models, Algorithms and Applications, 147–178 (1999), DOI: 10.1007/978-1-4615-5533-9_7

[4] Hartmann, S., Kolisch, R.: Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research* 127(2), 394–407 (2000).

[5] Pellerin, R., Perrier N., Berthaut, F.: A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research* 280(2), 395–416, (2020).

[6] Hartmann, S., Briskorn, D.: A survey of variants and extensions of the resource-constrained project scheduling problem," *European Journal of Operational Research* 207(1), 1–14 (2010).

[7] Zhang, Y.-H., Gong, Y.-J., Zhang, J., Ling, Y.-b.: A hybrid evolutionary algorithm with dual populations for many-objective optimization. *IEEE Congress on Evolutionary Computation (CEC)*, (2016).

[8] Taha, K.: Methods That Optimize Multi-Objective Problems: A Survey and Experimental Evaluation. *IEEE Access* (8), 80 855–80 878 (2020).

[9] Dai, H., Cheng, W., Guo, P.: An Improved Tabu Search for Multi-skill Resource-Constrained Project Scheduling Problems Under Step-Deterioration. *Arabian Journal for Science and Engineering* 43(6), 3279–3290 (2018).

[10] Schnabel, A., Kellenbrink, C., Helber, S.: Profit-oriented scheduling of resource-constrained projects with flexible capacity constraints. *Business Research* 11(2), 329-356 (2018).

[11] Tirkolaee, E., Goli, A., Hematian, M., Sangaiah, A., Han, T.: Multi-objective multi-mode resource constrained project scheduling problem using Pareto-based algorithms. *Computing* 101(6), 547–570 (2019).

[12] Tabrizi, B.: Integrated planning of project scheduling and material procurement considering the environmental impacts. *Computers & Industrial Engineering* 120, 103–115 (2018).

[13] Dridi, O., Krichen, S., Guitouni, A.: A multiobjective hybrid ant colony optimization approach applied to the assignment and scheduling problem. *International Transactions in Operational Research* 21(6), 935–953 (2014).

[14] S. H. Huang, J. H. Huang, H. C. Lee and Y. Y. Tong, "A new hybrid algorithm for solving the vehicle routing problem with route balancing," *International Journal of Industrial Engineering and Management*, vol. 14, no. 1, pp. 51–62, March 2023.

[15] P. Renna, "Workload control order release with controllable processing time policies: an assessment by simulation," *International Journal of Industrial Engineering and Management*, vol. 13, no. 3, pp. 194–205, Sept. 2022.

[16] M. K. Adeyeri, S. P. Ayodeji, E. O. Olutomilola and O. J. Abayomi, "The Automated Process Control Model for Energy Consumption Optimization within Plantain Flour Processing Facility," *International Journal of Industrial Engineering and Management*, vol. 13, no. 3, pp. 206–214, Sept. 2022, DOI: 10.24867/IJIEM-2022-3-313.

[17] N. Fernandes, M. Thürer, F. Rodrigues, L. Pinto Ferreira, F. J. G. Silva and P. Avila, "Worker Assignment in Dual Resource Constrained Systems Subject to Machine Failures: A Simulation Study," *International Journal of Industrial Engineering and Management,* vol. 13, no. 2, pp. 110–118, 30 June 2022, DOI: 10.24867/IJIEM-2022-2-305.

[18] D. A. Kurniady, Nurochim, A. Komariah, Turwelis, H. T. Hoi and V. H. Ca, "Construction project progress evaluation using a quantitative approach by considering time, cost and quality," *International Journal of Industrial Engineering and Management*, vol. 13, no. 1, pp. 49–57, March 2022.

[19] S. Gao, J. Daaboul and J. Le Duigou, "Layout and scheduling optimization problem for a reconfigurable manufacturing system," *International Journal of Industrial Engineering and Management*, vol. 12, no. 3, pp. 174–186, Sept. 2021, DOI: 10.24867/IJIEM-2021-3-286.

[20] T. Trisna, M. Marimin, Y. Arkeman and T. C. Sunarti, "Fuzzy multi-objective optimization for wheat flour supply chain considering raw material substitution," *International Journal of Industrial Engineering and Management*, vol. 11, no. 3, pp. 182–191, Sept. 2020.

[21] L. Nagy, "Classical and quantum genetic optimization applied to coverage optimization for indoor access point networks", *Infocommunications Journal*, vol. 4, no. 4, 2012.

[22] K. Kubícek, J. Novotný, P. Švenda, and M. Ukrop, "New results on reduced-round tiny encryption algorithm using genetic programming," *Infocommunications Journal*, vol. 8, no. 1, pp. 2–9, 2016.

[23] L. Kovács, E. Varga and P. Mileff, "Application of Neural Network Tools in Process Mining". *Infocommunications Journal,* vol 15. pp. 13–19. ISSN 2061-2079, 2023, DOI: 10.36244/ICJ.2023.5.3

[24] S. K. Karimi, S. J. Sadjadi and S. G. J. Naini, "A bi-objective production planning for a flexible supply chain solved using NSGA-II and MOPSO," *International Journal of Industrial Engineering and Management*, vol. 13, no. 1, pp. 18–37, March 2022, DOI: 10.24867/IJIEM-2022-1-298.

[25] P. Chetthamrongchai, O. Stepanenko, N. Saenko, S. Bakhvalov, G. Aglyamova and A. Iswanto, "A Developed Optimization Model for Mass Production Scheduling Considering the Role of Waste Materials," *International Journal of Industrial Engineering and Management*, vol. 13, no. 2, pp. 135–144, 30 June 2022, DOI: 10.24867/IJIEM-2022-2-307.

[26] K. Mihály and Gy. Kulcsár, "A New Many-Objective Hybrid Method to Solve Scheduling Problems", *International Journal of Industrial Engineering and Management*, vol. 14, no. 4, pp. 326–335, Dec. 2023.

[27] R. V. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 7, no. 1, pp. 19–34, 2016, DOI: 10.5267/j.ijiec.2015.8.004.

[28] PSLIB Single Mode Scheduling Benchmark Dataset, http://www.om-db.wi.tum.de/psplib/getdata_sm.html, last accessed 2024/11/10

Krisztián Mihály is an assistant lecturer at the Institute of Information Science, University of Miskolc (Hungary). He received his M.Sc. in Information Engineering from the Budapest University of Technology and Economics (Hungary) in 2008. He is currently working on his PhD thesis. He is working as a development architect at SAP Hungary Ltd., responsible for SAP PLM for Process Industry solutions. He is recipient of the "Award of the Best Trainer of the Year at SAP". His research interests include production planning and scheduling, project planning and scheduling, many-objective optimization, metaheuristics, and software architectures.

Gyula Kulcsár is an associate professor in the Institute of Information Science at the University of Miskolc (Hungary). He received an M.Sc. in Information Science from the University of Miskolc (Hungary) in 2001. and a Ph.D. in Information Science from the University of Miskolc (Hungary) in 2008. His research interests include production planning and scheduling, project planning and scheduling, many-objective optimization, metaheuristics, discrete event-driven simulation, and manufacturing control. He is a member of the public body of the Hungarian Academy of Sciences. He is the recipient of the Young Researcher Scientific Award from the Hungarian Academy of Sciences in 2008.

Mónika Kulcsárné Forrai an associate professor in the Institute of Information Science at the University of Miskolc (Hungary). She received an M.Sc. in Information Engineering from the University of Miskolc (Hungary) in 2001, and a Ph.D. in Information Science and Technology from the University of Miskolc (Hungary) in 2018. Her research interests include scheduling, search algorithms, optimization, production planning and control, enterprise resource planning, and project scheduling. She is a member of the public body of the Hungarian Academy of Sciences.

Intelligent Intrusion Detection Systems – A comprehensive overview of applicable AI Methods with a Focus on IoT Security

Olivér Hornyák

***Abstract*—The rapid advancement of technology and the increasing complexity of cyber threats have necessitated the development of more sophisticated security measures. This paper presents a structured analysis of how artificial intelligence (AI) methods enhance the accuracy, adaptability, and efficiency of Intrusion Detection Systems (IDS). Different AI approaches, including machine learning, deep learning, and reinforcement learning are categorized and evaluated, highlighting their practical applications and limitations. The main focus is on enhancing the detection capabilities of IDS. By examining supervised, unsupervised, and reinforcement learning approaches, the study highlights how these methods can improve the accuracy, efficiency, and adaptability of IDS in identifying both known and novel threats. Additionally, the paper addresses the challenges associated with AI-based IDS, such as the need for extensive datasets, computational demands, and vulnerability to adversarial attacks. The findings underscore the transformative impact of AI on IDS and suggest directions for future research to further advance the field. With the exponential growth of Internet of Things (IoT) devices, securing networked environments has become increasingly challenging due to their resource constraints, diverse communication protocols, and exposure to cyber threats. Lightweight IDS models may provide solutions for the computational overhead, the scalability and privacy issues. This overview aims to serve as a valuable resource for researchers and practitioners seeking to leverage AI to bolster cybersecurity defenses. This paper not only provides a historical perspective but also critically analyzes current advancements and future research directions with a particular focus on IoT security and lightweight intrusion detection models.**

***Index Terms*—Information security. Intrusion detection, Artificial intelligence, Machine learning**

I. Introduction

A. Background

In cybersecurity, "intrusion" refers to any unauthorized access or attempt to gain access to a computer system, network, or data. This can include exploiting vulnerabilities to access sensitive information, installing malicious software, or disrupting normal operations. Intrusions can range from simple unauthorized logins to sophisticated, multi-stage attacks involving malware, phishing, or advanced persistent threats. The goal of Intrusion Detection Systems (IDS) is to identify and alert on such activities to help protect against data breaches and other security incidents.

The concept of detecting intrusions has been part of computer security efforts since the early days of computer networking. One of the first recorded intrusion detection systems, called "Cops" (Computer Oracle and Password System) was developed in the 1980s [1]. It was a collection of programs to warn the users of potential problems.

The IDS field matured after this period as computer security specialists increasingly understood the need to detect and respond to malicious activities in networks, ultimately leading to the more advanced and intelligent systems used today.

In the past decades, the rapid advancement of information technology has significantly transformed various industries and aspects of human life. Computer networks are necessary in business, industry, and everyday activities, necessitating the development of reliable and secure networks. However, this technological progress has also introduced numerous challenges, particularly in ensuring the availability, integrity, and confidentiality of network resources.

Among the various threats to network security, Denial of Service (DoS) attacks [63] stand out as particularly damaging. DoS attacks aim to disrupt the availability of services to end users by overwhelming network resources and systems with excessive, illegitimate requests. This type of attack first gained widespread attention in 2000 when Yahoo became one of the earliest high-profile victims. Today, web services and social media platforms are frequent targets of such attacks.

In addition to DoS attacks, other forms of cyber threats, such as Remote to Local (R2L) and User to Root (U2R) attacks, pose significant risks. R2L attacks involve an external attacker gaining local access rights to network resources that are typically restricted to local users, often exploiting vulnerabilities in services like file servers. U2R attacks, on the other hand, involve granting privileges from a normal user to a root user, providing full access to system resources to the attacker.

Cyber threats' dynamic and evolving nature makes it challenging for all attack types to use fixed, traditional security measures. Consequently, Intrusion Detection Systems have become an essential part of network security, developed to monitor network traffic and issue alerts upon detecting suspicious activities. IDS can be implemented as host-based systems, monitoring specific devices, or as network-based

This paragraph of the first footnote will contain the date on which you submitted your paper for review. This work was supported in part by University of Miskolc, grant No....

