International Journal on Advances in Systems and Measurements

2024 vol. 17 nr. 1&2

The *International Journal on Advances in Systems and Measurements* is published by IARIA.
ISSN: 1942-261x
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Systems and Measurements
Volume 17, Number 1 & 2, 2024

International Journal on Advances in Systems and Measurements
Volume 17, Numbers 1 & 2, 2024

CONTENTS

Devendra S. Gurjar, National Institute of Technology Silchar, India
Suneel Yadav, Indian Institute of Information Technology Allahabad, India

pages: 67 - 82
Identifying Semantic Similarity for UX Items from Established Questionnaires Using ChatGPT-4
Stefan Graser, RheinMain University of Applied Sciences, Germany
Martin Schrepp, SAP SE, Germany
Stephan Böhm, RheinMain University of Applied Sciences, Germany

Enhanced Simulation of Pipeline Fluid Transport with Phase Transition Detection and Pipe Subdivision Algorithm

Mehrnaz Anvari
Fraunhofer Institute for Algorithms and Scientific Computing
Sankt Augustin, Germany
email: Mehrnaz.Anvari@scai.fraunhofer.de

Anton Baldin
PLEdoc GmbH and Fraunhofer Institute for Algorithms and Scientific Computing
Sankt Augustin, Germany
email: Anton.Baldin@scai.fraunhofer.de

Tanja Clees
University of Applied Sciences Bonn-Rhein-Sieg and Fraunhofer Institute for Algorithms and Scientific Computing
Sankt Augustin, Germany
email: Tanja.Clees@scai.fraunhofer.de

Bernhard Klaassen
Fraunhofer Institute for Algorithms and Scientific Computing
Sankt Augustin, Germany
email: Bernhard.Klaassen@scai.fraunhofer.de

Igor Nikitin
Fraunhofer Institute for Algorithms and Scientific Computing
Sankt Augustin, Germany
email: Igor.Nikitin@scai.fraunhofer.de

Lialia Nikitina
Fraunhofer Institute for Algorithms and Scientific Computing
Sankt Augustin, Germany
email: Lialia.Nikitina@scai.fraunhofer.de

Sabine Pott
Fraunhofer Institute for Algorithms and Scientific Computing
Sankt Augustin, Germany
email: Sabine.Pott@scai.fraunhofer.de

***Abstract*—This work considers a stationary simulation of pipeline fluid transport, in the presence of impurities and phase transitions. This simulation finds applications in diverse areas such as energy carrier transportation, including natural gas and hydrogen, as well as the efficient transport of carbon dioxide from emission sources to designated storage sites. Particularly for the transport of carbon dioxide, which is preferably carried out in a liquid or supercritical state, the accurate detection of phase transitions is of utmost importance. Additionally, evaluating the simulation precision based on the selected pipe subdivision is crucial for transporting fluids of any kind. Our implementation includes an algorithm that utilizes the Homogeneous Equilibrium Model and the GERG-2008 thermodynamic equation of state for phase transition detection. We have also developed an optimal pipe subdivision algorithm using empirical formulas derived from extensive numerical experiments. Rigorous testing of the algorithms has been conducted on realistic fluid transport scenarios, confirming their effectiveness in addressing the stated technical challenges.**

Index Terms*—simulation and modeling; mathematical methods and numerical algorithms; advanced applications; fluid transport; carbon dioxide transport; pipe subdivision.

I. Introduction

This paper is an extension of our conference paper [1], which focused on the stationary simulation of carbon dioxide pipeline transport with impurities and phase transition detection. In this current study, we have expanded our simulations to include other fluids such as natural gas and hydrogen. Furthermore, we have developed an algorithm for pipe subdivision to enhance the precision of the simulation as desired.

To reduce greenhouse gas emissions into the atmosphere, Carbon dioxide Capture and Storage (CCS) systems are currently being developed. Typically, such systems consist of 3 parts: (1) capturing carbon dioxide (CO_2) at its source; (2) transporting CO_2 through pipelines to special storage sites; (3) and finally injecting it into wells, when underground storage is used. In this paper, we focus on the second part of the aforementioned process. It is generally required that CO_2 be in the liquid or supercritical phase during transport in order to increase the density and mass flows. It is essential to avoid the transition of fluid phase to gas, which leads to cavitation and destruction of the pipeline during transportation. To ensure reliable operation of the CO_2 pipeline, both an extensive experimental base and stable numerical simulation of the transportation process are required. At the same time, for a long-term planning, it is sufficient to simulate a stationary process of the transportation, with CO_2 in a 1-phase state and an indication of a possible phase transition, in order to prevent it.

The pioneering work [2] has considered in detail the stationary process of transporting pure CO_2 through a pipeline and pumping it into an underground storage, taking into account phase transitions. Subsequent papers, including [3]–[10], have highlighted the significance of considering impurities that can significantly impact transportation parameters even at low

concentrations. These papers have investigated both stationary and dynamic aspects of CO_2 transport. Papers [2]–[9] consider a Homogeneous Equilibrium Model (HEM), in which different phases of a fluid are homogeneously mixed and have the same speed, pressure, temperature and chemical potential. Papers [5]–[7], [9], [10] have also explored the concept of phase split, where the phases are geometrically separated, and phase slip, where the phases have different velocities. Additionally, works such as [5], [7], [9] have examined the formation of a solid phase of CO_2 (dry ice). Other studies [6], [7], [9], [10] have focused on fast transient processes that occur during pipe depressurization, including relevant experimental investigations. Furthermore, the economic aspects of pipeline CO_2 transport have been addressed in papers [11]–[14].

In this paper, we describe a stationary simulation of the CO_2 transport process with the possibility of considering impurities, phase transitions, several sources with different composition, and networks of complex topology. Simulations of this type have extended the capabilities of our software MYNTS [15]–[19]. The system provides an open, freely configurable and user-friendly specification of modeling, defined as a list of variables and equations. An open Python code for workflow procedures is also provided. The main calculations are performed in a fast C++ solver. The system also has a Graphical User Interface (GUI) with the ability to edit networks and scenarios. This architecture allows to formulate and quickly solve very large network problems, as well as the ability to model different energy carriers and couple different energy sectors.

For problems of stationary transportation of fluids, we implement standard pipe transport equations with friction terms by Nikuradse [20], Hofer [21] and spatial discretization of type [22]. The GERG equation of state [23], [24], which is currently the ISO standard [25], is used to accurately model the thermodynamics of fluids, in particular CO_2 with impurities and phase transitions. Additionally, we have developed an algorithm for detecting the proximity to the region of phase transitions. Numerical experiments were conducted to validate the implemented algorithms. These experiments demonstrate that the presence of phase transitions in the system can induce fast and occasionally abrupt behavior, which in turn influences the convergence properties of the numerical algorithms employed for the solution. In the scenarios we have considered, the divergence, if it occurs, is entirely localized in the region of phase transitions. On the other hand, scenarios without phase transitions are converging, which makes it possible to solve them with detection of proximity to the region of phase transitions.

Our implementation is based on standard numerical methods for solving systems of nonlinear equations, described in [26]–[28], applied to piecewise linear resistive systems in [29]–[31]. Questions of discretization of differential equations are considered in detail in general form in [26] and in application to gas networks in [32]–[34]. Our contribution to this area is the formulation of global convergence conditions for solution of nonlinear resistive systems [16] and their application to

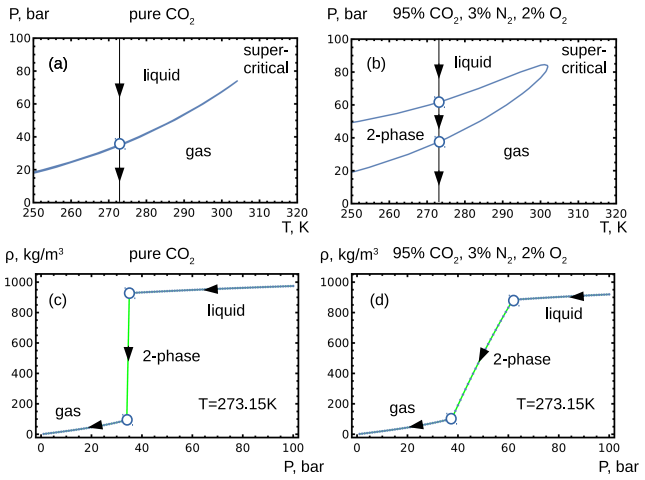


Fig. 1. Phase transitions at fixed temperature: (a),(c) – for pure CO_2; (b),(d) – for CO_2 with impurities. Image from [1].

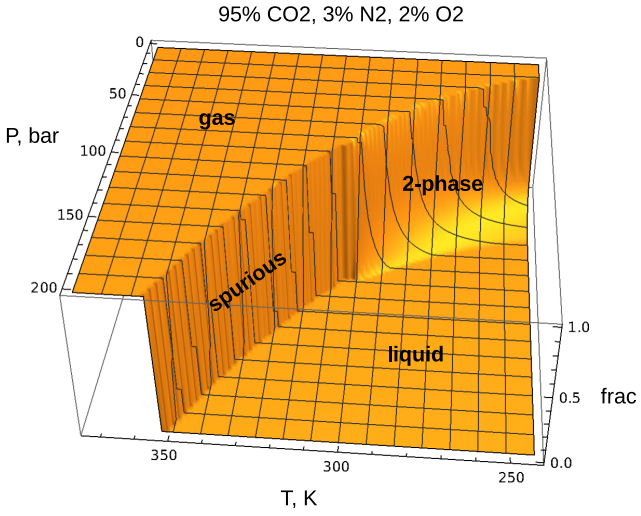


Fig. 2. Fraction of gaseous phase as a function of pressure and temperature. Image from [1].

stationary simulation of gas transport networks. We also constructed a pipe subdivision algorithm to achieve a given precision of stationary simulation of fluid transport networks and present it in this work.

Section II reviews the physics of phase transitions applied to CO_2 with impurities. Section III discusses the transport equations used. Section IV presents the pipe subdivision algorithm. In Section V, we describe numerical experiments, with particular attention paid to the questions of convergence of iterative processes and precision of simulation. Finally, in Section VI, we summarize our results.

II. Physics of Phase Transitions

Phase transitions exhibit slight variations in their occurrence between pure substances and their mixtures. Figure 1a shows the phase transition for pure CO_2. At a constant

temperature, the pressure decreases starting in the region of the liquid state. There is a line of phase transitions on the diagram. When the pressure decreases, the process proceeds until it intersects with this line, after that the pressure decrease stops until all the fluid passes from the liquid state to the gaseous state. At the same time, Figure 1c shows that during this process, the average density changes from large values, typical for the liquid phase, to small values, typical for a gas. In Figure 1b, the behavior of a mixture, i.e., 95% CO_2, 3% N_2, 2% O_2, is depicted. In this case, the two-phase state is not represented by a single line but rather a region on the (T, P)-diagram. The boundary of this region is called the Vapour-Liquid Equilibrium (VLE) diagram, or *phase envelope*. When the pressure decreases, the point enters this region and the fluid also passes from the liquid state to the gaseous state, but here the pressure continues to decrease. Figure 1d shows that in the 2-phase state, the density decreases in the same way as for pure substance, but at a decreasing pressure.

The 3D diagram in Figure 2 shows the behavior of $frac$-value, which varies in the interval $[0, 1]$ and measures the fraction of the gaseous phase in the fluid. In this figure, one can observe the region where the phase transition takes place, which occurs continuously for mixed compositions. Additionally, the diagram exhibits a discontinuity along a line originating from the critical point, although this transition is considered spurious. Above the critical point, there is no significant distinction between gas and liquid phases. However, based on the descriptive framework, a transition from gas to liquid is required at some point. While there is a formal jump in the quantity $frac$, physically measurable quantities do not exhibit such jumps along this line.

Interestingly, this surface resembles the surfaces considered in the theory of functions of a complex variable. Namely, if we take this surface, as well as the $1-frac$ surface and join them together, we get an object that looks like a Riemann surface for a complex square root. The similarity is not accidental, in both cases there is a 2-sheeted surface without the possibility of continuously separating the sheets from each other.

For the thermodynamic description of the fluid, the GERG equation of state and its accompanying implementation [23]–[25] are used. Technically, it is delivered as a software library where one can access a variety of functions describing the fluid state. In addition to the already mentioned phase envelope and $frac$-value, we use the Equation Of State (EOS) and energy functions

$$z = z(T, P, x),\ W = W(T, P, x), \tag{1}$$

where T is absolute temperature, P is pressure, x is a vector describing fluid composition, $W = (H, U, G, A)$ is a vector describing molar energies of different types: enthalpy, internal energy, Gibbs energy, Helmholtz energy, respectively. Compressibility factor z enters in the gas law $P = \rho RTz/\mu$, where R is the universal gas constant, ρ is the mass density, μ is the molar mass.

As an essential parameter for the user, the $frac$-value or a conservative algorithm utilizing $frac$-values in the vicinity of the solution can be employed to identify the proximity of phase transitions:

Algorithm (proximity-alarm):

```
given (T0,P0,x,dT,dP,val)
for T in (T0-dT,T0,T0+dT)
  for P in (P0-dP,P0,P0+dP)
    if frac(T,P,x)!=val return true
return false.
```

The algorithm considers a 3x3 grid created by $(\pm dP, \pm dT)$-variations, and if $frac$ differs from the user-specified val at least at one point, triggers a proximity alarm. This simple algorithm is applied to every node in the network. It has the advantage that it works even in the networks with many fluid compositions, i.e., variable x-values. Alternative algorithms based on the construction of the phase envelope produce many diagrams for different compositions, which complicates the analysis. At the same time, this algorithm has one drawback, it can produce a false alarm when approaching a spurious line. In this case, the user can visually control the solution trajectory on the (T, P)-diagram by constructing a phase envelope for the local network segment with constant x. Our future plans include the development of additional algorithms for automatic detection of phase transitions that can handle variable composition of the fluid within the network.

III. Pipe transport equations

In the stationary case, a pressure drop in the pipe is described by the equation:

$$dP/dL = -\lambda\rho v|v|/(2D) - d(\rho v^2)/dL - \rho g\, dh/dL, \tag{2}$$

where L is the running length along the pipe, v is the speed of the fluid, D is the internal diameter of the pipe, g is the gravitational acceleration, and h is the height. The first term on the right hand side is usually dominant, describing the contribution of the friction force, defined in terms of the dimensionless friction coefficient $\lambda(k/D, Re)$ using the Nikuradse [20] formula or the more accurate Hofer [21] formula. Here, k is the pipe roughness, $Re = 4|Q_m|/(\pi\mu_{visc}D)$ is the Reynolds number, where μ_{visc} is the dynamic viscosity and $Q_m = \rho v\pi D^2/4$ is the mass flow constant along the pipe. In addition, the right-hand side includes the convective and gravitational terms. The flow $Q_m = Q_N\rho_N$ is often expressed in terms of the normal volume flow Q_N and mass density ρ_N at normal conditions $P_N = 1.01325\ bar$, $T_N = 273.15\ K$.

For discretization purposes, we consider a short pipe segment of length L and integrate the equation over it. Expressing the velocity in terms of the mass flow, and keeping only the leading first term for illustration, we get $dP/dL = c_1/\rho$, where c_1 is constant. When performing integration, we substitute the variable density ρ with the average $\bar{\rho} = (\rho_1 + \rho_2)/2$ between the endpoints of the segment. In other words, $P_2 - P_1 = c_1 L/\bar{\rho}$. As an alternative, we multiply the original equation by P, use the gas law $P/\rho = RTz/\mu$, replace the variables T and z with the end averages and, thereby, we get

$(P_2^2 - P_1^2)/2 = c_1 L R \bar{T} \bar{z}/\mu$, in a more familiar quadratic form for gas dynamics [22].

Temperature profiles are described by the equation

$$dH/dL = -\pi D c_h (T - T_s)\mu/Q_m, \quad (3)$$

according to which the enthalpy change in a segment of the pipe is equal to the heat exchange with the soil or other environment. Here, c_h is the heat transfer coefficient, T_s is the soil temperature. Note that when the heat exchange is switched off ($c_h = 0$), the process described by this formula is isenthalpic $dH = 0$, and the temperature change is related to the pressure change by the well-known formula $dT = \mu_{JT} dP$, where $\mu_{JT} = -(\partial H/\partial P)_T/(\partial H/\partial T)_P$ is the Joule-Thomson coefficient. The equation can also be modified by introducing kinetic and gravitational terms.

For discretization purposes, the equation in the form $dH/dL = c_2(T - T_s)$ with a constant c_2, the variable temperature T is substituted with a constant T_x. The value of T_x can be chosen as the average temperature $\bar{T}$ at the endpoint or the outflow temperature T_{out}, depending on the scenario that better represents longer segments. After integration, we get $H_2 - H_1 = c_2 L(T_x - T_s)$. Further, in an iterative solution process in which the pressure profile and fluid composition are kept constant, the enthalpy values can be linearized using the formula $H(T^{i+1}) = H(T^i) + c_p(T^i)(T^{i+1} - T^i)$, where the superscripts indicate the number of iterations and $c_p = (\partial H/\partial T)_P$ is the isobaric molar heat capacity, also calculated by the GERG software library.

Next, we will consider in more detail the process of convergence of the iterations used for the solution. In our previous work [19], the architecture of MYNTS system has been described. Due to software-technical reasons, the solution was divided into 2 parts: (1) *Pressure-Massflow (PM)-iterations*, solved by a sparse non-linear Newtonian solver; and (2) *mix-iterations*, solved by a sparse linear solver. PM iterations determine the pressure, density and mass flow, by solving a relatively small nonlinear system. This system, however, has strong numerical instabilities associated with nearly zero Jacobi matrix eigenvalues and requires special stabilization measures [18]. Mix iterations solve a large linear system defining a multicomponent fluid composition, determine temperature and call external modules, such as GERG that would otherwise be called too often in a fully coupled system. After the temperature linearization described above, all mix equations of the system at each iteration become linear, their solution can be produced by a sparse linear solver such as Pardiso [35]. Further, these two processes are iterated, while using an additional stabilization algorithm called *weighted relaxation* [19], the result of the combined PM-mix-iteration $h(x)$ is replaced by a weighted average $x_{i+1} = wh(x_i) + (1 - w)x_i$.

Among the modeling limitations, it should be mentioned that the GERG module does not consider the solid phase and derives equilibrium conditions for the liquid and gaseous phases under the HEM assumptions. The transport equations considered here treat 2-phase solutions as 1-phase, with the values of thermodynamic parameters calculated by the GERG module in the *total* system, which also means calculations within the HEM framework.

TABLE I
PARAMETERS OF TEST SCENARIOS

parameter	symbol [units]	value
total pipe length	$L_{tot}[km]$	150
pipe internal diameter	$D[m]$	0.5
pipe roughness	$k[mm]$	0.5
heat transfer coefficient	$c_h[W/(m^2K)]$	4
inlet temperature	$T_1[K]$	313.15
soil temperature	$T_s[K]$	283.15
fluid composition	$x(CO_2, N_2, O_2)$	(0.95,0.03,0.02)
inlet pressure	pset [bar]	100
outlet norm.vol.flow, scen1	qset1 $[10^3 m^3/h]$	200
outlet norm.vol.flow, scen2	qset2 $[10^3 m^3/h]$	310
fluid composition	$x(CO_2)$	1
inlet pressure	pset [bar]	96.01325
outlet norm.vol.flow, scen3a	qset $[10^3 m^3/h]$	200
fluid composition	$x(CH_4)$	1
inlet pressure	pset [bar]	50.01325
outlet norm.vol.flow, scen3b	qset $[10^3 m^3/h]$	50
fluid composition	$x(H_2)$	1
inlet pressure	pset [bar]	50.01325
outlet norm.vol.flow, scen3c	qset $[10^3 m^3/h]$	50

At the conclusion of this section, it is important to address a general aspect concerning the simulation. It is common for users to assume the uniqueness of solutions obtained in simulations. However, it should be noted that in general, this assumption may not hold true. Existence and uniqueness theorems for solutions are only formulated in rare cases. So, for example, they are guaranteed for the PM subsystem under the conditions of generalized resistivity [15]. Being combined with the mix system, the uniqueness of the solution is not guaranteed. Theoretically imaginable is the situation when there are two stationary solutions, one 1-phase, the other 2-phase, and it may happen that the stationary solver finds the first one, but in reality the second one will be realized. Consideration of dynamic simulation can decide which solution the trajectory will go to when integrating from a given initial state. But even for a dynamic solver, saddle points, i.e., bifurcations of the solution are possible, where, with a small variation, the solution can go in one direction or the other. Questions about the uniqueness of stationary solutions and the stability of dynamic solutions must be investigated in the practical analysis of simulation results.

IV. PIPE SUBDIVISION ALGORITHM

First, we conduct a series of numerical experiments to analyze the relationship between precision and pipe subdivision. Next, we explore various discretization techniques and evaluate their precision and stability criteria. By carefully analyzing these criteria, we can determine which discretization methods are best suited for our needs. Finally, we implement these derived formulas into our subdivision algorithm. Based on the data gathered from our experiments and evaluations, we derive empirical formulas for pipe subdivision.

TABLE II
DERIVED PARAMETERS OF TEST SCENARIO 3

parameter	symbol [units]	value scen3a	value scen3b	value scen3c
fluid composition		pure CO_2	pure CH_4	pure H_2
inv. molar mass	$\mu^{-1}[mol/kg]$	22.722	62.332	496.06
mass flow	$m[kg/s]$	109.83	17.937	2.2471
heat capacity, molar isobaric, inlet-outlet	$c_p[J/(molK)]$	317.31-109.50	41.675-40.720	29.167-29.008
characteristic length, inlet-outlet	$x_1[km]$	126.03-43.491	7.4158-7.2457	5.1744-5.1462

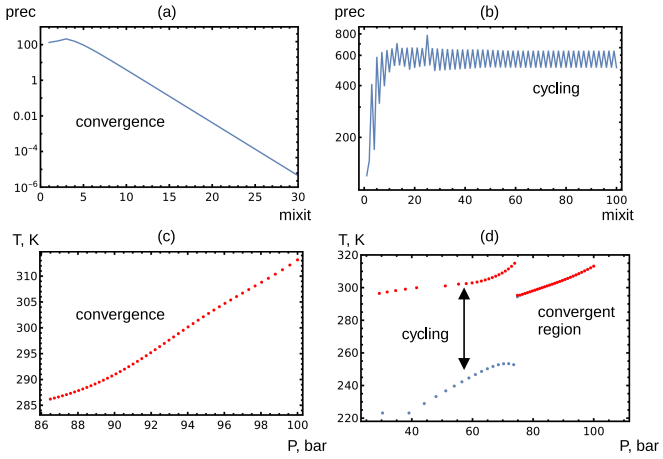


Fig. 3. (a),(c) – convergent iterations for scenario without phase transitions; (b),(d) – cycling iterations for scenario with phase transitions, red color - iteration 100, blue color – iteration 99. Image from [1].

Testing pipe subdivision: a set of simulations with variable pipe subdivision is considered,

$$L = L_{tot}/N_{div},\ N_{div} = 2^n,\ n = 1...n_{max}. \quad (4)$$

In our test case, $L_{tot} = 150km$, $n_{max} = 10$ are selected. In this section, we designate the symbol L to represent the length of a pipe segment. The coordinate along the pipe will be denoted as x, where x ranges from 0 to L_{tot}. Additionally, the symbol m will be used to represent the mass flow.

Precision is defined as maximal deviation of n-th solution from the most precise n_{max}-th solution:

$$\delta P_n = \max_x |P_n(x) - P_{n\_max}(x)|, \quad (5)$$

here for P, and similarly for other variables. The resulting dependencies of the precision on n are shown on Figure 5. The dependencies are mostly following $\sim L$ profile. Three scenarios are considered, with settings described in Table I. For $scen3a/CO_2$, subdivision $N_{div} \sim 16$, $n = 4$, corresponds to $\delta P \sim 0.1bar$. For $scen3b/CH_4$, similar δP is achieved at $N_{div} \sim 6$. For $scen3c/H_2$, such δP is achieved already at original pipe, $N_{div} \sim 1$, no subdivision needed.

Further details: the number of mix iterations is set to $nmixit = 30$, their convergence is controlled. Phase transitions do not happen for all scenarios. A slight bent at $n = 9$ is a methodical issue: since n_{max} subdivision is not exact answer, $n_{max} - 1$ level feels the error of n_{max} level, while the other n-levels are less sensitive to this error.

Various discretizations: the schemes, briefly described in the previous section, will be considered in more details now. In Hofer friction law (2), we track the leading $\sim L/D$ term $dP = -f_R dx$, where $f_R(x) \sim 1/\rho(x)$ with coefficient constant along the pipe segment.

Hofer-quad: multiply both sides by ρ, using that $\rho \sim P$ approximately for gases (gas law $P = \rho RTz/\mu$, where the coefficient RTz/μ is assumed to change slowly along the pipe segment and is represented by its nodal average), integration gives: $lhs = \int PdP = (P_2^2 - P_1^2)/2$ and $rhs \sim \int dx = L$, a known formula for quadratic hydraulic resistance applicable for gases.

Hofer-lin: in f_R take the nodal average $\rho \to (\rho_1 + \rho_2)/2 = const$, then integrate straightforwardly: $lhs = P_2 - P_1$, $rhs \sim \int dx/\rho = L/((\rho_1 + \rho_2)/2)$. A surprising equivalence: the leading terms for Hofer-quad and Hofer-lin coincide.