O. Hornyák is with the Institute of Information Science, University of Miskolc, Hungary, (e-mail: oliver.hornyak@uni-miskolc.hu).

DOI: 10.36244/ICJ.2025.5.8

systems, overseeing all network traffic. These systems are further categorized into anomaly-based and misuse-based IDS. Anomaly-based IDS detects attacks by comparing current traffic patterns against established baselines of normal behavior, offering the advantage of identifying novel attacks but often generating higher false positive rates. Misuse-based IDS rely on known attack signatures, effectively identifying known threats but potentially missing new, unknown attack vectors.

The structure of the rest of the paper is as follows: an overview of the historical development of IDS is presented, highlighting key milestones and technological shifts that have shaped modern security approaches. Following this, various AI methodologies applied in IDS, including supervised, unsupervised, and reinforcement learning techniques, are examined, with their strengths, limitations, and practical applications discussed.

A significant portion of the paper is dedicated to addressing the challenges and emerging trends in AI-driven IDS, particularly within the context of IoT security. The increasing demand for lightweight IDS models is analyzed, emphasizing their importance in resource-constrained environments. Critical factors such as computational efficiency, energy consumption, scalability, and real-time performance are evaluated. Recent advancements in edge computing, federated learning, and adversarial defense mechanisms are also explored, demonstrating their role in improving IDS effectiveness in modern cyber-physical systems. The paper concludes with a discussion on future research directions and open challenges in AI-driven IDS.

B. The goal of the paper

The primary goal of this paper is to provide a comprehensive and structured analysis of how artificial intelligence (AI) methods enhance the detection capabilities, adaptability, and efficiency of Intrusion Detection Systems (IDS), with a specific focus on IoT security. The paper aims to categorize and evaluate different AI approaches—including machine learning, deep learning, and reinforcement learning—highlighting their practical applications, strengths, and limitations in intrusion detection.

TABLE I
MILESTONES OF IDS

Year	Milestone
1980	First IDS concept by James Anderson
1987	Development of IDES at SRI International
1990	First commercial IDS: Haystack
1994	Deployment of Network-based IDS (NIDS)
1999	DARPA IDS Evaluation Dataset
2000	Launch of Snort, an open-source IDS
2003	Introduction of anomaly-based IDS
2005	Commercial use of machine learning in IDS
2010	Widespread adoption of SIEM systems
2015	Rise of AI-driven IDS solutions
2020	Integration of IDS with cloud security

A key objective is to analyze the challenges and emerging trends in AI-driven IDS, particularly in resource-constrained IoT environments. The paper discusses the necessity of lightweight IDS models to address issues related to computational overhead, scalability, and privacy concerns, ensuring that AI-based IDS solutions remain viable for IoT networks. Additionally, the study compares various AI techniques and examines their effectiveness in identifying both known and novel cyber threats.

Another objective is to establish a set of evaluation criteria for IoT-based IDS, ensuring that security solutions can be measured against essential performance and efficiency benchmarks.

II. HISTORY OF IDS

A. Main milestones

Intrusion Detection Systems have emerged as a fundamental component of cybersecurity, evolving significantly since their inception. The historical development of IDS provides a valuable overview of how these systems have adapted to the ever-changing landscape of cyber threats. From the early days of simple anomaly detection methods to the sophisticated, multi-layered defense mechanisms we see today, the history of IDS reflects the growing complexity and sophistication of both cyber-attacks and the technologies developed to counter them. Table I gives an overview of the significant milestones of IDS.

The concept of intrusion detection systems was first introduced by James Anderson in his seminal paper [3], which laid the foundation of understanding how monitoring system logs could help to detect unauthorized access to computer systems.

B. Early ages

This early concept focused on investigating audit trails to identify anomalies indicative of security breakings. The Intrusion Detection Expert System (IDES) was developed at SRI International's Computer Science Lab by Denning and Neumann in 1987 [4]. IDES was one of the first practical implementations of an IDS, designed to detect intrusions in real-time by analyzing system logs and network traffic for suspicious activity. It used statistical methods for reducing and analyzing audit trails [5]. After reengineering the prototype, the so-called Next-Generation Intrusion Detection Expert System (NIDES) was created to reach the production quality of the system. The observed behavior of the individual's system usage was compared to a profile-based value [6].

C. Commercial systems

„*Haystack*" was the first commercial IDS developed for the US Air Force in 1990. It marked the transition of IDS from research to practical application, providing real-time monitoring and alerting for potential security breaches within network environments [2].

D. Network-based IDS

1994 saw the Deployment of Network-based IDS (NIDS) which started to become widely recognized in the mid-1990s. These systems monitor network traffic in real-time, analyzing packet data to detect suspicious patterns that may indicate an attack [7]. The deployment of NIDS expanded the scope of IDS from individual host monitoring to entire network segments.

NIDS are typically deployed at strategic points within the network, such as at the gateway or in front of critical servers. They continuously capture and inspect packets traversing the network in real-time. NIDS uses predefined signatures or patterns of known threats to identify potential attacks. These signatures are similar to virus definitions used in antivirus software. When a packet matches a known signature, an alert is generated. In addition to signature-based methods, some NIDS employ anomaly-based detection. This approach involves creating a baseline of normal network behavior and flagging deviations from this norm as potential threats. Anomaly detection can help identify novel or previously unknown attacks.

There are challenges to NDIS. The biggest one is the high volume of data. NIDS needs to process large volumes of network traffic, which can be resource-intensive and may result in performance bottlenecks. Anomaly-based detection methods can generate false positives, where normal traffic is incorrectly flagged as malicious. The increasing use of encryption for network traffic can block NIDS's ability to inspect packets' content.

E. DARPA

The DARPA Intrusion Detection Evaluation Dataset, created by MIT Lincoln Laboratory in 1999, became a benchmark for testing and evaluating IDS performance. This dataset provided a standardized set of network traffic data containing both normal and malicious activities, enabling researchers to assess the effectiveness of various IDS approaches. The dataset records all network traffic, including the entire payload of each packet, in tcpdump format to enable comprehensive evaluation. It includes sniffed network traffic, Solaris BSM audit data, Windows NT audit data (for the DARPA 1999 dataset), and file system snapshots to identify intrusions against a test network composed of real and simulated machines. Background traffic was artificially generated, while attacks targeted real machines [8].

The dataset categorizes attacks into five main classes: Probe/Scan attacks, which scan networks to find valid IP addresses, active ports, OS types, and vulnerabilities; Denial of Service (DoS) attacks, which disrupt host or network services; Remote to Local (R2L) attacks, where an attacker gains local access without an account; User to Remote (U2R) attacks, where a local user obtains superuser or administrator privileges; and Data attacks, which involve exfiltration of sensitive files.

Despite its value, the DARPA dataset has faced criticism for its synthetic nature [12], which does not fully represent real-world traffic, and its limited representation of attack types, which may not reflect recent or diverse attack vectors. Additionally, performance evaluated with the DARPA 1999 dataset may not predict IDS effectiveness against modern threats or different network infrastructures.

Nevertheless, the DARPA dataset remains a significant tool for IDS research due to its detailed attack scenarios and comprehensive traffic records. It underscores the challenges of modeling network traffic and the need for continuous updates to reflect evolving threats and user behaviors. While more realistic datasets are needed for future research, the DARPA dataset's availability has been crucial for developing and evaluating IDS technologies.

F. SNORT

Snort, developed by Martin Roesch and released in 2000, revolutionized the IDS by providing a flexible, and open-source IDS tool. Snort's rule-based detection engine allowed users to write custom rules for identifying specific attack patterns, making it widely adopted in both academic and commercial environments. Snort is also recognized for its significant prevention capabilities, being the pioneer of the Intrusion Detection and *Prevention* System (IDPS) that supports both IDS and IPS modes, with significant prevention capabilities. As a Network Intrusion Detection and Prevention System (NIDPS). Snort is easy to configure and can effectively monitor network traffic. It compares received packets against known attack signatures and logs of detected attacks. In its IPS mode, Snort not only detects but also actively blocks malicious packets, preventing potential threats from causing harm to the network.

Snort uses Libpcap for packet capturing, followed by a decoder to interpret the captured packets. The preprocessor normalizes these packets, converting the traffic into a form that the detection engine can understand. The detection engine then applies predefined rules to identify and respond to malicious packets. In IPS mode, Snort's ability to block malicious packets in real time enhances its prevention capabilities, making it a crucial tool for network security.

Developed in 1998 and continually updated by an active community, Snort [9] remains relevant in modern network security. Despite lacking a Graphical User Interface, this limitation can be addressed with open-source visualization tools. With the introduction of the multi-threaded variant, Snort 3 further enhances its efficiency and capability in preventing network intrusions.

G. Anomaly-based IDS

Anomaly-based IDS approaches were introduced to detect unknown attacks by identifying deviations from established normal behavior patterns [10]. These systems use statistical models, machine learning, and other techniques to learn what constitutes normal activity and flag any deviations as potential threats. This approach is particularly effective against novel attacks that do not match any known signatures.

H. Machine Learning

Commercial use of Machine Learning (ML) in IDS appeared in 2005 [11]. By this time, machine learning techniques began to be integrated into commercial IDS products. These techniques improved the accuracy of intrusion detection by enabling systems to learn from historical data [13] and adapt to evolving threat landscapes. Machine learning-based IDS [12] [14] could better identify complex attack patterns and reduce false positive rates. The following chapter will give an overview of the most common methods

I. SIEM

Another milestone in 2010 was the widespread adoption of Security Information and Event Management (SIEM) systems: tools in modern cybersecurity, designed to provide comprehensive monitoring, detection, and response capabilities [15]. They work by aggregating, correlating, and analyzing security data from various sources within an IT infrastructure. The primary goal of SIEM systems is to detect potential security threats, ensure compliance with regulatory requirements, and offer a centralized view of an organization's security posture.

SIEM systems collect data from multiple sources such as firewalls, intrusion detection/prevention systems (IDS/IPS), antivirus software, and other security devices. This data is normalized to maintain a consistent format, facilitating easier analysis. Once collected, the data is correlated to identify patterns that might indicate security threats. This correlation process links events from different systems to provide context and identify potential incidents, often based on predefined rules and policies.

Real-time monitoring is a critical function of SIEM systems, which continuously monitor network traffic and system activities for signs of malicious behavior. When potential threats are detected, the system generates alerts based on predefined thresholds and anomaly detection mechanisms. These alerts prompt security personnel to investigate further.

Incident response is another vital component of SIEM systems. They can automate responses to certain types of incidents by executing predefined actions such as blocking IP addresses or isolating affected systems. For more complex incidents, security teams use SIEM systems to investigate alerts, analyze the context, and determine appropriate response actions. SIEM systems also provide robust log management capabilities. They store logs from various sources for extended periods, which is crucial for compliance audits, forensic investigations, and trend analysis. Advanced search functionalities allow security analysts to quickly query logs and retrieve relevant information.

SIEM systems enhance an organization's ability to detect, respond to, and manage security threats, playing a crucial role in maintaining a robust security posture.

J. AI-driven solutions

The rise of artificial intelligence (AI) and advanced analytics led to the development of AI-driven IDS solutions. These systems leverage deep learning, neural networks, and other AI technologies to detect sophisticated attacks with higher accuracy. AI-driven IDS can identify patterns and anomalies that traditional methods might miss, providing enhanced security insights. The following section will overview these methods.

K. Cloud security

With the increasing adoption of cloud computing, IDS solutions began to integrate with cloud security platforms around 2020. These integrated solutions provide comprehensive security monitoring across on-premises and cloud environments [16]. They address the unique challenges of cloud security, such as elastic scaling and dynamic infrastructure, ensuring continuous protection against cyber threats.