Proof: in Hofer-quad integrated formula, divide $lhs = (P_2^2 - P_1^2)/2$ to $(P_1 + P_2)/2$, use approximate linearity $\rho \sim P$ above, obtain $rhs \sim L/((\rho_1 + \rho_2)/2)$ coincident with Hofer-lin integrated formula. □

This is what we see on Figure 5, the coincidence of precision for both schemes, perfect for gases (methane at 50bar), slightly deviating for liquids (supercritical CO_2). The deviation is due to the omitted terms in the friction law, which are indeed different for two integration schemes, and due to details of taking nodal average: $Tz \to (T_1 z_1 + T_2 z_2)/2$ vs $(T_1 + T_2)/2 \cdot (z_1 + z_2)/2$, etc. At $n = n_{max}$ level, Hofer-quad and Hofer-lin results coincide at high precision ($\sim 10^{-6}$bar), so that both schemes provide a consistent discretization for the same continuous equation.

Stability considerations: according to [16], for convergence of the simulation, the signature of the whole equation must be $\partial eq/\partial(P_1, P_2, m) \sim (+--)$, while $\partial\rho/\partial P > 0$. In Hofer-quad, this criterion requires to replace $lhs = (P_2|P_2| - P_1|P_1|)/2$, unfolding the expression to $P < 0$ unphysical domain with correct P-signature.

In Hofer-lin, other discretization schemes can be considered, that theoretically can be more stable. Recovering m-dependence: $rhs \sim Lm|m|/\rho$, if ρ is replaced to the nodal average $(\rho_1 + \rho_2)/2$, it will have a wrong signature w.r.t. P_1 or P_2, dependently on the sign of m. A possible alternative is $rhs \sim Lm|m|/(m > 0?\rho_1 : \rho_2)$, C^1-continuous in $m = 0$. Although less precise than the nodal average, it possesses

correct P-signatures and should lead to a more stable solution process. This approach can be extended to other terms in the friction law in its different formulations.

Practically, Hofer-lin scheme is not yet usable in complex scenarios like N85 tests described below. In large natural gas networks, it shows worse stability ($div/tot = 7/85$) in comparison with Hofer-quad ($div/tot = 1/85$). Here div is the number of divergent cases, while tot is the total number of cases. The source of the instability can be in the violation of signature rule, in the leading as well as in sub-leading terms.

Empirical formulas: since numerical experiments show $\sim L$ behavior of precision at large n (small L), here it's only needed to find empirically plausible factors. Employing that the pressure drop is proportional to $dP \sim m^2/\rho \cdot L$, omitting constant factors, evaluating relative error of pressure drop due to ρ-variation, we have $\delta dP/dP = d\rho/\rho$, where δ denotes error estimation, d is a change over the pipe. Absolute values for all changes are taken. This formula assumes intermediate ρ values in pipe changing arbitrarily between their nodal values. An empirical factor 0.5 can be introduced, if the change is taken between the nodal values and the nodal average. The resulting empirical formula for relative error of pressure drop is

$$errP = \delta dP/dP = 0.5 d\rho/\rho. \quad (6)$$

For T-dependence, a simplified exponential model can be used: $T = T_s + (T_1 - T_s)\exp(-x/x_1)$, with the characteristic length $x_1 = c_p m/(\mu D c_h \pi)$. Here the main factor is proportional to L/x_1, since subdivisions with $L \sim x_1$ give roughly acceptable quality, $L \ll x_1$ fine quality, $L \gg x_1$ inacceptable. What "roughly acceptable" means can be found in comparison of discrete and continuous integration of the simplified model: $\delta T = |T_1 - T_s||(1 + L/x_1)^{(-x_1/L)} - 1/e|$, we obtain $errT = 0.132121$ at $L = x_1$, conservatively giving a factor 0.2. Thus, the following expression can serve as an estimator of relative T-precision:

$$errT = \delta dT/dT = 0.2 L/x_1. \quad (7)$$

The values of the characteristic length for 150km pipe scenario with different fluid composition are shown in Table II. Figure 5 confirms that the derived formulas provide tight conservative estimators for the pipe subdivision error.

Implementation of pipe subdivision: taking the desired level of relative error, we construct an estimator for subdivision number, with maximum taken over P- and T-estimators. This estimator should be evaluated for every pipe. If further accepted by the user, it provides the following subdivision algorithm with a minimal user assistance.

Algorithm (pipe-subdivision):

```
given (x1min,mmin,nmax,err_desired)
for every pipe
  compute x1,errT,errP
  if x1<x1min or |m|<mmin
    errT=0
  err=max(errT,errP)
```

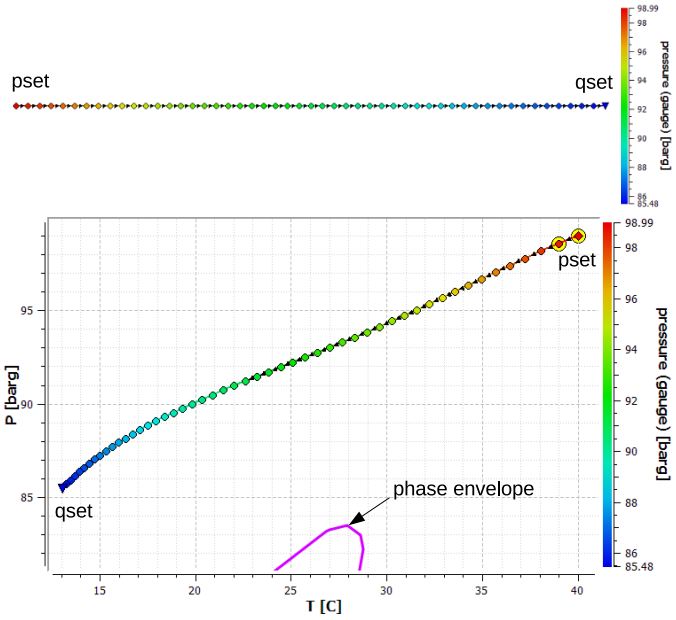


Fig. 4. Screenshot of MYNTS GUI for scenario without phase transitions. Image from [1].

```
n_suggested=[err/err_desired]+1
if n_suggested>nmax
  n_suggested=nmax.
```

Details: for better efficiency, some cutoffs are necessary. At small x_1 values, in particular, at small m, the value T rapidly jumps to T_s. Under these conditions, uniform subdivision algorithm would provide too large $n\_suggested$. To prevent this, for $x_1 < x_{1,min}$ or $|m| < m_{min}$, $errT = 0$ is set, subdivision is defined by $errP$ only. In addition, if $n\_suggested > n_{max}$, it is set to n_{max}.

The discretizations used here possess $\sim L$ precision dependence. A study of the other schemes with a higher order $\sim L^n$ dependence is in our further plans. The tradeoff between precision and stability should be also considered. For higher order schemes, the empirical formulas should be upgraded, while the subdivision algorithm remains the same.

The approach requires at least one preliminary simulation to find all necessary parameters. The pipe subdivision is sensitive to such details as mass flow, temperature, density, gas composition, and is performed for a given scenario. If scenario is changed, subdivision should be repeated starting from the raw level.

The usage of the algorithm proceeds via specification of input parameters, listed in Table III. The resulting output values are stored per pipe and can be visualized. The application of the algorithm to realistic networks is presented in the next section.

V. Numerical Experiments

To test the implemented algorithms, we apply them to a number of realistic network problems. At first, we test phase transition detection, then pipe subdivision algorithm.

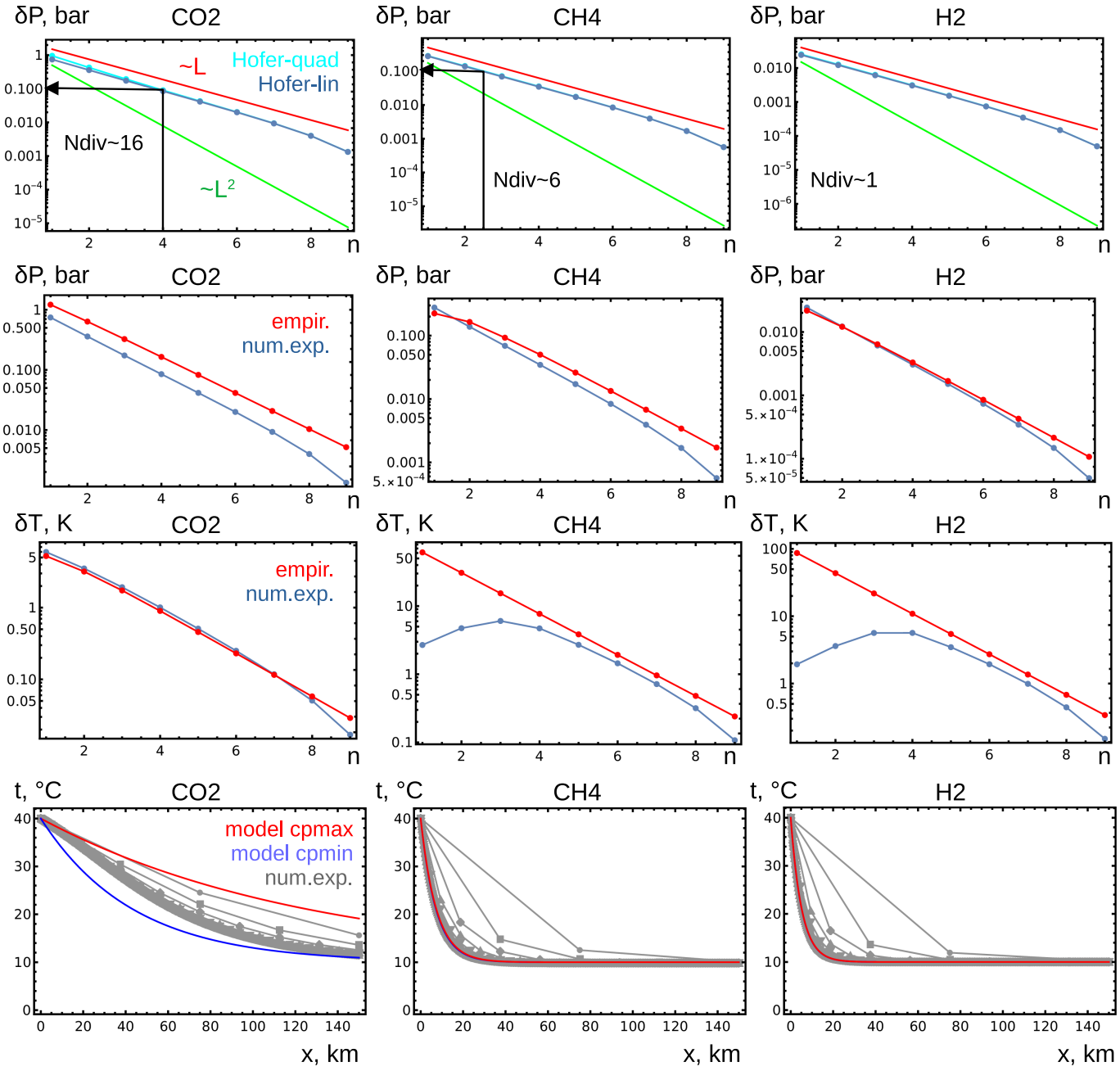


Fig. 5. Dependence of the simulation precision of pressure δP and absolute temperature δT on subdivision level n; dependence of relative temperature t on coordinate along the pipe x.

Testing phase transition detection (scen1-2): here we use a pipe segment with parameters taken from [2]. In our experiments, two scenarios are considered, see Table I. In the first scenario, a small flow is set, at which phase transitions do not occur. The entire pipe is filled with liquid or supercritical fluid. In the second scenario, a larger flow is set, the pressure drops more strongly, and a phase transition occurs in the system. Both scenarios use a mixture of 95% CO_2, 3% N_2, 2% O_2, see Figure 9 in [2]. The pipe is laid horizontally with $h = 0$.