These milestones highlight the significant advancements in IDS technology over the years, reflecting the ongoing efforts to enhance network security and protect against evolving cyber threats.

III. Machine Learning and Artificial Intelligence

A. ML Methods

Machine learning and artificial intelligence approaches in intrusion detection systems comprise a wide range of techniques. ML methods rely on labeled data to train models that can classify network traffic as normal or malicious. Unsupervised learning techniques help identify patterns and anomalies in data without requiring labeled examples. Semi-supervised learning uses a small amount of labeled data to make use of a larger unlabeled dataset. Reinforcement learning is used to perform optimal actions for maximizing cumulative rewards in dynamic environments.

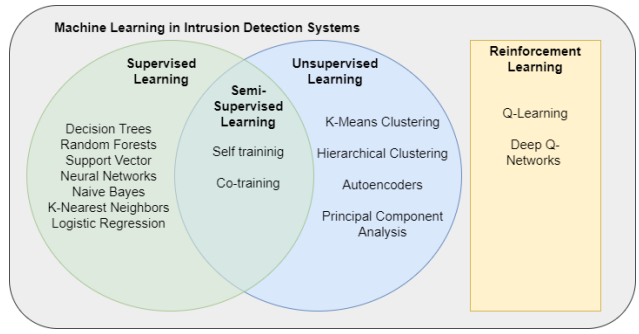


Fig. 1. Machine Learning methods in IDS

1) Decision trees

Th use of Decision trees technique IDS is popular due to their simplicity, interpretability, and efficiency in classifying network traffic. Decision trees operate by recursively partitioning the data into subsets based on the value of input features. This creates a tree-like model of decisions. Each node in the tree represents a feature, each branch represents a decision rule, and each leaf node represents an outcome or class label.

In the context of IDS, decision trees are used to analyze network traffic and identify patterns that distinguish normal activity from malicious activity. The process typically begins with the collection of network data, which is then pre-processed to handle missing values, noise, and irrelevant features. The decision tree algorithm is trained on this pre-processed data, learning the decision rules that best separate normal traffic from various types of attacks.

One of the key advantages of using decision trees in IDS is their ability to handle both categorical and numerical data, making them versatile for analyzing different types of network

features. Moreover, decision trees can be easily visualized, allowing security analysts to understand the decision-making process and interpret the results.

The application of a C4.5-based decision tree for detecting intrusions in imbalanced datasets is demonstrated by [22]. Their approach involves using a supervised relative random sampling technique to balance the data before training the decision tree, which helps improve detection accuracy for minority attack classes. This method achieved high accuracy on benchmark datasets like NSL-KDD and CICIDS2017, proving the effectiveness of decision trees in handling real-world IDS challenges. This approach underscores the adaptability of decision trees in different IDS scenarios, from simple binary classification of attacks to more complex multi-class problems. Additionally, [23] highlights the use of decision trees for both binary and multiclass classification in IDS.

Decision trees can provide a robust framework for intrusion detection by leveraging their inherent interpretability and ability to model complex decision boundaries. Their application in IDS continues to evolve, integrating with advanced techniques to improve detection rates and reduce false positives.

2) Random forest

In IDS applications, Random Forests are particularly effective in classifying network traffic and detecting anomalies, which helps in identifying potential security threats. It is valuable for its accuracy, and ability to handle large datasets with a high number of dimensions. Random Forest is a collective learning method that builds multiple decision trees during training and merges their results to improve classification accuracy and control overfitting. Each decision tree in the forest is trained on a random subset of the training data with replacement (bootstrap sampling), and a random subset of features is used for splitting at each node. The final classification is determined by majority voting among the trees, which enhances the model's generalization capability and stability.

In the field of IDS, Random Forests have shown significant improvements in detection accuracy and reduction in false positives. For example, research conducted on the NSL-KDD dataset demonstrated that Random Forest models achieve high accuracy and stability in detecting various types of network attacks [19]. The study revealed that the model could handle imbalanced datasets effectively, which is crucial for IDS where the number of normal traffic samples often far exceeds that of attack samples.

Enhanced versions of Random Forest, combined with techniques like the Synthetic Minority Over-sampling Technique (SMOTE), further improve the detection performance by addressing the class imbalance. These enhancements help in generating more balanced training datasets, leading to better classification of minority classes (i.e., attack types). Additionally, studies have optimized Random Forest models by integrating feature selection methods to identify the most relevant features for intrusion detection, which boosts the model's accuracy and reduces computational complexity. In [20] Boruta feature selection was used with RF, providing some worse accuracy but at reduced memory usage. Boruta aims to find all features that are relevant for prediction can be computationally expensive, especially with large datasets [21].

3) Support Vector Machines

Support Vector Machine (SVM) is a machine learning technique used in intrusion detection systems in classifying network traffic. SVMs are particularly effective for IDS because they can handle high-dimensional data and provide a clear decision boundary between normal and malicious activities. SVMs are employed to detect intrusions by learning the characteristics of both normal and attack traffic. The SVM algorithm works by mapping input features into a high-dimensional space and finding the hyperplane that best separates the different classes of data. This makes SVMs highly effective for binary classification tasks commonly found in intrusion detection, where the goal is to distinguish between benign and malicious traffic.

One significant application of SVMs in IDS is detailed in a comprehensive survey of their use on the KDDCUP'99 and NSL-KDD datasets. This research highlights how SVMs can achieve high detection accuracy and low false positive rates by efficiently learning the distinctions between normal and attack behaviors in network traffic. The study emphasizes that SVMs perform well in various IDS scenarios, from detecting known attack patterns to identifying novel threats by modeling normal behavior and spotting deviations [24].

Another notable application is the use of One-Class Support Vector Machines (OCSVM) in anomaly-based Intrusion Detection Systems. OCSVM is particularly useful when only normal data is available for training. It works by constructing a hyperplane that separates normal data from potential anomalies. This method is advantageous in environments where labeled attack data is scarce or hard to obtain. Research combining OCSVM with autoencoders for feature extraction has shown improved detection rates, demonstrating the effectiveness of SVM in semi-supervised learning scenarios for IDS [25].

4) Neural Networks

Traditional neural networks, specifically Multi-Layer Perceptrons (MLPs), are widely used in intrusion detection systems that rely on labeled datasets. These networks consist of multiple layers of neurons, including an input layer, one or more hidden layers, and an output layer. For each layer each neuron is connected to every neuron in the next layer, with each connection having a weight that is adjusted during training.

In IDS applications, MLPs are trained on labeled datasets where each instance of network traffic is marked as either normal or malicious. The network learns to map input features to the correct labels through a process of forward propagation, where inputs are passed through the network to produce an output, and backpropagation, where errors are propagated back through the network to adjust the weights.

For example, [26] utilized an MLP trained on the KDD CUP 99 dataset, achieving high accuracy in detecting various types of network intrusions. Another study [28] demonstrated that MLPs, when trained on labeled datasets like NSL-KDD, could effectively distinguish between different types of attacks,

providing a reliable method for intrusion detection. [27] compared MLP networks having different structures and achieved a detection accuracy of 95.6%.

These studies show that traditional neural networks, when properly trained on labeled datasets, can achieve significant performance in identifying and classifying network intrusions, thereby enhancing the security of network systems.

5) Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem, assuming independence among predictors. In intrusion detection systems using labeled datasets, Naive Bayes is employed to classify network traffic as normal or malicious by calculating the probability of each class given the input features. [29] investigated Naive Bayes' efficacy in IDS by training on the NSL-KDD 99 dataset. The classifier effectively identified various types of intrusions with high accuracy due to its simplicity and efficiency in handling large datasets. [30] highlighted the use of Naive Bayes in conjunction with feature selection methods, which improved the detection performance on the NSL-KDD dataset. The classifier's ability to handle categorical and numerical data made it suitable for various IDS applications, ensuring quick and accurate detection of network threats. [31] underscore Naive Bayes' role in IDS, leveraging its simplicity and effectiveness in processing labeled datasets to provide reliable network security.

6) K-Nearest Neighbors

K-Nearest Neighbors (kNN) is a simple, instance-based learning algorithm used in intrusion detection systems to classify network traffic using labeled datasets. The algorithm classifies a data point based on the majority class of its *k* nearest neighbors in the feature space.

The process starts with calculating the distance between the input data point and all points in the training set. Common distance metrics include Euclidean, Manhattan, and Minkowski distances. The *k* smallest distances are identified, and the class labels of these *k* nearest neighbors are retrieved. The input data point is then assigned the class that appears most frequently among these neighbors.

In IDS applications, kNN is effective for detecting various types of network intrusions. [32] investigated the algorithm's accuracy on the KDD CUP 99 dataset. The flexibility of KNN allows it to adapt to different types of network traffic and attacks, making it suitable for dynamic IDS environments. [33] highlighted KNN's ability to classify network activities efficiently using the NSL-KDD dataset, emphasizing the importance of selecting an appropriate *k* value and distance metric for optimal performance.

7) Logistic regression

Logistic regression is a statistical method used in IDS for binary classification tasks, distinguishing between normal and malicious network traffic using labeled datasets. The algorithm models the probability that a given input belongs to a particular class by applying a logistic function to a linear combination of input features.

In logistic regression, the algorithm learns the weights of input features during training to maximize the likelihood of correctly classifying the training data. The logistic function $L(x) = \frac{1}{1+e^{-x}}$, or – in neural network context often called as sigmoid function – maps the output to a probability between 0 and 1, which can be thresholded to decide the class label.

For IDS applications, logistic regression is employed to analyze network traffic features and predict the likelihood of an intrusion. [34] showed that logistic regression, when trained on the NSL-KDD dataset, effectively identified various types of network attacks with high accuracy. [35] demonstrated that logistic regression with multinominal regression model could enhance detection performance and reduce misclassification.

8) Semi-supervised learning techniques

Self-training and co-training are semi-supervised learning techniques used in intrusion detection systems to leverage both labeled and unlabeled data, enhancing detection performance without relying solely on extensive labeled datasets. Self-training involves an iterative process where an initial model is trained on a small, labeled dataset. This trained model then classifies the unlabeled data, and the high-confidence predictions are added to the labeled dataset as pseudo-labels. The model is retrained using this expanded dataset, iterating this process to gradually improve its accuracy. For example, [36] demonstrated that self-training improved IDS performance by generating more reliable pseudo-labels through uncertainty reduction techniques, such as using similarity graphs and graph convolutional networks to enhance the confidence and accuracy of predictions.

Co-training, on the other hand, uses two or more classifiers trained on different views or subsets of the features. Initially, each classifier is trained on a labeled dataset. Each classifier then labels the unlabeled data, and the most confident predictions from one classifier are added to the training set of the other classifier. This mutual reinforcement continues iteratively, improving the robustness of the model. Research has shown that co-training can significantly enhance IDS performance by combining classifiers' strengths and using ensemble methods to improve classification accuracy and reduce false positives [37].

These techniques are beneficial in IDS applications as they allow for the utilization of large amounts of unlabeled data, which is easier and cheaper to obtain than labeled data. By iteratively refining their models, self-training and co-training help IDS systems detect both known and unknown intrusions more effectively.

9) K-means clustering

K-means clustering is an unsupervised learning algorithm used for partitioning a dataset into distinct groups or clusters. In the context of intrusion detection systems, K-means clustering is applied to identify patterns in network traffic, distinguishing between normal and potentially malicious activities based on feature similarities. This method does not rely on labeled datasets; instead, it organizes the data into clusters where each data point belongs to the cluster with the nearest mean.