Figure 3 shows the convergence characteristics for our test scenarios, left column for scen1, right column for scen2. The dimensionless precision parameter $prec = \max(res_i/norm_i)$ is defined as the maximum of the residuals of the equations divided by the normalizing value, for each equation its own. For the Kirchhoff equation of conservation of flow, the friction law in quadratic form, and the gas law expressed with respect to density, the normalization factors $norm = (1kg/s, 100bar^2, 1kg/m^3)$ are chosen, respectively. In our system, the equations and their normalizing factors can be

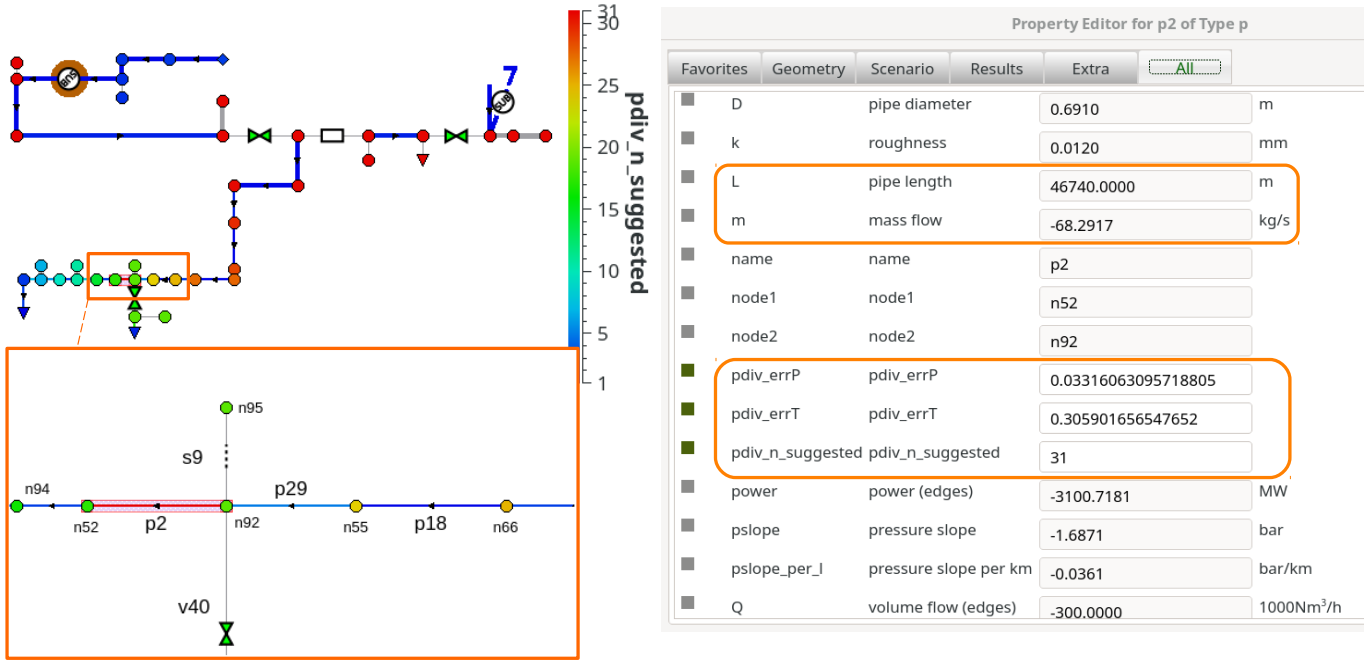


Fig. 6. Top-left: distribution of pipe subdivision estimator $n\_suggested$ over the test network N1. Bottom-left: closeup to the pipe with the largest subdivision. Right: parameters of the selected pipe.

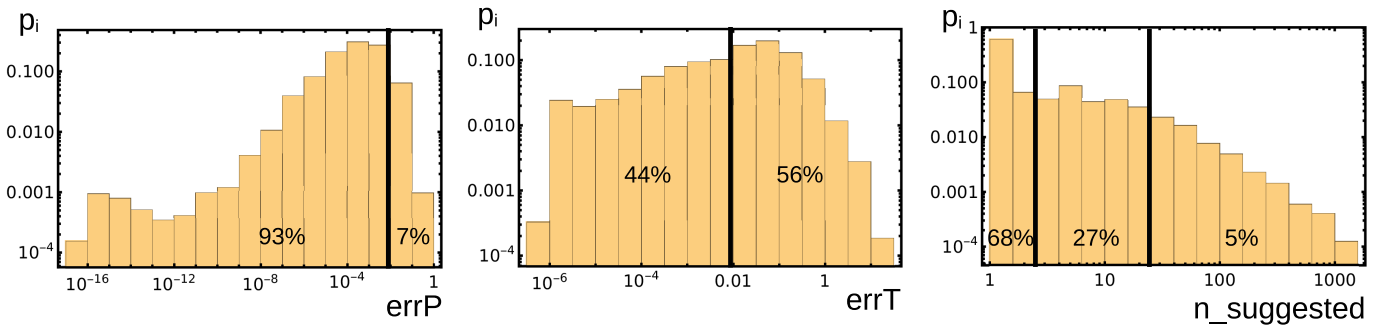


Fig. 7. Distribution of pipe subdivision estimators for the test networks N85.

TABLE III
PARAMETERS OF PIPE SUBDIVISION ALGORITHM

parameter	meaning
activation: pdiv	0/1 (default 0, inactive)
input parameters:	
pdiv_x1min	x1-cutoff (default 1 [m])
pdiv_mmin	m-cutoff (default 1 [kg/s])
pdiv_nmax	n-clamp (default 1000)
pdiv_err_desired	relative error desired (default 0.01)
output values, per pipe:	
pdiv_errP	estimated relative error of pressure change
pdiv_errT	estimated relative error of temperature change
pdiv_n_suggested	number of subdivisions suggested

freely configured by the user. For a purely 1-phase solution scen1 shown in Figure 3(a) and (c), the value of $prec$ decreases exponentially with the number of iterations and the solution procedure converges. For scen2, as seen in Figure 3(b) and (d), the procedure has cycling. In more detail, we see that there is a converging region for the 1-phase and a part of the 2-phase state, after which a temperature jump occurs, and oscillations are observed in the remaining pipe segment.

Along with the two main scenarios, we ran a number of additional simulations with small qset variations around the specified values. Simulations show stability of the effects, convergence in the 1-phase solution, and divergence in the 2-phase solution. The reason for this divergence is that EOS and the enthalpy function receive large derivatives in the phase transition region. These functions are actually jump-like for a pure substance and formally continuous for a mixture, but at a low concentration of impurities, the derivatives are still large.

A prototype example of such instability is the logistic map: $x_{i+1} = rx_i(1-x_i)$, which characterizes the behavior of simple iterations near the root $x = 1-1/r$. When r rises from 1, and

passes the value 3, the absolute value of the r.h.s. derivative of the logistic map equation exceeds 1, which is a critical value for the convergence of simple iterations. Below this value, the iterations converge. Above it, limit cycles appear, first with a multiplicity of 2, then they double, and finally the system goes to chaos.

Qualitatively, the same effects happen in our case. In principle, the stabilization algorithm helps to overcome such divergences, but for an ever higher derivative it becomes less and less effective. We are going to explore this problem in more detail in our future work. In order to overcome the divergence, we can try to adjust the weight parameter in the stabilizing algorithm. The dynamic solver behaves in much the same way as weighted relaxation with a low weight; with a decrease in the integration step, the stability of the integration also increases. As shown in Figure 1, high derivatives only occur for EOS in the form $\rho(T, P)$, changing variables to $P(T, \rho)$ could also be a solution of the problem.

At the same time, only scenarios in which phase transitions and associated divergences do not occur are to be considered within the scope of the technical task set. For such solutions, it is required to determine the proximity of the solution to the region of phase transitions. That can be done using the proximity-alarm algorithm described above.

Figure 4 shows the screenshots for scen1 solution in MYNTS GUI. At the top, there is the pipe geometry with the pressure profile shown in color. At the bottom, there is the solution on the (T, P)-plane, where a part of the phase envelope is also shown. The yellow disks show the proximity-alarm triggered in the given node for the values $dT = 1K$, $dP = 1bar$. The first 2 nodes near pset appear to be close to the spurious line on the phase diagram. The alarm in them can be canceled, because they are located top-right to the phase envelope, in the supercritical region. In general, this visual criterion is difficult to automate, since phase envelopes can have a more complex appearance than in the figures of this paper. Further, the figure shows how the solution trajectory passes at a safe distance from the phase envelope, providing the required CO_2 transport without phase transitions.

Testing pipe subdivision algorithm (scen3): here we consider the same pipe, filled with pure $\{CO_2, CH_4, H_2\}$ fluids, in various subdivisions (4). Other settings are given in Table I. The results are displayed in Figure 5.

The first row shows δP precision dependence on subdivision number n. Numerical experiments are shown by blue line with dots. This line closely follows $\sim L$ dependence on the length of pipe segment, for all fluids. The values of $N_{div} = 2^n$ for $\delta P = 0.1bar$ are shown. Two discretizations, Hofer-quad and Hofer-lin, shown by shades of blue, are almost coincident for CO_2 and coincident for other fluids.

The second and the third rows show the same numerical experiments in comparison with δP and δT empirical estimators. The estimators restrict the experiments from above almost everywhere, they are almost coincident for $\delta P(H_2)$ and $\delta T(CO_2)$, and closely approaching the experimental points at large n for other cases.

The last row shows the temperature distribution in these numerical experiments in comparison with the simplified model, used in the derivation of δT-estimator. Red and blue lines show the model results for maximal and minimal value of the heat capacity c_p, see Table II. For CO_2, these lines are different, they restrict the experimental subdivision, shown by gray lines with dots, from above and from below. For CH_4/H_2, the upper and lower c_p-values are very close and produce visually coincident model curves. Also typical for supercritical/liquid CO_2 are much larger values of c_p, in comparison with gaseous CH_4/H_2. This leads to a larger characteristic length x_1 and a slower temperature drop over the length.

Details: the simplified model should not coincide with simulation exactly, it contains only the simplest $c_p dT$ term in the temperature equation, while the simulation is more precise, contains additional terms, such as Joule-Thomson effect, gravity term, etc. In the cases considered, the pressure drop is small and the pipe is laid horizontally, so such effects are negligible. However, they can be activated in other scenarios. Also, the simplified model is valid only for constant c_p and x_1, while their variation over the pipe makes the model solution more approximate. In our simulation, the c_p dependence on pressure and temperature is computed by the GERG module.

Applying pipe subdivision algorithm (scen4-6): in the next scenario scen4, we consider a natural gas network N1 of moderate size, shown in Figure 6 top left. It contains 100 nodes, connected by 111 edges, 34 of them are pipes. The pipe subdivision algorithm with default settings produced $n\_suggested$ value, visualized in the figure. The value is peaked at $n\_suggested = 31$ on a 47km long pipe, shown on a closeup (Figure 6 bottom left), with parameters shown in Figure 6 right. The pipe possesses a moderate flow and large $errT$, that defines the subdivision.

In the following scenario scen5, we consider the same 150km pipe as in scen1-3 experiments, filled by pure CO_2, other parameters selected as in scen3a. The first iteration of pipe subdivision algorithm produces: $errP = 0.173$, $errT = 2.04$, $n\_suggested = 205$. Again, T-estimator defines the subdivision. Taking the suggested value, the pipe is subdivided to 205 equal pieces, and the second iteration of the algorithm is applied. It produces $errP$ varied in the range $0.00389 - 7.39 \cdot 10^{-5}$ from inlet to outlet, and $errT$ inbetween $0.00973 - 0.01005$, the value $n\_suggested$ is now $1 - 2$. We see that $errT$ is very close to the desired value 0.01, while $n\_suggested$ is balanced on the border of this value, so that no further subdivision is necessary. The final result satisfies all criteria and can be accepted.

Details: $errP$ changes over segments, due to the coupling to a non-linear density profile. As a result, a uniform subdivision dx gives non-uniform subdivision $d\rho$, initially large, then smaller, while our estimation supposed uniform $d\rho$. In the given scenario, $errP < err\_desired$, this effect is not important. In other scenarios, if $errP > err\_desired$, one more iteration of the algorithm can be needed. The alternative is to construct an adaptive subdivision, following $d\rho/\rho$ profile.

In the next scenario scen6, we consider a set of 85 natural

gas networks of large size [19], provided for benchmarking by our industrial partner. Each has 3000 to 4000 edges, mostly pipes. One iteration of pipe subdivision algorithm has been applied. The output is shown as a histogram in Figure 7, here p_i is probability of location in the i-th bin. The results for $err\_desired = 0.01$ show that 68% of the pipes do not require subdivision ($n\_suggested \sim 1 - 2$), 27% require moderate subdivision ($n\_suggested \sim 3 - 20$), and only the remaining 5% require large subdivision ($n\_suggested > 20$).

VI. Conclusion

In this paper, we have considered a numerical simulation of the stationary process of CO_2 transport with impurities and phase transitions. We have developed the algorithms that allow to solve scenarios of CO_2 transport in the liquid or supercritical phase and to detect proximity to the phase transition region. We have analyzed a convergence of the solution algorithms in connection with fast and abrupt changes of the equation of state and the enthalpy function in the region of phase transitions.

The performed numerical experiments show that the scenarios with a single CO_2 phase converge. For the obtained temperature and pressure profiles, a conservative algorithm for detecting the proximity of phase transitions can be applied, giving the solution to the technical problem posed. At the same time, divergences can occur in scenarios with phase transitions due to the abrupt change of thermodynamic parameters. Questions about the possible suppression of these divergences as well as improved detection of phase transitions are the subject of our further work.

Also, in this paper, an algorithm for subdivision of pipes for achieving a required precision of simulation is constructed. The algorithm uses empirical formulas for conservative error estimation, derived on the basis of numerical experiments. The application of the algorithm to realistic $CO_2/CH_4/H_2$ transport scenarios shows a good correspondence of predicted and measured precision. An additional study is planned on implementation of higher order finite difference schemes and an adaptive non-uniform subdivision for further improvement of the efficiency of the algorithm.

Acknowledgments

The work has been supported by Fraunhofer research cluster CINES. We acknowledge support from Open Grid Europe GmbH in the development and testing of the software. We also thank the organizers and participants of the conference INFOCOMP 2023 for fruitful discussions.

References

[1] M. Anvari et al., “Simulation of pipeline transport of carbon dioxide with impurities”, in Proc. of INFOCOMP 2023, the 13th International Conference on Advanced Communications and Computation, pp. 1-6, IARIA, 2023.

[2] M. Nimtz, M. Klatt, B. Wiese, M. Kühn, and H.-J. Kraautz, “Modelling of the CO2 process- and transport chain in CCS systems – Examination of transport and storage processes”, Chemie der Erde – Geochemistry, vol. 70, suppl. 3, 2010, pp. 185-192.

[3] S. Liljemark, K. Arvidsson, M. T. P. Mc Cann, H. Tummescheit, and S. Velut, “Dynamic simulation of a carbon dioxide transfer pipeline for analysis of normal operation and failure modes”, Energy Procedia, vol. 4, 2011, pp. 3040-3047.

[4] M. Chaczykowski and A. J. Osiadacz, “Dynamic simulation of pipelines containing dense phase/supercritical CO2-rich mixtures for carbon capture and storage”, International Journal of Greenhouse Gas Control, vol. 9, 2012, pp. 446-456.

[5] P. Aursand, M. Hammer, S. T. Munkejord, and Ø. Wilhelmsen, “Pipeline transport of CO2 mixtures: Models for transient simulation”, International Journal of Greenhouse Gas Control, vol. 15, 2013, pp. 174-185.

[6] L. Raimondi, “CO2 Transportation with Pipelines - Model Analysis for Steady, Dynamic and Relief Simulation”, Chemical Engineering Transactions, vol. 36, 2014, pp. 619-624.

[7] M. Drescher et al., “Towards a Thorough Validation of Simulation Tools for CO2 Pipeline Transport”, Energy Procedia, vol. 114, 2017, pp. 6730-6740.

[8] B. Chen, H. Guo, S. Bai, and S. Cao, “Optimization of process parameters for pipeline CO2 transportation with impurities”, IOP Conf. Series: Earth and Environmental Science, vol. 300, 2019, 022002.

[9] M. Vitali et al., “Risks and Safety of CO2 Transport via Pipeline: A Review of Risk Analysis and Modeling Approaches for Accidental Releases”, Energies, vol. 14, 2021, 4601.

[10] L. Raimondi, “CCS Technology - CO2 Transportation and Relief Simulation in the Critical Region for HSE Assessment”, Chemical Engineering Transactions, vol. 91, 2022, pp. 43-48.

[11] S. T. McCoy and E. S. Rubin, “An engineering-economic model of pipeline transport of CO2 with application to carbon capture and storage”, International Journal of Greenhouse Gas Control, vol. 2, 2008, pp. 219-229.

[12] X. Luo, M. Wang, E. Oko, and C. Okezue, “Simulation-based Techno-economic Evaluation for Optimal Design of CO2 Transport Pipeline Network”, Applied Energy, vol. 132, 2014, pp. 610-620.

[13] V. E. Onyebuchi, A. Kolios, D. P. Hanak, C. Biliyok, and V. Manovic, “A systematic review of key challenges of CO2 transport via pipelines”, Renewable and Sustainable Energy Reviews, vol. 81, part 2, 2018, pp. 2563-2583.

[14] H. Lu, X. Ma, K. Huang, L. Fu, and M. Azimi, “Carbon dioxide transport via pipelines: A systematic review”, Journal of Cleaner Production, vol. 266, 2020, 121994.

[15] T. Clees et al., “MYNTS: Multi-phYsics NeTwork Simulator”, in Proc. of SIMULTECH 2016, International Conference on Simulation and Modeling Methodologies, Technologies and Applications, pp. 179-186, SciTePress, 2016.

[16] T. Clees, I. Nikitin, and L. Nikitina, “Making Network Solvers Globally Convergent”, Advances in Intelligent Systems and Computing, vol. 676, 2018, pp. 140-153.

[17] A. Baldin, T. Clees, B. Klaassen, I. Nikitin, and L. Nikitina, “Topological Reduction of Stationary Network Problems: Example of Gas Transport”, International Journal On Advances in Systems and Measurements, vol. 13, 2020, pp. 83-93.

[18] A. Baldin et al., “Principal component analysis in gas transport simulation”, in Proc. of SIMULTECH 2022, International Conference on Simulation and Modeling Methodologies, Technologies and Applications, pp. 178-185, SciTePress, 2022.

[19] A. Baldin et al., “On Advanced Modeling of Compressors and Weighted Mix Iteration for Simulation of Gas Transport Networks”, Lecture Notes in Networks and Systems, vol. 601, pp. 138-152, 2023.

[20] J. Nikuradse, “Laws of flow in rough pipes”, NACA Technical Memorandum 1292, Washington, 1950.

[21] P. Hofer, “Error evaluation in calculation of pipelines”, GWF-Gas/Erdgas, vol. 114, no. 3, 1973, pp. 113-119 (in German).

[22] J. Mischner, H. G. Fasold, and K. Kadner, System-planning basics of gas supply, Oldenbourg Industrieverlag GmbH, 2011 (in German).

[23] O. Kunz and W. Wagner, “The GERG-2008 wide-range equation of state for natural gases and other mixtures: An expansion of GERG-2004”, J. Chem. Eng. Data, vol. 57, 2012, pp. 3032-3091.

[24] W. Wagner, Description of the Software Package for the Calculation of Thermodynamic Properties from the GERG-2008 Wide-Range Equation of State for Natural Gases and Similar Mixtures, Ruhr-Universität Bochum, 2022.

[25] ISO 20765-2: Natural gas – Calculation of thermodynamic properties – Part 2: Single-phase properties (gas, liquid, and dense fluid) for extended

ranges of application, International Organization for Standardization, 2015.
[26] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, Numerical Recipes in C, Cambridge University Press, 1992.
[27] C. T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, 1995.
[28] E. L. Allgower and K. Georg, Introduction to Numerical Continuation Methods, SIAM, 2003.
[29] J. Katzenelson, "An algorithm for solving nonlinear resistor networks", Bell System Technical J., vol. 44, 1965, pp. 1605-1620.
[30] M. J. Chien and E. S. Kuh, "Solving piecewise-linear equations for resistive networks", Int. J. of Circuit Theory and Applications, vol. 4, 1976, pp. 1-24.
[31] A. Griewank, J.-U. Bernt, M. Radons, and T. Streubel, "Solving piecewise linear systems in abs-normal form", Linear Algebra and its Applications, vol. 471, 2015, pp. 500-530.
[32] T.-P. Azevedo-Perdicoúlis, F. Perestrelo, and R. Almeida, "A note on convergence of finite differences schemata for gas network simulation", in Proc. of the 22nd International Conference on Process Control, pp. 274-279, IEEE, 2019.
[33] C. Himpe, S. Grundel, and P. Benner, "Next-gen gas network simulation", in: Progress in Industrial Mathematics at ECMI 2021, pp. 107-113, Springer, 2022.
[34] C. Himpe, S. Grundel, and P. Benner, "Model order reduction for gas and energy networks", J. Math. Industry, vol. 11:13, 2021, pp. 1-46.
[35] O. Schenk and K. Gärtner, "PARDISO", in: D. Padua (eds) Encyclopedia of Parallel Computing, Springer, 2011.

Mapping Technologies and Tools to the Activities of the Customer Experience Management Process

Marie-Noëlle Forget
Dept. of Analytics, Operations and IT
ESG UQAM
Montreal, Canada
email: forget.marie-noelle@uqam.ca

Pierre Hadaya
Dept. of Analytics, Operations and IT
ESG UQAM
Montreal, Canada
email: hadaya.pierre@uqam.ca

***Abstract*—While good Customer Experience Management (CEM) is said to give organizations a competitive advantage, we still lack some guidance in how to optimize CEM, especially from a technological standpoint. Indeed, despite the many CEM technologies and tools discussed in the literature, none can support the CEM process from beginning to end. Moreover, we do not know how to integrate the many CEM technologies and tools to allow for seamless management of the customer experience. This paper identifies 52 CEM technologies and tools and maps them to each activity of the CEM process in which they can be used. It also proposes four preliminary integration guidelines that can help organizations in their integration efforts.**

Keywords-customer experience management; tools; technologies; literature review.

I. INTRODUCTION

This article extends the work that was presented at the BUSTECH 2023 conference [1]. Gaining and sustaining a competitive advantage is a daunting challenge in today's fast-changing environment. According to some, customer experience is what organizations will now have to compete in to stand out from their competitors [2]. This would be the case in any industry, whether it be banking [3][4][5], hospitality and tourism [6][7][8], communications [9][10], retail [11][12], and online commerce [13][14]. For instance, supply chain management, which used to focus on product development and order processing, is now also looking at Customer Experience Management (CEM or CXM) as a source of competitive advantage [15].

Customer experience can be defined as the "customer sensorial, physiological, psychological responses such as cognitive as well as affective responses evoked by customer direct (offline) and indirect (online) interactions with the firm or firm offerings across all the touch points throughout the customer purchase journey" [16]. The emotional and sensorial components of the customer experience, as well as the fact that it encompasses all interactions that a customer has with a brand [17], make it challenging to manage.

To date, several studies have been conducted to investigate what good CEM entails. Others have also explored the usefulness and the effect of some technologies (e.g., software and algorithms) and tools (e.g., methods and canvas) in CEM. Indeed, the customer experience and its management are closely related to Information Technology (IT). To begin with, customers often learn about a brand through online advertising and social media [18]. They interact with organizations through various channels, including on their mobile phones, and interact closely with technology even in brick-and-mortar stores (e.g., self-check-out [19]). The simple fact of using IT can help organizations improve customer experience even in unexpected settings, such as in temples [20].

Unfortunately, properly leveraging CEM technologies and tools to gain a competitive advantage is an arduous task. First, there are a plethora of technologies and tools available on the market and the literature does not provide any guidance as to which ones can be used to support the CEM process throughout. Indeed, most studies on CEM technologies focus only on one or a few activities of the CEM process. This problem is exacerbated by the fact that the "CEM software" that have recently begun appearing on the market are not yet mature and only focus on a limited number of activities of the CEM process, such as collecting and analyzing feedback from customers. Managers using these new technologies thus risk neglecting crucial activities of CEM to the detriment the organization's competitive advantage. Second, the academic and professional literatures do not provide any guidance as to how CEM technologies and tools can be used in conjunction to support the CEM process. Organizations must know how to go about integrating their CEM technologies and tools to optimize the CEM process and seamlessly manage the customer experience. Only then can they gain a sustainable competitive advantage from their CEM.

To offset these important limits, the objective of this paper is to determine which technologies and tools can support each activity of the CEM process as well as to propose a set of preliminary technology integration guidelines to enable the seamless management of the CEM process from beginning to end. To do so, we first identify the activities that make up the CEM process (Section II). Second, we review the literature to identify technologies and tools that can support the CEM process (Section III). Third, we map the identified technologies and tools to the activities of the CEM process (Section IV). Finally, we propose four preliminary integration guidelines that can be used as a starting point to integrate CEM technologies and tools (Section V).

II. IDENTIFYING THE CEM PROCESS ACTIVITIES

Several concepts led the way to CEM, including consumer behavior, service quality, and relationship marketing [21]. The concept of customer experience first appeared in the literature in the late 1990s, when Pine and Gilmore [22] stated in the Harvard Business Review that providing experiences was the next discipline that would enable organizations to remain competitive. They argued that although some confuse the delivery of an experience with that of a service, they are two distinct approaches. According to the authors, while products and services are external to the customer, "experiences are inherently personal, existing only in the mind of an individual who has been engaged on an emotional, physical, intellectual, or even spiritual level" [22].

Customer experience is described as the "aggregate of feelings, perceptions and attitudes" formed by the customer throughout his journey, at each touchpoint [23]. Since customer experience is such a complex, multi-faceted concept, its management is naturally just as intricate. CEM is defined as "the cultural mindsets toward CEs, strategic directions for designing CEs, and firm capabilities for continually renewing CEs, with the goals of achieving and sustaining long-term customer loyalty" [24].