The K-means algorithm works as follows:

1. Initialize *k* cluster centroids randomly.
2. Assign each data point to the nearest centroid, forming k clusters.

3. Recalculate the centroids as the mean of all points in each cluster.
4. Repeat steps 2 and 3 until the centroids no longer change significantly or a maximum number of iterations is reached.

In IDS applications, K-means clustering helps in anomaly detection by grouping similar network behaviors. Points that fall into small or distant clusters may indicate anomalies or potential intrusions.

Comparing K-means clustering with K-Nearest Neighbors (KNN), we find that K-means is used for unsupervised learning, whereas KNN is a supervised learning algorithm. KNN classifies a data point based on the majority class of its *k* nearest neighbors, using labeled datasets to make predictions. K-means focuses on clustering similar data points without prior labels and is more suited for exploratory data analysis and anomaly detection.

[38] showed that while KNN excels in classification tasks with labeled datasets, K-means is more effective for initial pattern recognition and grouping of data in the absence of labels. Both methods can complement each other in IDS by using K-means to identify potential clusters of interest, which can then be further analyzed or classified using KNN with labeled data. These insights underscore the versatility of both algorithms in enhancing the detection capabilities of IDS through different approaches to data analysis.

10) Hierarchical clustering

Hierarchical clustering is a technique in cluster analysis aimed at constructing a hierarchy of clusters. It is especially valuable in intrusion detection systems for recognizing patterns and anomalies in network traffic. Unlike K-means clustering, hierarchical clustering does not require specifying the number of clusters as an input parameter of the algorithm. This method can be either agglomerative (bottom-up) or divisive (top-down). In agglomerative hierarchical clustering, each data point initially forms its own cluster. The algorithm then successively merges the nearest pairs of clusters until a single cluster is formed or a specified stopping criterion is reached. The steps are:

1. Compute the distance matrix for all data points.
2. Each data point forms a single cluster.
3. Merge the two closest clusters based on a chosen distance metric (e.g., single-linkage, complete-linkage).
4. Update the distance matrix to reflect the merger.
5. Repeat steps 3 and 4 until the desired number of clusters is achieved or all points are in one cluster.

In IDS applications, hierarchical clustering helps to discover the underlying structure of network traffic data, identifying groups of similar behaviors which can indicate potential intrusions. For example, [39] presented hierarchical clustering in identifying anomalies by analyzing the hierarchical structure of network connections and traffic patterns.

Hierarchical clustering, on the other hand, is unsupervised and does not use labeled data directly. It is more suited for exploratory data analysis and understanding the overall structure and relationships within the dataset, which can then be used to inform further analysis or classification tasks, potentially utilizing algorithms like KNN on labeled data.

11) Autoencoders

Autoencoders are a type of neural network used for unsupervised learning, particularly for feature learning and anomaly detection. In intrusion detection systems with labeled datasets, autoencoders are employed to detect anomalies by learning a compressed representation of the data and then reconstructing it [40]. The network consists of two main parts: the encoder, which compresses the input data into a latent space representation, and the decoder, which reconstructs the input data from this representation. The autoencoder is trained to minimize the reconstruction error, which is the difference between the input data and its reconstruction [41]. For IDS applications, the training is typically done on normal (benign) data. When the autoencoder encounters new data, it will reconstruct normal data well but will struggle with anomalous (malicious) data, resulting in higher reconstruction errors. These errors can then be used to flag potential intrusions.

12) Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction technique used in intrusion detection systems to transform high-dimensional data into a lower-dimensional space while preserving most of the variance in the data. PCA identifies the principal components, which are the directions of maximum variance in the dataset, and projects the data onto these new axes. This process reduces the number of features while retaining the most important information, making it easier to analyze and visualize the data. In IDS applications with labeled datasets, PCA is often used as a pre-processing step to enhance the performance of machine learning algorithms. By reducing the dimensionality, PCA helps in mitigating the curse of dimensionality, reducing computational costs, and improving the efficiency of the learning algorithms.

For example, [42] used PCA for feature reduction in a network Intrusion Detection System. The researchers of [43] applied PCA to the KDD CUP 99 dataset, reducing the number of features before applying classification algorithms like SVM and kNN. The results showed improved detection accuracy and reduced training time. NSL-KDD Dataset based on PCA-Fuzzy Clustering-KNN was used in [44].

13) Reinforcement learning methods

Reinforcement learning (RL) methods are used in Intrusion Detection Systems to enable models to learn optimal actions through trial and error by interacting with an environment. The primary goal is to maximize a cumulative reward signal. Unlike immediate rewards, which are given after each action, the cumulative reward is the total accumulated reward over a sequence of actions. The primary objective in RL is to develop a strategy (or policy) that maximizes this cumulative reward over time, not just the immediate reward.

$$G_t = \sum_{k=0}^{\infty} \gamma^k \cdot R_{t+k+1}$$

where

G_t is the cumulative reward starting from time step t

γ is the discount factor, which lies between 0 and 1 and

determines the importance of future rewards.
R_{t+k+1} is the reward received at time step t+k+1.

Two prominent RL methods in IDS are Q-learning and Deep Q-Networks (DQN). Q-learning is a model-free RL algorithm that aims to learn the optimal policy for an agent to take actions in a given environment. The core component of Q-learning is the Q-table, which stores the value of taking a particular action in a particular state. The algorithm updates the Q-values using the Bellman equation:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \cdot [r + \gamma \cdot max_{a'} Q(s',a') - Q(s,a)]$$

where
s is the current state,
a is the action taken,
r is the reward received,
s' is the next state,
a′ is the next action,
α is the learning rate, and
γ is the discount factor.

In IDS applications, Q-learning helps in adapting to new types of attacks by updating policies based on feedback from the network environment. [45] used the effectiveness of Q-learning in detecting and responding to network intrusions, showing that Q-learning could dynamically adjust its detection strategy based on observed network behaviors. Deep Q-Networks (DQN) extend Q-learning by integrating deep learning, allowing the algorithm to handle high-dimensional state spaces. Instead of maintaining a Q-table, DQN uses a neural network to approximate the Q-values. This enables DQN to scale to more complex environments that are impractical for Q-learning due to the large state-action space. The DQN algorithm involves training a neural network to predict Q-values and using experience replay to stabilize training.

In IDS applications, DQNs can detect complex patterns and adapt to evolving threats more effectively than traditional Q-learning. [46] applied DQNs to network intrusion detection, demonstrating the deep neural network's ability to learn intricate features from raw network traffic data. The study showed that DQN outperformed traditional Q-learning and other Machine Learning methods in identifying sophisticated attacks.

Comparing Q-learning and DQN, Q-learning is simpler and more straightforward, suitable for environments with smaller state-action spaces. DQN, on the other hand, is more powerful and scalable, capable of handling high-dimensional data and complex scenarios at the cost of increased computational resources and training time.

B. AI methods

Artificial intelligence approaches in IDS include expert systems and rule-based systems that use predefined rules to detect known threats. Signature-based detection, a form of expert systems, recognizes patterns that match known threat signatures. Anomaly detection uses statistical methods, heuristic methods, and behavioral analysis to identify deviations from normal behavior, which might indicate potential threats. Deep learning techniques, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory (LSTM) networks, are employed to analyze complex data patterns and sequential data like log files or time-series data. Hybrid systems combine machine learning with traditional rule-based methods, integrating multiple models to enhance detection accuracy and reduce false positives.

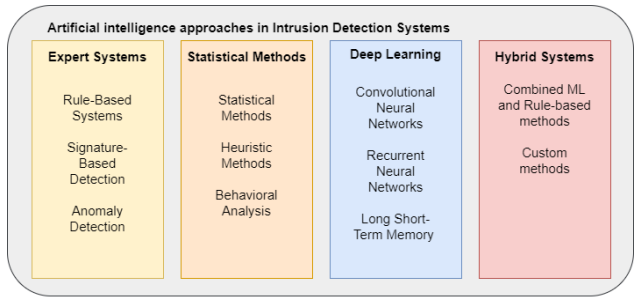


Fig. 2. AI approaches in IDS

1) Expert systems

Expert systems are artificial intelligence systems that use knowledge and inference procedures to solve problems that typically require human expertise. In the context of Intrusion Detection Systems, expert systems include rule-based systems, signature-based detection, and anomaly detection. Rule-based systems operate by applying predefined rules to the data being analyzed. These rules are created by domain experts and represent the knowledge about what constitutes normal and abnormal behavior in a network. The system compares incoming data against these rules to detect potential intrusions. For example, if a rule specifies that multiple failed login attempts within a short period indicate a brute-force attack, the system will flag such occurrences. Signature-based detection is a subtype of rule-based systems where the rules, or signatures, are patterns of known threats. These signatures are derived from previous attack patterns and behaviors. When network traffic matches a known signature, the system raises an alert. Signature-based IDS are highly effective at detecting known attacks but struggle with new or evolving threats. [47] demonstrated the efficacy of signature-based IDS in quickly identifying known malware based on established signatures.

Anomaly detection, on the other hand, builds a model of normal network behavior and identifies deviations from this model as potential intrusions. This method can detect previously unknown threats by recognizing unusual patterns of activity. Anomaly detection can use statistical methods, machine learning algorithms, or a combination of these to define what constitutes normal behavior [48].

Comparing these approaches, rule-based systems are easy to understand but can be limited by the completeness and accuracy of the rules. Signature-based detection is very effective for known threats but cannot detect new or modified attacks. Anomaly detection offers the advantage of identifying novel threats by focusing on deviations from normal behavior, but it

can generate false positives if the model of normal behavior is not accurately defined or if legitimate activities deviate from the norm.

2) Statistical methods

Statistical methods for Intrusion Detection Systems rely on mathematical models to analyze network data and detect anomalies indicative of potential intrusions [49]. These methods can be broadly categorized into heuristic methods and behavioral analysis. Heuristic methods use simple, rule-of-thumb strategies based on prior knowledge and experience to detect anomalies in network traffic. These methods often involve setting thresholds for various metrics, such as the number of failed login attempts or the volume of traffic from a single IP address. When these thresholds are exceeded, the system flags the activity as suspicious. Heuristic methods [51] are relatively easy to implement and understand but can generate false positives if the thresholds are not set accurately or if normal behavior varies widely. Behavioral analysis [50] involves creating profiles of normal network behavior and monitoring for deviations from these profiles. This approach uses statistical techniques to model the typical patterns of activity for users and systems. Any significant deviation from these patterns is considered an anomaly and potentially an intrusion. Behavioral analysis can adapt to changes in network behavior over time, reducing the likelihood of false positives. However, it requires a substantial amount of historical data to build accurate models and can be computationally intensive.

In IDS applications, heuristic methods provide quick and straightforward detection mechanisms. Behavioral analysis, on the other hand, offers a more dynamic and comprehensive approach. Comparing these methods, heuristic methods are simpler and faster to deploy but may lack the sophistication needed to handle complex or evolving threats. Behavioral analysis provides a deeper and more adaptive understanding of network activity but requires more data and computational resources to implement effectively.

3) Convolutional Neural Networks

Convolutional Neural Networks (CNNs) belong to deep learning models particularly effective for analyzing grid-like data structures, such as images and time-series data. In Intrusion Detection Systems, CNNs are employed to detect complex patterns in network traffic data, enhancing the ability to identify intrusions [52][53].