While there is no agreed-upon CEM process in the literature, several similar processes are suggested, some of which are adapted to a particular industry. The most comprehensive come from two literature reviews that had the objective of proposing a CEM process. First, Du Plessi and de Vries [25] conducted a literature review and used inductive thematic analysis to describe the CEM process in four steps and twelve sub-steps. The first step, Customer Experience Understanding, includes segmenting customers and defining their needs. The second step, Customer Experience Design, consists of mapping the desired customer journeys. The third step, Customer Experience Measurement, consists of monitoring the customer experience. The last step, Customer Experience Change Implementation, consists of identifying the gaps between the current and the desired experience, and taking action to close those gaps. Rahimian, ShamiZanjani, Manian and Esfidani's [26] literature review, in turn, proposed four high-level CEM stages (Customer Identification, Customer Experience Design, Customer Experience Implementation, and Customer Experience Monitoring), each containing steps. These stages and steps were identified through a systematic review of the literature on the hotel, tourism, and hospitality industry. The four stages are very similar to the steps identified by [25] and cover approximately the same activities.

TABLE I. CEM PROCESS

Phase	Step/Activity	Reference(s)
1) Customer Identification	1- Assessing customers' characteristics and past experiences with other competitors and understanding their needs, expectations, and values	[25][26][27][28]
	2- Segmenting customers	[25][26]
2) Customer Experience Design	1- Developing a plan/strategy	[26][27][28]
	2- Designing/mapping customer journeys and touchpoints	[25][26][27][28]
	3- Prioritizing touchpoints	[26]
3) Customer Experience Implementation	1- Identifying gaps in experience design versus current organizational capability	[25]
	2- Prioritizing improvement initiatives	[25][27][28]
	3- Implementing required changes to IT systems and other support systems	[27][28]
	4- Implementing the improvement initiatives	[26][28]
	5- Interacting with customers and personalizing services	[26]
4) Customer Experience Monitoring	1- Defining internal and external measurements	[25]
	2- Monitoring experiences	[25][26][27][28]
	3- Adapting and deploying improvement initiatives	[27][28]

In addition to [25] and [26], other studies propose activities to manage the customer experience. For instance, Popa and Barna [27] proposed "seven steps to better customer experience management". Johnston and Kong [28], for their part, proposed a "road map for improving the customer experience" containing ten CEM activities.

Based on [25] [26][27][28], we thus characterize the CEM process as comprising four complementary phases and 13 steps/activities (see Table I). The first phase, Customer Identification, aims to gain a better understanding of the customers. It comprises two activities. During the first activity, *Assessing customers' characteristics and past experiences with other competitors and understanding their needs, expectations, and values*, organizations build knowledge of their customers. During the second activity, *Segmenting customers*, customers are divided into segments that share certain characteristics.

The second phase, Customer Experience Design, aims to determine what is the desired customer experience. During the first activity, *Developing a plan/strategy*, the organization's global customer experience strategy is determined. During the second activity, *Designing/mapping customer journeys and touchpoints*, the desired customer experience journey is mapped, including all touchpoints. During the third activity, *Prioritizing touchpoints*, the organization decides which touchpoints should be the main interaction points with the customer and thus, which touchpoints should receive the most attention.

The third phase, Customer Experience Implementation, aims to implement the desired customer experience that was designed during the second phase. During the first activity, *Identifying gaps in experience design versus current organizational capability*, the gaps that need to be filled in order to implement the desired customer experience are identified. During the second activity, *Prioritizing improvement initiatives*, the initiatives that will allow the implementation of the desired customer experience are

prioritized. During the third activity, *Implementing required changes to IT systems and other support systems*, all changes to IT systems (e.g., developing customer-centric information architecture, deploying workflow-based tools) and to other support systems (e.g., revise employee training documentation) that need to be done to allow the implementation of the improvement initiatives are implemented. During the fourth activity, *Implementing the improvement initiatives*, touchpoints are changed and/or developed to reflect the desired customer experience designed in the second phase. During the fifth activity, *Interacting with customers and personalizing services*, changes that were required to implement the desired customer experience designed in the second phase are implemented and the organization interacts with the customers through the upgraded and/or new touchpoints.

The fourth phase, Customer Experience Monitoring, aims to monitor the customer experience to identify issues and opportunities for improvement. During the first activity, *Defining internal and external measurements*, measurements and escalation mechanisms that will be used to monitor the customer experience are determined. During the second activity, *Monitoring experiences*, the measurements defined in the first activity are used to assess the performance of the current customer experience, to flag issues, and to enhance the organization's understanding of its customers. During the third activity, *Adapting and deploying improvement initiatives*, the customer experience is adjusted according to the data collected during the second activity.

To conclude this section, it is important to mention that although the phases and activities are presented in a logical sequence, the CEM process is iterative, since the customers, as well as their experiences, are constantly changing.

III. IDENTIFYING CEM TECHNOLOGIES AND TOOLS: A LITERATURE REVIEW

This section first details the methodology followed to identify CEM technologies and tools and second, exposes our findings from the literature review.

A. Methodology

We conducted a literature review with the objective of identifying CEM technologies and tools. The flow diagram of the literature review is presented in Figure 1. First, we searched the online databases ABI/INFORM and Business Source Complete. We limited our scope to peer-reviewed articles. A preliminary search informed us that relevant articles used at least the expression 'customer experience' and most contained the word 'technology' or 'technologies'. We also decided to use the keywords 'software' and 'tool' to be as comprehensive as possible. We consequently used three search strings, presented in Figure 1. The expression "technolog*" was used to account for both singular and plural forms of the word. To focus on articles more closely related to our research topic, we search everywhere except full text. We thereby identified 572 articles in ABI/INFORM and 645 articles in Business Source Complete and ended up with 840 articles once the duplicates were removed.

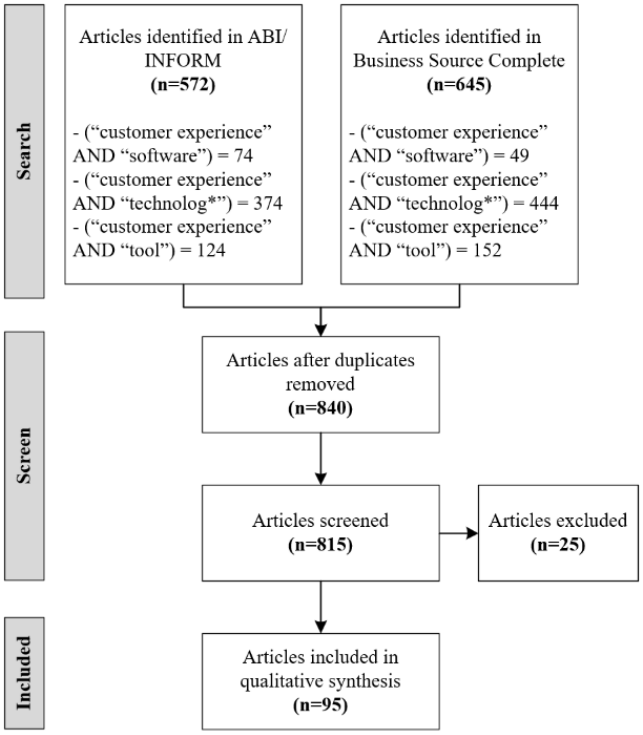


Figure 1. PRISMA flow diagram of the review process.

The second step consisted of screening the identified studies. We read the identified articles' title and abstract. Twenty-five (25) articles were excluded because they were not in English.

In the third step, we read the articles relevant to our research topic in their entirety. We found 95 articles that proposed CEM technologies and/or tools. It should be noted that the use of the Internet was not retained, as it is broad and omnipresent in all organizations nowadays. Websites and technologies that are specific to a certain industry, such as exhibition service systems [29], were also excluded. The included articles' publication dates range from 1998 to 2023, with only one article published before 2003 and 74 since 2018. Hence, as shown in Figure 2, almost 70% of the articles included in our qualitative analysis were published in the last five years.

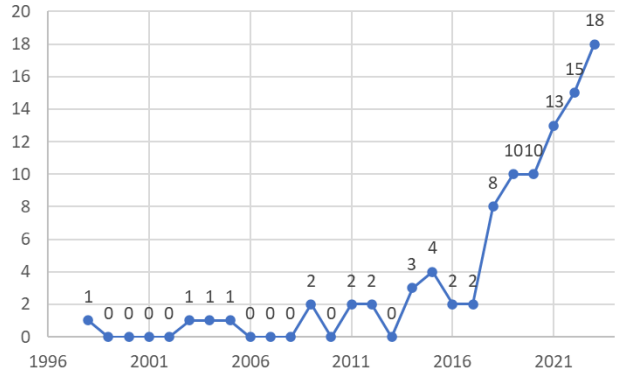


Figure 2. Articles published per year.

B. Results

After extracting the technologies and tools identified in the 95 articles found in the literature and synthesizing then, we came up with a list of 52 CEM technologies and tools. They are presented in alphabetical order in Table II. A few similar technologies were gathered as one because they were often talked about as one and were difficult to separate. For instance, we identified "analytics" as encompassing text analytics, descriptive analytics, predictive analytics, and prescriptive analytics.

TABLE II. CEM TECHNOLOGIES AND TOOLS

Technology/tool	Reference(s)
Analytics (text analytics, descriptive analytics, predictive analytics, prescriptive analytics)	[30][31][32][33]
Artificial Intelligence (AI)/Machine learning	[30][34][35][36][37][38][39][40] [41][42][43][44][45]
Augmented Reality (AR)	[12][13][39][46][47][48][49][50] [51][52][53][54][55][56][57][58] [59]
Balanced Scorecard (BSC)	[60]
Big data, data mining	[31][61]
Biometrics	[62]
Call center technology (Voice Response Units (VRU) and Interactive Voice Response (IVR))	[63]
Chatbots	[30][32][36][51][64][65][66][67] [68][69][70][71]
Chat GPT	[72]
Cloud computing	[73]
CRM tools/software	[7][74]
Customer experience/journey mapping and modeling tools	[30][31][75][76][77]
Customer identity card	[63]
Database management	[7]
Digital kiosk	[19][57][63][78][79][80]
Digital twins	[35][81]
Diminished Reality (DR)	[55]
Drones	[12]
(Face) recognition technologies	[32][39][45][80][82]
Exoskeletons/Exosuits	[80]
Eye-tracker	[82]
Geolocation technology, location-based and wearables	[30][35][39][57][80][82]
Human Enhancement Technology (HET)	[41][83]
Human resources software	[7]
Immersive technology	[84]
In-store tablet, touchpoint, monitor, LCD screen, multi-touch display	[32][46][85][86][80][87][88]
Internet of Things (IoT)	[12][89]
Marketing technology (pop-up ads, targeted ads, coupons)	[90]
Messaging applications	[30][32]
Metaverse	[91]
Mixed-Reality (MR)	[51][52]
Near Field Communication (NFC)	[92]
Net promoter score	[30][31][32]
Neuroscience	[35]
On-line catalogues	[93]
Product-service system	[94]
Property management system	[7]
Quick Response (QR) code	[46]
Radio Frequency Identification (RFID)	[86][95] [57]
Robotic Process Automation (RPA)	[96][97]
Service robots	[57][98]
Self-service technologies (SST)	[12][19][36][57][62][99][100] [101][102][103][104][105][106]
Smart services/devices	[107][108] [57]
Smart wearable devices	[57][109]
Social media	[7][8][19][32][35][110][111]
Technologies/applications enabling co-creation	[18][112]
Technology for faster billing (automatic checkout, mobile checkout, mobile payment)	[57][113]
Video recording	[114]
Virtual assistant	[51]
Virtual Reality (VR)	[35][39][46][51][52][55][57][115] [116]
Voice assistant	[71]
Web services	[117]

It is also worth noting that some of the technologies identified are closely related to one another. For example, many use Artificial Intelligence (AI), such as chatbots, Internet of Things (IoT), and both Virtual Reality (VR) and Augmented Reality (AR). We kept them separate because some authors proposed specific usage for each of the technology, and as such we did not want to lose their individual purpose.

IV. MAPPING THE TECHNOLOGIES AND TOOLS TO EACH ACTIVITY OF THE CEM PROCESS

After having extracted all the technologies and tools found in the literature, we identified, for each of them, in which activities they could be useful according to the way they were described in the studies from which they were extracted. Table III presents all the technologies and tools (identified in Section III) that can assist management in each activity of the CEM process (described in Section II). Some technologies and tools can be useful in more than one activity and thus appear more than once in the table.

It is no surprise that the activities in the first phase of the CEM process, Customer Identification, can be supported by technologies such as analytics, data mining, and database management. These technologies are helpful in segmenting customers, assessing their characteristics, and understanding their needs. The same technologies can support the two activities of this first phase.

The second phase of CEM, Customer Experience Design, consists primarily of designing the customer journeys and experiences as a whole. Customer experience/journey mapping and modeling tools are thus essential. Co-creation is a prominent concept in customer experience literature. To that end, technologies/applications enabling co-creation can allow organizations to engage customers in the design of their customer experience. Although this phase is central to CEM, very few supporting technologies and tools were found. Indeed, none were found for the activity *Developing a plan/strategy* and only one was found for *Prioritizing touchpoints*, i.e., Balanced Scorecard (BSC). Moreover, BSC

TABLE III. TECHNOLOGIES AND TOOLS RELEVANT TO EACH ACTIVITY OF THE CEM PROCESS

Phases	Step/Activity	Relevant Technologies and Tools
Customer Identification	Assessing customers' characteristics and past experiences with other competitors and understanding their needs, expectations, and values	- Analytics - Big data, data mining - Database management
	Segmenting customers	- Analytics - Big data, data mining - Database management
Customer Experience Design	Developing a plan/strategy	- None
	Designing/mapping customer journeys and touchpoints	- Customer experience/journey mapping and modeling tools - Technologies/applications enabling co-creation
	Prioritizing touchpoints	- Balanced Scorecard (BSC)
Customer Experience Implementation	Identifying gaps in experience design versus current organizational capability	- Customer experience/journey mapping and modeling tools
	Prioritizing improvement initiatives	- Balanced Scorecard (BSC)
	Implementing required changes to IT systems and other support systems	- Database management
	Implementing the improvement initiatives	- Artificial Intelligence (AI)/Machine learning - Augmented Reality (AR) - Call center technology (Voice Response Units (VRU) and Interactive Voice Response (IVR)) - Chatbots - Customer identity card - CRM tools/software - Database management - Digital kiosk - Digital twins - Drones - (Face) recognition technologies - Geolocation technology, location-based and wearables - Human resources software - In-store tablet, touchpoint, monitor, LCD screen, multi-touch display - Internet of Things (IoT) - Marketing technology (pop-up ads, targeted ads, coupons) - Messaging applications - Near Field Communication (NFC) - On-line catalogues - Property management system - Quick Response (QR) code - Radio Frequency Identification (RFID) - Robotic Process Automation (RPA) - Self-service technologies - Service robots - Smart services/devices - Smart wearable devices - Social media - Technologies/applications enabling co-creation - Technology for faster billing (automatic checkout, mobile checkout, mobile payment) - Virtual Reality (VR) - Web services
	Interacting with customers and personalizing services	- Artificial Intelligence (AI)/Machine learning - Augmented Reality (AR) - Call center technology (Voice Response Units (VRU) and Interactive Voice Response (IVR)) - Chatbots - Customer identity card - CRM tools/software - Database management - Digital kiosk - Digital twins - Drones - (Face) recognition technologies - Geolocation technology, location-based and wearables - Human resources software - In-store tablet, touchpoint, monitor, LCD screen, multi-touch display - Internet of Things (IoT) - Marketing technology (pop-up ads, targeted ads, coupons) - Messaging applications - Near Field Communication (NFC) - On-line catalogues - Property management system - Quick Response (QR) code - Radio Frequency Identification (RFID) - Robotic Process Automation (RPA) - Self-service technologies - Service robots - Smart services/devices - Smart wearable devices - Social media - Technologies/applications enabling co-creation - Technology for faster billing (automatic checkout, mobile checkout, mobile payment) - Virtual Reality (VR) - Web services
Customer Experience Monitoring	Defining internal and external measurements	- None
	Monitoring experiences	- Analytics - Big data, data mining - Call center technology (Voice Response Units (VRU) and Interactive Voice Response (IVR)) - Chatbots - CRM tools/software - Database management - Digital twins - (Face) recognition technologies - Geolocation technology, location-based and wearables

		- In-store tablet, touchpoint, monitor, LCD screen, multi-touch display - Internet of Things (IoT) - Messaging applications - Neuroscience - Social media - Radio Frequency Identification (RFID) - Smart services/devices - Smart wearable devices - Video recording
	Adapting and deploying improvement initiatives	- Artificial Intelligence (AI)/Machine learning - Augmented Reality (AR) - Call center technology (Voice Response Units (VRU) and Interactive Voice Response (IVR)) - Chatbots - Customer identity card - CRM tools/software - Database management - Digital kiosk - Digital twins - Drones - (Face) recognition technologies - Geolocation technology, location-based and wearables - Human resources software - In-store tablet, touchpoint, monitor, LCD screen, multi-touch display - Internet of Things (IoT) - Marketing technology (pop-up ads, targeted ads, coupons) - Messaging applications - Near Field Communication (NFC) - On-line catalogues - Property management system - Quick Response (QR) code - Radio Frequency Identification (RFID) - Robotic Process Automation (RPA) - Self-service technologies - Service robots - Smart services/devices - Smart wearable devices - Social media - Technologies/applications enabling co-creation - Technology for faster billing (automatic checkout, mobile checkout, mobile payment) - Virtual Reality (VR) - Web services

was discussed in only one of the articles found in the literature. The activities included in this phase are thus neglected in the literature from a technical standpoint.

The third phase, Customer Experience Implementation, is the one for which the most useful technologies and tools were found. More specifically, a large number of technologies can support the activities *Implementing the improvement initiatives* and *Interacting with customers and personalizing services*. Indeed, there are many different technologies and tools that allow organizations to interact and engage with their customers, such as Artificial Intelligence (AI), Augmented reality (AR), Virtual Reality (VR), call centers, chatbots, digital kiosks, monitors, messaging applications, social media, smart devices, etc. Digital twins, which are "a dynamic virtual representation of a physical object or system across its lifecycle" [81] also allow organizations to interact with customers, as well as to collect data which is useful in the next phase of the CEM process, i.e., Customer Experience Monitoring. Looking at Table III, it is evident that there is a keen interest in such technologies and tools in the literature. However, other activities included in this phase are not supported by nearly as many technologies. Indeed, only customer experience/journey mapping and modeling tools can be used to support the activity *Identifying gaps in experience design versus current organizational capability*. Similarly, the activity *Prioritizing improvement initiatives* can be supported only by BSC, at least among the identified technologies and tools in our review of the literature.

The fourth and last phase of the CEM process, i.e., Customer Experience Monitoring, can also be supported by a fairly large number of technologies and tools. This is especially true for the activity *Monitoring experiences*. Indeed, analytics, data mining, chatbots, geolocation technology, Internet of Things (IoT), and video recording are all examples of technologies and tools that can be used to measure customer experience and flag incidents. CRM tools/software can also be used and they themselves contain powerful analytics capabilities. Most of the technologies identified for this activity were also identified for the activity *Interacting with customers and personalizing services* in the previous phase. This is because most of them can gather feedback from customers as they are interacting with them. For instance, a chatbot can interact with a customer at a touchpoint, being part of the customer's journey. The input from this customer in the chatbot can then be analyzed and used to monitor its experience. Another example is smart wearable devices. While these devices can enrich the customer's experience, they can also collect data that can be used to monitor customer experience, such as the path that customers take in a physical store. Indeed, this technology could identify possible areas of improvements in the layout of the store. Other technologies and tools are especially useful for monitoring experiences. For example, video recording will not allow interaction with customers, but can be helpful in assessing the customer experience. While many technologies and tools were found to support *Monitoring experiences*, there is however an activity in this last phase of the CEM process that is not supported by any technology, which is *Defining internal and external measurements*.

To conclude this section, three main observations arise from mapping the CEM technologies and tools to the activities of the CEM process. First, when looking at the distribution of technologies and tools in Table III in comparison with the activities of the CEM process identified in Table I, we can see that there is a clear discrepancy between the portion of literature interested in the CEM process and the portion of literature interested in CEM technologies and tools. Indeed, while the activity *Designing/mapping customer journeys and touchpoints* (Phase 2) was described by all references included in Table I, only two technologies and tools were found that can support this activity. The activity *Interacting*

with customers and personalizing services (Phase 3), for its part, was only proposed by one of the references included in Table I, but can be supported the largest number of technologies and tools. This discrepancy could potentially be a hindrance to good CEM. Indeed, it is crucial that organizations provide the right customer experience and Designing the Customer Experience is thus a crucial phase of CEM. Indeed, the implementation can only be as good as the design. Therefore, the activities in the phases Customer Identification and Customer Experience Design should not be neglected and there seems to be an opportunity to explore what technologies and tools could potentially support customer experience design.

Second, while there is an emphasis on the different technologies and tools that allow different types of interactions with the customers, simply using these technologies and tools is not guaranteed to improve the customer experience. In fact, some have observed that, in some cases, they could rather have a negative impact on customer experience (e.g., self-checkout [118]).

Finally, and most importantly, while a large number of CEM technologies and tools was identified, no single technology can support all the activities of the CEM process. Yet, for CEM to procure a real competitive advantage, it is essential that the technologies supporting the CEM process be integrated, thereby allowing the optimization of the CEM process for seamless management of the customer experience from beginning to end.

V. CEM TECHNOLOGIES AND TOOLS: PRELIMINARY INTEGRATION GUIDELINES

In the previous section, we concluded that the technologies and tools supporting the CEM process should be integrated for CEM to procure a sustainable competitive advantage. Of course, this is easier said than done. Considering the number of CEM technologies and tools identified as well as the countless combinations of technologies possible, there is no single way to go about it. In this section, we propose four guidelines – one per phase – to properly integrate the technologies and tools for the seamless integration of the CEM process.

In the first phase, "Customer Identification', we must optimize the usage and the update of customer knowledge. Therefore, the technologies and tools used to support the activity *Assessing customers' characteristics and past experiences with other competitors and understanding their needs, expectations, and values* should be integrated with the technologies and tools used in the activities *Interacting with customers and personalizing services* and *Monitoring experiences* (see Figure 3). Indeed, the data collected while interacting with customers and while monitoring their experience should be added to the data used to gain a better understanding of the customers. Additionally, the customer's data should be used when interacting with the customer to improve customer service and personalize its experience.

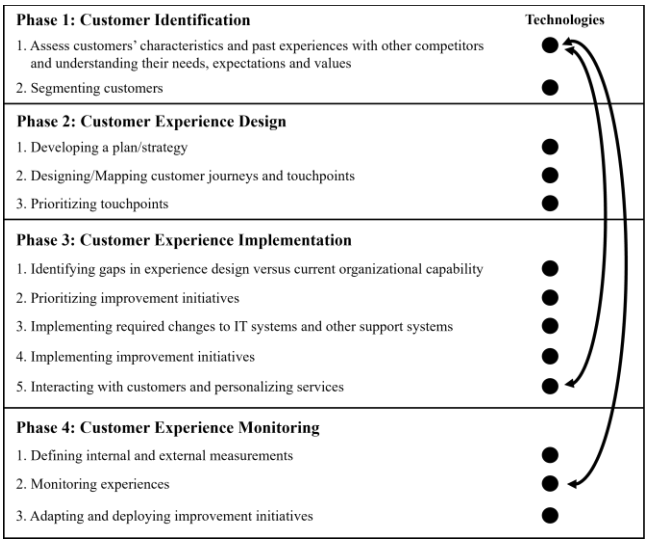


Figure 3. Integration Guideline for Customer Identification.

In the second phase, "Customer Experience Design', we must keep the customer strategy up to date as well as ensure that any changes to said strategy are reflected it the customer experience implementation. Therefore, while we did not find any technologies and tools relevant to the activity *Developing a plan/strategy*, this activity should be closely integrated with the technologies and tools used to support three activities, i.e., *Implementing improvement* initiatives, *Prioritizing improvement initiatives*, and *Monitoring experiences* (see Figure 4). Indeed, these integrations are required for updating the customer strategy and adjusting the customer experience. Since the customer experience is not static, it is crucial to adapt the customer experience strategy in accordance with the data collected while monitoring the customer experience, such as customers' feedback. Then, the changes to the customer experience strategy should be reflected in the prioritization of the improvement initiatives as well as in the implementation

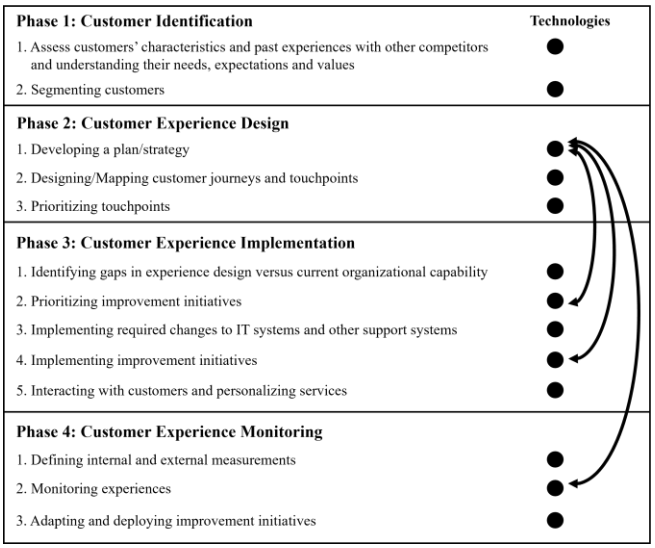


Figure 4. Integration Guideline for Customer Experience Design.

of improvement initiatives. Of course, changes to the strategy will eventually be reflected in the customer experience by following the normal CEM process. However, directly integrating the technologies and tools supporting these critical activities can allow for more agility and quicker response time, as small changes can be applied without going through the whole process.