A CNN consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers apply a series of filters (kernels) to the input data to extract high-level features, such as edges or patterns in network traffic. The pooling layers down-sample the spatial dimensions of the data, reducing the computational load and emphasizing the most critical features. The fully connected layers, usually at the end of the network, perform classification based on the extracted features. The CNN algorithm operates as follows:

- Input Layer: Raw network traffic data is fed into the network. In IDS applications, this data can be represented as a matrix where rows represent different time steps and columns represent various traffic features.
- Convolutional Layer: Filters slide over the input data, performing convolution operations to detect local patterns. Each filter generates a feature map that highlights the presence of specific patterns.
- Activation Function: An activation function, typically Rectified Linear Unit (ReLU) [56], is applied to introduce non-linearity into the model.
- Pooling Layer: Feature maps are down-sampled using operations like max pooling or average pooling, which reduce dimensionality while retaining important information.
- Fully Connected Layer: The pooled feature maps are flattened into a vector and passed through one or more fully connected layers, which combine the features to classify the input data.
- Output Layer: The final layer produces a probability distribution over the class labels, determining whether the input data represents normal traffic or an intrusion.

In IDS applications, CNNs can learn to recognize intricate patterns in network traffic that may indicate malicious activity. For example, [54] applied CNNs to network traffic data from the NSL-KDD dataset. The CNN model demonstrated superior performance in detecting various types of network intrusions compared to traditional machine learning methods, thanks to its ability to automatically extract relevant features from raw data. [55] used CNNs to analyze time-series data from network traffic. The researchers found that CNNs could effectively capture temporal patterns and detect anomalies with high accuracy, reducing false positives and improving overall IDS performance.

CNNs provide a powerful tool for IDS by leveraging their deep learning capabilities to automatically learn and detect complex patterns in network traffic, making them highly effective for identifying sophisticated intrusions.

4) Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a type of deep learning model designed to handle sequential data, making them suitable for intrusion detection systems that analyze time-series network traffic data. RNNs are particularly effective in capturing temporal dependencies and patterns, which are crucial for detecting anomalies and intrusions in network behavior over time.

RNNs operate by maintaining a hidden state that captures information from previous time steps, allowing them to process sequences of data. The hidden state is updated at each time step based on the current input and the previous hidden state, enabling the network to retain information about past inputs. The RNN algorithm functions as follows:

- Input Layer: Sequential network traffic data is fed into the RNN. This data can include various features such as packet size, inter-arrival time, and protocol type.
  - Hidden Layer: At each time step t the RNN computes the hidden state h_t using the current input x_t and the previous hidden state h_{t-1}. The hidden state is updated using the formula:

$$h_t = \sigma \cdot (W_h \cdot h_{t-1} + W_x \cdot x_t + b)$$

where W_h and W_x are weight matrices,
b is the bias, and
σ is an activation function such as hyperbolic tangent (tanh) or ReLU.

- Output Layer: The output at each time step is computed based on the current hidden state. For classification tasks in IDS, a softmax layer is typically used to produce a probability distribution over the possible classes (e.g., normal traffic or different types of attacks).
- Training: The network is trained using backpropagation through time (BPTT), which involves unrolling the RNN for a number of time steps and applying the backpropagation algorithm to update the weights.

RNNs are well-suited for IDS because they can model the temporal dynamics of network traffic, capturing patterns that may indicate an ongoing intrusion. For instance, a study in the IEEE Access journal applied RNNs to the NSL-KDD dataset, demonstrating that RNNs could effectively detect intrusions by learning the temporal relationships in network traffic data.

[58] highlighted the use of Long Short-Term Memory (LSTM) networks, a variant of RNNs, to improve IDS performance. LSTMs address the vanishing gradient problem in standard RNNs by incorporating memory cells that can retain information over longer periods, enhancing the model's ability to detect slow, evolving attacks. RNNs [57] and their variants, such as LSTMs, provide powerful tools for IDS by leveraging their capability to learn from sequential data, capturing temporal patterns that are essential for accurate intrusion detection.

IV. IDS FOR IOT

The Internet of Things (IoT) represents a revolutionary shift in how devices interact and communicate with each other, promising unprecedented levels of convenience, efficiency, and automation. IoT encompasses a wide range of devices, from smart home appliances and wearable fitness trackers to industrial sensors and connected vehicles. However, this interconnected landscape also introduces significant security challenges, making Intrusion Detection Systems crucial for ensuring the integrity, confidentiality, and availability of IoT networks.

IoT devices often operate in a distributed environment, connected to the internet and various other networks, making them susceptible to a wide array of cyber threats. These threats include unauthorized access, data breaches, denial of service (DoS) attacks, malware infections, and more. The sheer number of IoT devices, many of which have limited processing power and memory, further complicates security measures. Traditional security solutions like firewalls and antivirus software are often inadequate for IoT due to their resource-intensive nature and the diverse range of devices and protocols in use.

Intrusion Detection Systems play an important role in safeguarding IoT environments by monitoring network traffic, detecting anomalies, and identifying potential security breaches. IDS can be broadly categorized into two types: Network-based IDS (NIDS) and Host-based IDS (HIDS). NIDS monitors network traffic for suspicious activity, while HIDS monitors the activities within individual devices. Both types are essential for comprehensive IoT security.

One of the primary functions of IDS is to provide real-time monitoring and alerting for potential security threats. This capability is particularly important in IoT environments where rapid detection and response can prevent minor incidents from escalating into significant security breaches. For instance, in smart home networks, an IDS can detect unusual traffic patterns that may indicate a compromised device attempting to communicate with external servers.

IoT networks can have a variety of behaviors depending on the devices and their usage patterns. IDS equipped with anomaly detection can learn the normal behavior of the network and flag deviations that may signify an intrusion. For example, an IDS can detect an increase in network traffic from a typically dormant device, suggesting that the device might have been hijacked for a botnet attack.

IoT environments face a broad spectrum of threats, ranging from simple brute-force attacks to sophisticated malware designed to exploit specific vulnerabilities. IDS can help mitigate these threats by identifying known attack signatures and using heuristic methods to detect new, previously unknown threats. This dual approach ensures that both common and novel attacks can be detected and mitigated.

IoT devices often have constrained resources, making it challenging to implement heavy security protocols directly on each device. IDS can offload the burden of security monitoring to more capable devices or cloud services, ensuring that security measures do not impede the performance of the IoT devices themselves.

As IoT devices become more prevalent in sectors like healthcare, finance, and critical infrastructure, compliance with regulatory standards becomes essential. IDS can help organizations meet these requirements by providing detailed logs and reports of network activity, facilitating audits, and ensuring that security measures are in place to protect sensitive data.

Implementing IDS in IoT also presents several challenges. The heterogeneous nature of IoT devices means that they often run on different platforms and operating systems, making it difficult to create a one-size-fits-all security solution. Additionally, the resource constraints of many IoT devices limit the types of security measures that can be implemented directly on the device. Network-based IDS solutions must be capable of handling high volumes of data from numerous devices without becoming a bottleneck.

Another challenge is the need for continuous updates. As new threats emerge, IDS must be constantly updated with the latest signatures and detection algorithms to remain effective. This requires a robust infrastructure for delivering updates and patches to IoT devices and IDS solutions.

Privacy concerns also come into play. IDS involves monitoring and analyzing network traffic, which can raise privacy issues, especially in environments where sensitive personal data is transmitted. Ensuring that IDS solutions

comply with privacy regulations and protect user data is paramount.

Despite these challenges, the benefits of implementing IDS in IoT environments are substantial. By providing real-time monitoring, detecting anomalies, mitigating diverse threats, and ensuring compliance with regulatory requirements, IDS plays a critical role in protecting IoT networks. As IoT continues to expand and evolve, the importance of robust, adaptive IDS solutions will only grow, making them an indispensable component of modern cybersecurity strategies.

Research and development in IDS for IoT are ongoing, with many promising approaches being explored [60]. Machine learning and artificial intelligence are increasingly being integrated into IDS to enhance their ability to detect and respond to new and sophisticated threats. [60] demonstrated that a Raspberry PI can fulfill the role of IDS in an IoT environment. A system designed for preventing botnet attacks is discussed in [62]. The state-of-the-art IDS methods in IoT networks are reviewed in [61].

In an IoT environment, a lightweight IDS system needs to be used with optimized algorithms that reduce computational overhead while maintaining effective threat detection capabilities. These models should extensively use techniques such as feature selection, dimensionality reduction (e.g., Principal Component Analysis), and low-complexity classifiers (e.g., decision trees, Naïve Bayes, and lightweight neural networks) to enhance efficiency. Additionally, hybrid approaches that combine signature-based detection with anomaly-based methods can improve detection accuracy while minimizing false positives. To further optimize performance, edge computing-based IDS solutions distribute detection tasks across IoT gateways reducing the need for constant cloud communication and enhancing real-time threat response. By adopting these lightweight strategies, IDS systems can effectively safeguard IoT networks against evolving cyber threats without imposing excessive computational demands.

A. Evaluation criterions

Intrusion Detection Systems (IDS) are evaluated based on several key factors that determine their effectiveness, efficiency, and applicability in different environments. While generic IDS and IoT IDS share some common evaluation criteria, IoT-specific IDS must address additional challenges due to the unique constraints and characteristics of IoT environments. The table below summarizes the key evaluation factors for both types of IDS.

TABLE II
IDS EVALUATION CRITERIONS

Evaluation Factor	Generic IDS	IoT IDS
Detection Accuracy	Measures the system's ability to correctly identify intrusions and minimize false positives/negatives.	Equally important but must also consider lightweight detection techniques to maintain efficiency.
False Positive Rate	Essential to minimize unnecessary alerts that make the security team overload	False positives can disrupt normal IoT operations and lead to unnecessary energy consumption.
False Negative Rate	IDS should avoid missing actual attacks, as this can compromise security.	More critical in IoT, as undetected attacks can disrupt real-time operations, e.g., healthcare or industrial IoT.
Real-time Performance	Response time is important but may tolerate slight delays in non-critical systems.	Extremely crucial in IoT environments where real-time threat detection is necessary (e.g., autonomous vehicles, industrial control systems).
Scalability	Must handle large networks but is generally deployed on powerful infrastructure	Must efficiently manage thousands of resource-constrained IoT devices across distributed environments.
Computational Overhead	Can be relatively high, especially for AI-based IDS, since enterprise systems have powerful computing resources	Must be **low**, as many IoT devices have limited CPU, memory, and energy constraints.
Network Overhead	IDS may introduce moderate network overhead for monitoring and logging.	Must **minimize communication overhead**, as IoT devices rely on low-bandwidth networks.
Adaptability to New Attacks	AI-based IDS and rule-based IDS must be regularly updated to detect evolving threats.	Requires lightweight, adaptive models that can learn new threats with minimal retraining, as frequent updates may not be feasible for IoT.
Energy Efficiency	Not a primary concern in traditional IDS.	**Highly important**, as IoT devices often run on battery power and cannot support continuous, power-intensive monitoring.
Privacy & Data Sensitivity	Monitors user/system activity but typically operates within a secure infrastructure.	Critical in IoT healthcare, smart homes, and industrial IoT, where personal/sensitive data must be protected from breaches.
Deployment Model	Typically centralized in enterprise networks with a dedicated security team.	Often decentralized, relying on edge computing or fog computing to distribute detection closer to the IoT devices.
Robustness Against Adversarial Attacks	IDS must handle sophisticated attack strategies like polymorphic malware.	IoT IDS is more vulnerable to adversarial ML attacks, sensor spoofing, and firmware-based exploitation.
Integration with Security Frameworks	Works alongside firewalls, SIEM systems, and endpoint security tools.	Needs **lightweight security integration**, often in resource-constrained environments where traditional firewalls may not be available.

Lightweight Security Integration refers to the incorporation of security mechanisms, into resource-constrained environments while maintaining minimal impact on computational resources, energy consumption, and network performance, which focuses on efficient, adaptive, and

decentralized security measures that ensure protection without degrading device functionality. It utilizes event-driven detection (e.g., anomaly detection triggered by specific behaviors) instead of continuous monitoring. It is important to use lightweight cryptographic protocols (e.g., elliptic curve cryptography) instead of computationally expensive security methods.

V. Sample Datasets for IDS Development

Intrusion Detection Systems datasets are essential for researchers as they provide a standardized and reliable foundation for developing, testing, and validating IDS algorithms. These datasets typically contain a mix of normal and malicious network traffic, enabling researchers to simulate real-world scenarios and measure the effectiveness of their detection methods. By using these datasets, researchers can benchmark their solutions against existing techniques, identify strengths and weaknesses, and iteratively improve their models. Furthermore, standardized datasets facilitate reproducibility in research, allowing different researchers to compare their results and advancements consistently.

The relevance of IDS datasets lies in their ability to represent a wide range of attack vectors and network behaviors, making them invaluable for developing robust and adaptive IDS solutions. Historical datasets like KDD Cup 1999 and DARPA [17] have laid the groundwork for intrusion detection research, while more recent datasets such as CIC-IDS2017 and TON_IoT reflect contemporary network environments and sophisticated attack strategies. These datasets not only help in understanding the evolution of cyber threats but also in developing next-generation IDS that can protect against emerging threats. By providing diverse and comprehensive data, IDS datasets empower researchers to push the boundaries of cybersecurity and enhance the resilience of networked systems.

IDS datasets are provided in various formats, each serving different purposes and offering unique advantages for data analysis and intrusion detection research, see Table II. Understanding these formats is crucial for effectively utilizing the datasets in IDS development and evaluation.

PCAP (Packet Capture) files contain raw network traffic data captured at the packet level. Each packet includes details such as source and destination IP addresses, port numbers, protocols, and payload data. PCAP is widely used in network analysis because it allows researchers to reconstruct network sessions and analyze the detailed behavior of network traffic.

NetFlow is a network protocol developed by Cisco for collecting IP traffic information. NetFlow records summarize flows of network traffic, providing aggregated information about the source and destination addresses, ports, protocols, and the volume of data transferred. This format is useful for analyzing traffic patterns and detecting anomalies over longer periods.

Logs are text-based files that record events generated by network devices, operating systems, and applications. Each log entry typically includes a timestamp, the source of the log, and a message describing the event. Logs are invaluable for forensic analysis and for identifying patterns of behavior that may indicate security incidents.

TABLE III
IDS DATASETS

Name	Description/ Location	Year	Labelled	Format
MAWI	Traffic traces from the WIDE project, used for traffic analysis and anomaly detection https://mawi.wide.ad.jp/mawi	2006 -	No	PCAP
B TON_IoT	IoT and network traffic data from a simulated smart city environment. https://research.unsw.edu.au/projects/toniot-datasets	2019	Yes	CSV, JSON
CIC-DDoS2019	Data from various types of DDoS attacks for detection and mitigation research. https://www.unb.ca/cic/datasets/ddos-2019.html	2019	Yes	CSV, PCAP
AAGM2013	Network traffic data for evaluating anomaly detection methods. https://csr.lanl.gov/data/audit/	2013	Yes	CSV
BoT-IoT	Synthetic IoT traffic with normal and attack scenarios for intrusion detection.	2018	Yes	CSV, PCAP
UGR'16	Real traffic captures from UGR, including benign and malicious traffic. https://nesg.ugr.es/nesg-ugr16	2016	Yes	CSV, NetFlow
CTU-13	Botnet traffic mixed with normal traffic from the Czech Technical University. https://www.stratosphereips.org/datasets-ctu13	2011	Yes	PCAP, CSV
Twente	Network traffic data with various network events for traffic analysis and anomaly detection. https://data.4tu.nl/articles/dataset/TNTS/Twente_University_Network_Traffic_Dataset/12781370	2015	Yes	NetFlow, CSV
SUEE	Dataset from Sharif University of Technology with a variety of attack scenarios. http://ocslab.hksecurity.net/Datasets/suee	2012	Yes	CSV
CACTI	Comprehensive archive of cyber threat intelligence, including network traffic and log files. https://github.com/CACTI-dataset	2020	Yes	JSON, CSV
DARPA98	Network traffic data with simulated normal and attack activities, foundational for IDS research. https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset	1998	Yes	PCAP, Logs
KDD Cup 1999	Derived from DARPA98, this dataset is used for network intrusion detection research. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html	1999	Yes	CSV

VI. Further Discussion

In conclusion, the integration of AI methods into Intrusion Detection Systems represents a significant advancement in the field of cybersecurity. AI techniques, including machine learning, deep learning, and neural networks, offer robust and adaptive solutions for detecting a wide range of cyber threats. These methods enhance the capability of IDS to identify both known and novel attacks with greater accuracy and efficiency compared to traditional detection approaches.

The application of machine learning algorithms has proven effective in analyzing large volumes of network traffic data, identifying patterns, and detecting anomalies that may indicate potential security breaches. Techniques such as supervised learning, unsupervised learning, and reinforcement learning have been successfully employed to improve detection rates and reduce false positives in IDS.

Deep learning, particularly through the use of convolutional neural networks (CNNs) and recurrent neural networks (RNNs), has shown exceptional promise in handling complex and high-dimensional data. These models are capable of learning intricate features from raw data, enabling more precise identification of sophisticated attack vectors. Additionally, the ability of deep learning models to continually learn and adapt to new threats makes them invaluable in the ever-evolving landscape of cybersecurity.

Despite the advancements, there are challenges associated with the deployment of AI-based IDS. The need for large, high-quality datasets for training, the computational resources required for model training and deployment, and the potential for adversarial attacks on AI models are critical issues that need to be addressed. Ongoing research and development are essential to overcome these challenges and to further enhance the performance and reliability of AI-driven IDS.

In summary, AI methods have revolutionized intrusion detection systems, providing more intelligent, adaptable, and efficient tools for safeguarding network security. Future research should focus on refining these techniques, addressing their limitations, and exploring new AI paradigms to keep pace with the advancing threat landscape. The continuous evolution of AI-driven IDS will play a crucial role in protecting digital infrastructures and ensuring the integrity and confidentiality of information in the digital age.

The computational demands of Intrusion Detection Systems are a significant consideration, particularly with the integration of advanced artificial intelligence (AI) techniques. AI-driven IDS, such as those utilizing machine learning and deep learning algorithms, require substantial processing power to analyze vast amounts of network traffic data in real-time. Training complex models, especially deep learning networks, involves intensive computations that necessitate the use of high-performance computing resources, including GPUs and distributed computing frameworks. Additionally, the deployment of these models in operational environments demands continuous monitoring and analysis, which can strain the computational resources of a network. The need for high-speed data processing, large-scale storage, and efficient memory management further adds to the computational burden. Consequently, the infrastructure supporting AI-based IDS must be robust and scalable to handle the high computational requirements, ensuring that the system can operate effectively without compromising performance or security. Addressing these computational challenges is crucial for the successful implementation and operation of intelligent IDS.

In the context of IoT-based Intrusion Detection Systems (IDS), selecting the most suitable machine learning and artificial intelligence methods requires balancing accuracy, computational efficiency, adaptability, and real-time processing. Supervised learning methods are widely used for detecting known attack patterns. Decision trees provide a simple and interpretable approach with minimal computational overhead, making them suitable for IoT devices with limited processing power. Random forests improve upon decision trees by aggregating multiple classifiers, offering higher detection accuracy while remaining efficient. Naïve Bayes classifiers, due to their probabilistic nature, are extremely fast and can work effectively in low-resource environments, making them a viable choice for IoT IDS with minimal training data. Support vector machines perform well in binary classification tasks and are particularly useful when labeled attack data is available, though they can be computationally demanding when dealing with large datasets.

For detecting unknown threats, unsupervised learning methods play a critical role. K-means clustering is effective for identifying anomalies by grouping network behaviors into clusters, though its performance depends on selecting an optimal number of clusters. Autoencoders offer a more advanced approach by learning normal network behavior and identifying deviations as potential attacks. These models reduce the feature space while preserving important data characteristics, making them efficient for IoT environments. One-class support vector machines provide another effective anomaly detection technique by modeling normal traffic and flagging deviations, which is particularly useful when labeled attack data is scarce.

Hybrid AI approaches enhance intrusion detection by combining multiple techniques to improve both detection accuracy and efficiency. Federated learning is increasingly relevant in IoT security as it enables decentralized model training across multiple devices without transferring sensitive data to a central server. This method enhances privacy while allowing IoT devices to collaboratively improve their IDS models. Reinforcement learning introduces an adaptive mechanism that enables IDS to continuously learn from its environment and adjust detection strategies dynamically. This approach is particularly valuable for evolving IoT security threats, as it does not rely on pre-labeled datasets.

To minimize latency and energy consumption, lightweight AI models optimized for edge computing environments are essential. Lightweight neural networks, designed with quantization and model compression techniques, allow IDS to detect threats directly on IoT gateways or edge devices without relying on cloud-based processing. TinyML [64], a specialized subset of machine learning designed for microcontrollers, further enables on-device intrusion detection with ultra-low

power consumption, making it a promising solution for battery-operated IoT systems.

IoT IDS must integrate models that not only achieve high detection accuracy but also maintain low computational complexity and real-time responsiveness. Decision trees and Naïve Bayes are ideal for signature-based detection, while autoencoders and one-class SVMs are effective for anomaly detection. Federated learning and reinforcement learning offer adaptive, privacy-preserving solutions, while lightweight neural networks and TinyML enable real-time, energy-efficient intrusion detection at the network edge. The combination of these techniques ensures that IoT IDS can provide robust security without overburdening constrained devices and networks.

Acknowledgment

The authors would like to express their sincere gratitude to the University of Miskolc, Institute of Information Science, for their generous funding and support. This research would not have been possible without their financial assistance and the valuable resources they provided. We are deeply appreciative of the opportunities and facilities made available to us, which have greatly contributed to the advancement of this study.

References

[1] D. Farmer. *Cops (computer oracle and password system)* (1989) [Online]. Available: http://coast.cs.purdue.edu/pub/tools/unix/scanners/cops/

[2] S. E. Smaha. (1988). Haystack: an intrusion detection system. *[Proceedings 1988] Fourth Aerospace Computer Security Applications*, 37–44. DOI: 10.1109/acsac.1988.113412

[3] Anderson, J. P. (1980). "Computer Security Threat Monitoring and Surveillance." *Technical report*, James P. Anderson Co.

[4] Denning, Dorothy, and Peter G. Neumann. Requirements and model for IDES-a real-time intrusion-detection expert system. Vol. 8. Menlo Park: SRI International, 1985.

[5] Denning, D. E. (1987). "An Intrusion-Detection Model." *IEEE Transactions on Software Engineering*.

[6] Amoroso, E. G. (1999). "Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response." Intrusion. Net Books.

[7] Northcutt, S., & Novak, J. (2002). "Network Intrusion Detection." New Riders Publishing. DOI: 10.1201/1079/43253.27.7.20000101/30304.4

[8] Lippmann, R. P., et al. (2000). "Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation." *Proceedings DARPA Information Survivability Conference and Exposition*. DOI: 10.1109/discex.2000.821506

[9] Roesch, M. (1999). "Snort: Lightweight Intrusion Detection for Networks." *Proceedings of the 13th USENIX Conference on System Administration*.

[10] Patcha, A., & Park, J. M. (2007). "An overview of anomaly detection techniques: Existing solutions and latest technological trends." *Computer Networks*. DOI: 10.1016/j.comnet.2007.02.001

[11] Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26, 159–190. DOI: 10.1007/s10462-007-9052-3

[12] Sommer, R., & Paxson, V. (2010). "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection." *IEEE Symposium on Security and Privacy*. DOI: 10.1109/sp.2010.25

[13] Chuvakin, A., Schmidt, C., & Phillips, D. (2013). "Logging and Log Management: The Authoritative Guide to Understanding the Concepts Surrounding Logging and Log Management." *Syngress*. DOI: 10.1016/b978-1-59-749635-3.00014-2

[14] Buczak, A. L., & Guven, E. (2016). "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection." *IEEE Communications Surveys & Tutorials*. DOI: 10.1109/comst.2015.2494502

[15] Muhammad, Adabi Raihan, Parman Sukarno, and Aulia Arif Wardana. "Integrated security information and event management (SIEM) with intrusion detection system (ids) for live analysis based on machine learning." *Procedia Computer Science* 217 (2023): 1406–1415. DOI: 10.1016/j.procs.2022.12.339

[16] Modi, C., Dhiren Patel, Bhavesh Borisaniya, Hiren Patel, Avi Patel, Muttukrishnan Rajarajan (2013). "A survey of intrusion detection techniques in Cloud." *Journal of Network and Computer Applications*. DOI: 10.1016/j.jnca.2012.05.003

[17] Thomas, C., Sharma, V., & Balakrishnan, N. (2008). Usefulness of DARPA dataset for intrusion detection system evaluation. SPIE Defense + Commercial Sensing. DOI: 10.1117/12.777341

[18] Negandhi, P., Trivedi, Y., & Mangrulkar, R. (2019). Intrusion detection system using random forest on the NSL-KDD dataset. In *Emerging Research in Computing, Information, Communication and Applications: ERCICA* 2018, Volume 2 (pp. 519–531). Springer Singapore. DOI: 10.1007/978-981-13-6001-5_43

[19] Wu, T., Fan, H., Zhu, H., You, C., Zhou, H., & Huang, X. "Intrusion detection system combined enhanced random forest with SMOTE algorithm." *EURASIP Journal on Advances in Signal Processing* 2022.1 (2022): 39. DOI: 10.21203/rs.3.rs-270201/v1

[20] Hasan, M. A. M., Nasser, M., Ahmad, S., & Molla, K. I. (2016). Feature selection for intrusion detection using random forest. *Journal of Information Security*, 7(3), 129–140. DOI: 10.4236/jis.2016.73009

[21] Subbiah, S., Anbananthen, K. S. M., Thangaraj, S., Kannan, S., & Chelliah, D. (2022). Intrusion detection technique in wireless sensor network using grid search random forest with Boruta feature selection algorithm. *Journal of Communications and Networks*, 24(2), 264–273. DOI: 10.23919/jcn.2022.000002

[22] Panigrahi, R., Borah, S., Bhoi, A. K., Ijaz, M. F., Pramanik, M., Kumar, Y., & Jhaveri, R. H. (2021). A consolidated decision tree-based intrusion detection system for binary and multiclass imbalanced datasets. *Mathematics*, 9(7), 751 DOI: 10.3390/math9070751

[23] Rai, Kajal, M. Syamala Devi, and Ajay Guleria. "Decision tree based algorithm for intrusion detection." International Journal of Advanced Networking and Applications 7.4 (2016): 2828.

[24] Ngueajio, M. K., Washington, G., Rawat, D. B., & Ngueabou, Y. (2022, September). Intrusion detection systems using support vector machines on the KDDCUP'99 and NSL-KDD datasets: A comprehensive survey. In *Proceedings of SAI Intelligent Systems Conference* (pp. 609–629). Cham: Springer International Publishing. DOI: 10.1007/978-3-031-16078-3_42

[25] Mhamdi, L., McLernon, D., El-Moussa, F., Zaidi, S. A. R., Ghogho, M., & Tang, T. (2020, October). A deep learning approach combining autoencoder with one-class SVM for DDoS attack detection in SDNs. In *2020 IEEE Eighth International Conference on Communications and Networking (ComNet)* (pp. 1–6). IEEE. DOI: 10.1109/comnet47917.2020.9306073

[26] Ji, H., Kim, D., Shin, D., & Shin, D. (2018). A study on comparison of KDD CUP 99 and NSL-KDD using artificial neural network. In *Advances in Computer Science and Ubiquitous Computing: CSA-CUTE* 17 (pp. 452–457). Springer Singapore. DOI: 10.1007/978-981-10-7605-3_74

[27] Mohammed, A. J., Arif, M. H., & Ali, A. A. (2020). A multilayer perceptron artificial neural network approach for improving the accuracy of intrusion detection systems. *IAES International Journal of Artificial Intelligence*, 9(4), 609. DOI: 10.11591/ijai.v9.i4.pp609-615

[28] Devi, R. R., & Abualkibash, M. (2019). Intrusion Detection System Classification Using Different Machine Learning Algorithms on KDD- 99 and NSL-KDD Datasets - A Review Paper. *International Journal of Computer Science and Information Technology*. DOI: 10.2139/ssrn.3428211

[29] Muda, Z., Yassin, W., Sulaiman, M. N., & Udzir, N. I. (2011, July). Intrusion detection based on K-Means clustering and Naïve Bayes classification. In *2011 7th international conference on information technology in Asia* (pp. 1–6). IEEE. DOI: 10.1109/cita.2011.5999520

[30] Deshmukh, Datta H., Tushar Ghorpade, and Puja Padiya. "Intrusion detection system by improved preprocessing methods and Naïve Bayes classifier using NSL-KDD 99 Dataset." *2014 International Conference on Electronics and Communication Systems (ICECS)*. IEEE, 2014. DOI: 10.1109/ecs.2014.6892542

[31] Panda, M., & Patra, M. R. (2007). Network intrusion detection using Naive Bayes. International journal of computer science and network security, 7(12), 258–263.

[32] Li, L., Zhang, H., Peng, H., & Yang, Y. (2018). Nearest neighbors based density peaks approach to intrusion detection. *Chaos*, Solitons & Fractals, 110, 33–40. DOI: 10.1016/j.chaos.2018.03.010

[33] Belgrana, F. Z., Benamrane, N., Hamaida, M. A., Chaabani, A. M., & Taleb-Ahmed, A. (2021, January). Network intrusion detection system using neural network and condensed nearest neighbors with selection of NSL-KDD influencing features. In *2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)* (pp. 23–29). IEEE. DOI: 10.1109/iotais50849.2021.9359689

[34] Besharati, E., Naderan, M., & Namjoo, E. (2019). LR-HIDS: logistic regression host-based intrusion detection system for cloud environments. *Journal of Ambient Intelligence and Humanized Computing*, 10, 3669–3692. DOI: 10.1007/s12652-018-1093-8

[35] Wang, Y. (2005). A multinomial logistic regression modeling approach for anomaly intrusion detection. *Computers & Security*, 24(8), 662–674. DOI: 10.1016/j.cose.2005.05.003

[36] Wang, Y., Huang, Y., Wang, Q., Zhao, C., Zhang, Z., & Chen, J. (2023). Graph-based self-training for semi-supervised deep similarity learning. *Sensors*, 23(8), 3944. DOI: 10.3390/s23083944

[37] Mao, C. H., Lee, H. M., Parikh, D., Chen, T., & Huang, S. Y. (2009, March). Semi-supervised co-training and active learning based approach for multi-view intrusion detection. In *Proceedings of the 2009 ACM symposium on Applied Computing* (pp. 2042–2048). DOI: 10.1145/1529282.1529735

[38] Aung, Y. Y., & Min, M. M. (2018, June). Hybrid intrusion detection system using K-means and K-nearest neighbors algorithms. In *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)* (pp. 34–38). IEEE. DOI: 10.1109/icis.2018.8466537

[39] Horng, S. J., Su, M. Y., Chen, Y. H., Kao, T. W., Chen, R. J., Lai, J. L., & Perkasa, C. D. (2011). A novel intrusion detection system based on hierarchical clustering and support vector machines. *Expert systems with Applications*, 38(1), 306–313. DOI: 10.1016/j.eswa.2010.06.066

[40] Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: an ensemble of autoencoders for online network intrusion detection. *Network and Distributed Systems Security (NDSS) Symposium* 2018, San Diego, CA, USA, DOI: 10.14722/ndss.2018.23204

[41] Choi, H., Kim, M., Lee, G., & Kim, W. (2019). Unsupervised learning approach for network intrusion detection system using autoencoders. *The Journal of Supercomputing*, 75, 5597–5621.

[42] Syarif, I., Prugel-Bennett, A., & Wills, G. (2012). Unsupervised clustering approach for network anomaly detection. In Networked Digital Technologies: 4th International Conference, NDT 2012, Dubai, UAE, April 24-26, 2012. *Proceedings*, Part I 4 (pp. 135–145). Springer Berlin Heidelberg. DOI: 10.1007/s11227-019-02805-w

[43] Rajadurai, H., & Gandhi, U. D. (2021). An empirical model in intrusion detection systems using principal component analysis and deep learning models. *Computational Intelligence*, 37(3), 1111–1124. DOI: 10.1111/coin.12342

[44] Benaddi, H., Ibrahimi, K., & Benslimane, A. (2018, October). Improving the intrusion detection system for NSL-KDD dataset based on PCA-fuzzy clustering-KNN. In *2018 6th International conference on wireless networks and mobile communications (WINCOM)* (pp. 1–6). IEEE. DOI: 10.1109/wincom.2018.8629718

[45] Alavizadeh, H., Alavizadeh, H., & Jang-Jaccard, J. (2022). Deep Q-learning based reinforcement learning approach for network intrusion detection. *Computers*, 11(3), 41. DOI: 10.3390/computers11030041

[46] Suwannalai, E., & Polprasert, C. (2020, November). Network intrusion detection systems using adversarial reinforcement learning with deep Q-network. In *2020 18th International Conference on ICT and Knowledge Engineering (ICT&KE)* (pp. 1–7). IEEE. DOI: 10.1109/ictke50349.2020.9289884

[47] Hubballi, N., & Suryanarayanan, V. (2014). False alarm minimization techniques in signature-based intrusion detection systems: A survey. *Computer Communications*, 49, 1–17. DOI: 10.1016/j.comcom.2014.04.012

[48] Jabez, J., & Muthukumar, B. J. P. C. S. (2015). Intrusion Detection System (IDS): Anomaly detection using outlier detection approach. *Procedia Computer Science*, 48, 338–346. DOI: 10.1016/j.procs.2015.04.191

[49] Kabir, E., Hu, J., Wang, H., & Zhuo, G. (2018). A novel statistical technique for intrusion detection systems. *Future Generation Computer Systems*, 79, 303–318. DOI: 10.1016/j.future.2017.01.029

[50] Moon, D., Im, H., Kim, I., & Park, J. H. (2017). DTB-IDS: an intrusion detection system based on decision tree using behavior analysis for preventing APT attacks. *The Journal of supercomputing*, 73, 2881–2895. DOI: 10.1007/s11227-015-1604-8

[51] Devarajan, Rajagopal, and Padmanabhan Rao." An efficient intrusion detection system by using behaviour profiling and statistical approach model." *Int. Arab J. Inf. Technol.* 18.1 (2021): 114–124. DOI: 10.34028/iajit/18/1/13

[52] Mohammadpour, L., Ling, T. C., Liew, C. S., & Aryanfar, A. (2022). A survey of CNN-based network intrusion detection. *Applied Sciences*, 12(16), 8162. DOI: 10.3390/app12168162

[53] Azizjon, M., Jumabek, A., & Kim, W. (2020, February). 1D CNN based network intrusion detection with normalization on imbalanced data. In *2020 international conference on artificial intelligence in information and communication (ICAIIC)* (pp. 218–224). IEEE. DOI: 10.1109/icaiic48513.2020.9064976

[54] Ding, Y., & Zhai, Y. (2018, December). Intrusion detection system for NSL-KDD dataset using convolutional neural networks. In *Proceedings of the 2018 2nd International conference on computer science and artificial intelligence* (pp. 81–85). DOI: 10.1145/3297156.3297230

[55] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Applying convolutional neural network for network intrusion detection. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1222–1228). IEEE. DOI: 10.1109/icacci.2017.8126009

[56] Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. *arXiv preprint arXiv:1702.01923*.

[57] Sohi, Soroush M. et al. "RNNIDS: Enhancing network intrusion detection systems through deep learning." *Comput. Secur.* 102 (2020): 102151. DOI: 10.1016/j.cose.2020.102151

[58] Hossain, M. D., Inoue, H., Ochiai, H., Fall, D., & Kadobayashi, Y. (2020). LSTM-based intrusion detection system for in-vehicle can bus communications. *IEEE Access*, 8, 185 489–185 502. DOI: 10.1109/access.2020.3029307

[59] Albulayhi, K., Smadi, A. A., Sheldon, F. T., & Abercrombie, R. K. (2021). IoT intrusion detection taxonomy, reference architecture, and analyses. *Sensors*, 21(19), 6432. DOI: 10.3390/s21196432

[60] Sforzin, A., Mármol, F. G., Conti, M., & Bohli, J. M. (2016, July). Rpids: Raspberry Pi IDS—a fruitful intrusion detection system for IoT. In 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (pp. 440–448). IEEE. DOI: 10.1109/uic-atc-scalcom-cbdcom-iop-smartworld.2016.0080

[61] Ankit Thakkar; Ritika Lohiya; "A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT: Recent Updates, Security Issues, and Challenges", *Archives Of Computational Methods In Engineering*, 2020. DOI: 10.1007/s11831-020-09496-0

[62] Idriss Idrissi; Mohammed Boukabous; Mostafa Azizi; Omar Moussaoui; Hakim El Fadili; "Toward A Deep Learning-based Intrusion Detection System for IoT Against Botnet Attacks", *IAES International Journal Of Artificial Intelligence*, 2021 DOI: 10.11591/ijai.v10.i1.pp110-120

[63] Péter Orosz, Balázs Nagy, Pál Varga, "Detection strategies for post-pandemic DDoS profiles", *Infocommunications Journal*, Vol. XV, No 4, December 2023, pp. 26–39., DOI: 10.36244/ICJ.2023.4.4

[64] Ji Lin, Ligeng Zhu, Wei-Ming Chen, Wei-Chen Wang, HanCai, Guangxuan Xiao, Haotian Tang, Shang Yang, Yujun Lin, and Song Han, "*Tiny Machine Learning Projects*" [Online]. Available: https://hanlab.mit.edu/projects/tinyml

Oliver Hornyak was born in Kazincbarcika, Hungary in 1973. He received the M.S. degree in Mechanical Engineering from the University of Miskolc, Hungary, in 1997, and the Ph.D. degree in Information Engineering from the University of Miskolc, Hungary, in 2002.
From 2002 to 2006, he was an Associate Professor at the University of Miskolc, Hungary. From 2006 to 2012, he was a Senior Lecturer at the University of Miskolc. Since 2012, he has been an Associate Professor at the Department of Information Technology of the University of Miskolc. His research interests include network security, intrusion detection systems, data encryption, and cybersecurity applications. He is the author of about 100 articles and holds a patent.

Guidelines for our Authors

Format of the manuscripts

Original manuscripts and final versions of papers should be submitted in IEEE format according to the formatting instructions available on

https://journals.ieeeauthorcenter.ieee.org/
Then click: "IEEE Author Tools for Journals"
- "Article Templates"
- "Templates for Transactions".

Length of the manuscripts

The length of papers in the aforementioned format should be 6-8 journal pages.
Wherever appropriate, include 1-2 figures or tables per journal page.

Paper structure

Papers should follow the standard structure, consisting of *Introduction* (the part of paper numbered by "1"), and *Conclusion* (the last numbered part) and several *Sections* in between.
The Introduction should introduce the topic, tell why the subject of the paper is important, summarize the state of the art with references to existing works and underline the main innovative results of the paper. The Introduction should conclude with outlining the structure of the paper.

Accompanying parts

Papers should be accompanied by an *Abstract* and a few *Index Terms (Keywords)*. For the final version of accepted papers, please send the short cvs and *photos* of the authors as well.

Authors

In the title of the paper, authors are listed in the order given in the submitted manuscript. Their full affiliations and e-mail addresses will be given in a footnote on the first page as shown in the template. No degrees or other titles of the authors are given. Memberships of IEEE, HTE and other professional societies will be indicated so please supply this information. When submitting the manuscript, one of the authors should be indicated as corresponding author providing his/her postal address, fax number and telephone number for eventual correspondence and communication with the Editorial Board.

References

References should be listed at the end of the paper in the IEEE format, see below:

a) Last name of author or authors and first name or initials, or name of organization
b) Title of article in quotation marks
c) Title of periodical in full and set in italics
d) Volume, number, and, if available, part
e) First and last pages of article
f) Date of issue
g) Document Object Identifier (DOI)

[11] Boggs, S.A. and Fujimoto, N., "Techniques and instrumentation for measurement of transients in gas-insulated switchgear," IEEE Transactions on Electrical Installation, vol. ET-19, no. 2, pp.87–92, April 1984. DOI: 10.1109/TEI.1984.298778

Format of a book reference:

[26] Peck, R.B., Hanson, W.E., and Thornburn, T.H., Foundation Engineering, 2nd ed. New York: McGraw-Hill, 1972, pp.230–292.

All references should be referred by the corresponding numbers in the text.

Figures

Figures should be black-and-white, clear, and drawn by the authors. Do not use figures or pictures downloaded from the Internet. Figures and pictures should be submitted also as separate files. Captions are obligatory. Within the text, references should be made by figure numbers, e.g. "see Fig. 2."
When using figures from other printed materials, exact references and note on copyright should be included. Obtaining the copyright is the responsibility of authors.

Contact address

Authors are requested to submit their papers electronically via the following portal address:

https://www.ojs.hte.hu/infocommunications_journal/about/submissions

If you have any question about the journal or the submission process, please do not hesitate to contact us via e-mail:

Editor-in-Chief: Pál Varga – pvarga@tmit.bme.hu

Associate Editor-in-Chief:

József Bíró – biro@tmit.bme.hu

László Bacsárdi – bacsardi@hit.bme.hu

21st International Conference on Network and Service Management

AI and Sustainability in the Future of Network and Service Management

27 - 31 October, 2025 - Bologna, Italy

CALL FOR WORKSHOP PAPERS

The 21st International Conference on Network and Service Management (CNSM) is inviting authors to submit original contributions to network and service management research. CNSM is a selective single-track conference that covers all aspects of network and service management, pervasive systems, enterprises, and cloud computing environments. In particular, CNSM 2025 will focus on AI and Sustainability in the Future of Network and Service Management.

Papers accepted and presented at CNSM 2025 will be published as open access on the conference website and will be submitted for possible publication in IEEE Xplore.

The 21st International Conference on Network and Service Management (CNSM 2025) will feature a rich workshop program, including the following side events to be held on Monday, 27 October or Friday, 31 October, 2025:

- 1st Workshop on Reinforcement Learning Applied to Networking (RLAN)
- Workshop on Quality of Dataset in Network Telemetry (QoDaNeT)
- 2nd Workshop on Network Security Operations (NeSecOr 2025)
- Workshop on Distributed Ecosystem for Circularity through Open Digital Enablers and the Digital Product Passport (DECODE)
- Workshop on Beyond centralized network and trust management: Ledgers, Orchestration in the cloud edge Continuum, and Knowledge-driven 6G networks (BLOCK-6G)
- 5th International Workshop on Analytics for Service and Application Management (AnServApp)
- Workshop on TRansport Architectures for Intelligent Networks (TRAIN)
- 2nd Workshop on Integrated Wireless Networking and Computing (IWNC)

Paper Submission

Authors are invited to submit original contributions that have not been published or submitted for publication elsewhere. Papers should be prepared using the IEEE 2-column conference style and are limited to 6 pages including references. Papers must be submitted electronically in PDF format through the EDAS system at https://edas.info/N33785.

Papers exceeding page limits, multiple submissions, and self-plagiarized papers will be rejected without further review. All other papers will get a thorough single-blind review process, followed by a rebuttal phase.

For further information, please check http://www.cnsm-conf.org/2025/.

Important Dates

Workshop Paper Submission:
11 August 2025
Acceptance Notification:
8 September 2025
Camera Ready due:
14 September 2025
Workshop dates:
27 October and 31 October

Workshop Co-Chairs

Tomáš Čejka,
CESNET, Czech Republic
Müge Sayıt,
University of Essex, UK

General Co-Chairs

Walter Cerroni,
University of Bologna, Italy
Mauro Tortonesi,
University of Ferrara, Italy

SCIENTIFIC ASSOCIATION FOR INFOCOMMUNICATIONS

Who we are

Founded in 1949, the Scientific Association for Infocommunications (formerly known as Scientific Society for Telecommunications) is a voluntary and autonomous professional society of engineers and economists, researchers and businessmen, managers and educational, regulatory and other professionals working in the fields of telecommunications, broadcasting, electronics, information and media technologies in Hungary.

Besides its 1000 individual members, the Scientific Association for Infocommunications (in Hungarian: Híradástechnikai és Informatikai Tudományos Egyesület, HTE) has more than 60 corporate members as well. Among them there are large companies and small-and-medium enterprises with industrial, trade, service-providing, research and development activities, as well as educational institutions and research centers.

HTE is a Sister Society of the Institute of Electrical and Electronics Engineers, Inc. (IEEE) and the IEEE Communications Society.

What we do

HTE has a broad range of activities that aim to promote the convergence of information and communication technologies and the deployment of synergic applications and services, to broaden the knowledge and skills of our members, to facilitate the exchange of ideas and experiences, as well as to integrate and harmonize the professional opinions and standpoints derived from various group interests and market dynamics.

To achieve these goals, we…

- contribute to the analysis of technical, economic, and social questions related to our field of competence, and forward the synthesized opinion of our experts to scientific, legislative, industrial and educational organizations and institutions;
- follow the national and international trends and results related to our field of competence, foster the professional and business relations between foreign and Hungarian companies and institutes;
- organize an extensive range of lectures, seminars, debates, conferences, exhibitions, company presentations, and club events in order to transfer and deploy scientific, technical and economic knowledge and skills;
- promote professional secondary and higher education and take active part in the development of professional education, teaching and training;
- establish and maintain relations with other domestic and foreign fellow associations, IEEE sister societies;
- award prizes for outstanding scientific, educational, managerial, commercial and/or societal activities and achievements in the fields of infocommunication.

Contact information

President: **FERENC VÁGUJHELYI** • *elnok@hte.hu*

Secretary-General: **GÁBOR KOLLÁTH** • *kollath.gabor@hte.hu*

Operations Director: **PÉTER NAGY** • *nagy.peter@hte.hu*

Address: H-1051 Budapest, Bajcsy-Zsilinszky str. 12, HUNGARY, Room: 502

Phone: +36 1 353 1027

E-mail: *info@hte.hu*, Web: *www.hte.hu*