In the third phase, "Customer Experience Implementation", we must ensure that the right initiatives are implemented, that they have the desired effect on customer experience, and that the current customer experience design is being updated once the initiatives are implemented. Therefore, the technologies and tools used to support the activity Implementing improvement initiatives should be integrated with the technologies and tools used to support the activities *Designing/mapping customer journeys and touchpoints*, *Prioritizing improvement initiatives*, and *Monitoring experiences* (see Figure 5). Indeed, the improvement initiatives should be implemented according to the prioritization done in *Prioritizing improvement initiatives* which, as mentioned, is subject to change if the customer strategy itself changes. The implementation of new initiatives should be closely monitored to allow for prompt adjustment if these initiatives have an unexpected negative impact on customer experience. Once the initiatives implemented, the current customer experience journey map should also be updated consequently.

In the fourth phase, "Customer Experience Monitoring", we must of course monitor the customer experience. The three previous guidelines propose integrating the activity *Monitoring experiences* to activities from the three first phases of the CEM process. Another important activity of the fourth phase is *Adapting and deploying improvement initiatives*. We propose integrating the technologies and tools used to support this activity to those used to support the activities *Prioritizing improvement initiatives*, *Implementing required changes to IT systems and other support systems*, and *Implementing improvement initiatives* (see Figure 6). Indeed, while the literature proposes a distinct activity for adapting improvement initiatives following the monitoring of the customer experience, the concrete actions done in this activity are very similar, if not identical, to those done in the activities *Implementing required changes to IT systems and other support systems* and *Implementing improvement initiatives*. This illustrates the iterative nature of the CEM process.

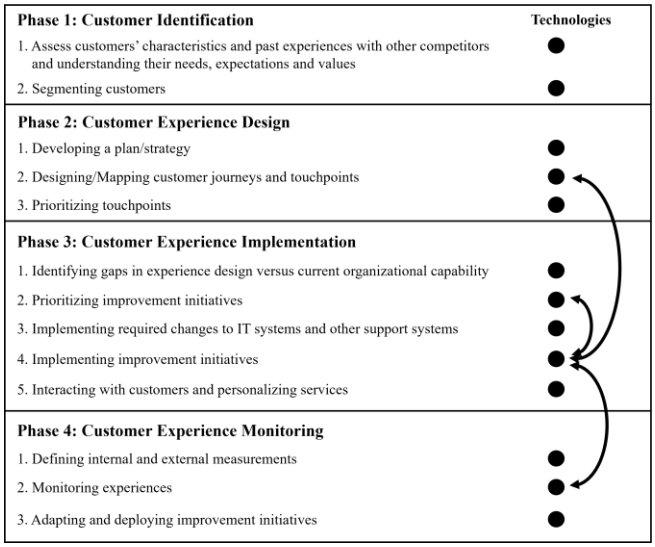


Figure 5. Integration Guideline for Customer Experience Implementation.

Finally, the integration between the different technologies and tools should allow the right people to have access to the right, updated data in a timely manner. As mentioned previously, the CEM process is iterative since the customer experience is in constant evolution. Therefore, all the activities required to manage the customer experience should also be adapted constantly. Organizations should be wary of creating punctual integration points, as the CEM process is hardly a one-and-done initiative. These four guidelines are only preliminary, but they still offer a good a starting point to determine how to integrate the technologies and tools that support the CEM process.

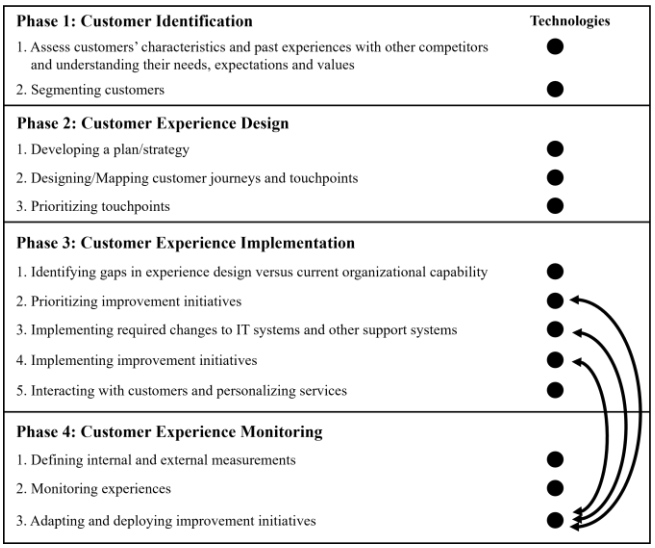


Figure 6. Integration Guideline for Customer Experience Monitoring.

VI. CONCLUSION

The objective of this paper was to determine the technologies and tools that can be used to support each activity of the CEM process and to offer guidelines for integrating these technologies and tools. We found 52 technologies and tools, and we mapped them to each activity of the CEM process in which they can be useful. We also proposed four preliminary guidelines, each related to one phase of the CEM process, to consider for the integration of CEM technologies and tools.

The results of this literature review have several contributions. First, they can help management identify relevant technologies and tools to support the CEM process, thereby leading to a better customer experience and, in turn, a

sustainable competitive advantage. Second, they highlight the fact that no single technology is sufficient to manage the whole CEM process, raising caution with managers and minimizing the risk that they neglect some CEM activities. Third, the four preliminary integration guidelines represent a good starting point to determine how to go about integrating all CEM technologies and tools.

This literature review shed light on gaps in the literature that could be the basis for future research avenues. First, more research should be conducted on technologies and tools that can support the activities for which few to no technologies were found, especially the activities that are key to CEM according to the literature, such as *Designing/mapping customer journeys and touchpoints*. Second, as we found that some technologies can actually be detrimental to the customer experience, more research should explore how to best select the technologies and tools to support CEM, especially the ones with which customers might interact. Third, more studies should further explore integration guidelines that could help organizations in integrating all technologies supporting their CEM process from beginning to end, thereby improving the customer experience.

REFERENCES

[1] M.-N. Forget, P. Hadaya, and É. Blanchet, "Technologies and tools in support of the customer experience management process: A literature review," in *BUSTECH 2023 : The Thirteenth International Conference on Business Intelligence and Technology*, Nice, France, 2023, pp. 18-23.

[2] T. Keiningham, J. Ball, S. Benoit, H. L. Bruce, A. Buoye, J. Dzenkovska *et al.*, "The interplay of customer experience and commitment," Journal of Services Marketing, vol. 31, no. 2, pp. 148-160, 2017.

[3] K. N. Kumar and P. R. Balaramachandran, "Robotic process automation-a study of the impact on customer experience in retail banking industry," Journal of Internet Banking and Commerce, vol. 23, no. 3, pp. 1-27, 2018.

[4] A. Kyguolienė and N. Makutėnas, "Measuring gen-y customer experience in the banking sector," Management of Organizations: Systematic Research, vol. 78, no. 1, pp. 77-93, 2017.

[5] C. I. Mbama, P. Ezepue, L. Alboul, and M. Beer, "Digital banking, customer experience and financial performance: An international journal," Journal of Research in Interactive Marketing, vol. 12, no. 4, pp. 432-451, 2018. http://dx.doi.org/10.1108/JRIM-01-2018-0026

[6] J. Kandampully, Z. Tingting, and E. Jaakkola, "Customer experience management in hospitality," International Journal of Contemporary Hospitality Management, vol. 30, no. 1, pp. 21-56, 2018. http://dx.doi.org/10.1108/IJCHM-10-2015-0549

[7] D. Sharma, "Enhancing customer experience using technological innovations: A study of the indian hotel industry," Worldwide Hospitality and Tourism Themes, vol. 8, no. 4, pp. 469-480, 2016.

[8] M. Veríssimo and N. Menezes, "Social media as a tool to enhance customer experience in hospitality industry," Revista Portuguesa de Marketing, vol. 38, no. 34, pp. 23-30, 2015.

[9] S. Shrivastava, "Digital disruption is redefining the customer experience: The digital transformation approach of the communications service providers," Telecom Business Review, vol. 10, no. 1, pp. 41-52, 2017.

[10] S. Joshi, "Enhancing customer experience: An exploratory study on the role of retailer as an effective touch-point for enhancing customer experience for cellular service providers," Drishtikon: A Management Journal, vol. 5, no. 1, pp. 88-100, 2013.

[11] J. C. Bustamante and N. Rubio, "Measuring customer experience in physical retail environments," Journal of Service Management, vol. 28, no. 5, pp. 884-913, 2017. https://doi.org/10.1108/JOSM-06-2016-0142

[12] M. Rodríguez, F. Paredes, and Y. Gaofeng, "Towards future customer experience: Trends and innovation in retail," Foresight and STI Governance, vol. 10, no. 3, pp. 18-28, 2016. http://dx.doi.org/10.17323/1995-459X.2016.3.18.28

[13] H. Kumar, "Augmented reality in online retailing: A systematic review and research agenda," International Journal of Retail & Distribution Management, vol. 50, no. 4, pp. 537-559, 2022. https://doi.org/10.1108/IJRDM-06-2021-0287

[14] S. Rose, N. Hair, and M. Clark, "Online customer experience: A review of the business ‐ to ‐ consumer online purchase context," International Journal of Management Reviews, vol. 13, no. 1, pp. 24-39, 2011.

[15] A. M. Fawcett, S. E. Fawcett, M. B. Cooper, and K. S. Daynes, "Moments of angst: A critical incident approach to designing customer-experience value systems," Benchmarking, vol. 21, no. 3, pp. 450-480, 2014. http://dx.doi.org/10.1108/BIJ-09-2012-0059

[16] J. Chepur and R. Bellamkonda, "Examining the conceptualizations of customer experience as a construct," Academy of Marketing Studies Journal, vol. 23, no. 1, pp. 1-9, 2019.

[17] R. T. R. Varma, "Enhancing and empowering: Customer experience," SCMS Journal of Indian Management, vol. 9, no. 3, pp. 71-78, 2012.

[18] J. Machala and D. Havíř, "You cannot do it yourself: Enhancing experience through co-creation," Trendy Ekonomiky a Managementu, vol. 13, no. 33, pp. 51-57, 2019. http://dx.doi.org/10.13164/trends.2019.33.51

[19] M. Stone, "The death of personal service: Will financial services customers who serve themselves do better than if they are served?," Journal of Database Marketing & Customer Strategy Management, vol. 19, no. 2, pp. 107-119, 2012. https://doi.org/10.1057/dbm.2012.8

[20] M. Rishi, A. Singh, and R. Shukla, "Confluence of technology and commercial factors at iskcon temple: Reflections on customer experience," Worldwide Hospitality and Tourism Themes, vol. 2, no. 5, pp. 539-553, 2010. https://doi.org/10.1108/17554211011090148

[21] K. N. Lemon and P. C. Verhoef, "Understanding customer experience throughout the customer journey," Journal of Marketing, vol. 80, no. 6, pp. 69-96, 2016.

[22] B. J. Pine and J. H. Gilmore, "Welcome to the experience economy," Harvard Business Review, vol. 76, pp. 97-105, 1998.

[23] R. Jain, J. Aagja, and S. Bagdare, "Customer experience–a review and research agenda," Journal of Service Theory and Practice, vol. 27, no. 3, pp. 642-662, 2017.

[24] C. Homburg, D. Jozić, and C. Kuehnl, "Customer experience management: Toward implementing an evolving marketing concept," Journal of the Academy of Marketing Science, vol. 45, no. 3, pp. 377-401, 2017.

[25] L. Du Plessis and M. De Vries, "Towards a holistic customer experience management framework for enterprises," South African Journal of Industrial Engineering, vol. 27, no. 3, pp. 23-36, 2016. 10.7166/27-3-1624

[26] S. Rahimian, M. ShamiZanjani, A. Manian, and E. Mohammad Rahim, "A framework of customer experience management for hotel industry," International Journal of Contemporary Hospitality Management, vol. 33, no. 5, pp. 1413-1436, 2021. https://doi.org/10.1108/IJCHM-06-2020-0522

[27] V. Popa and M. Barna, "Customer and shopper experience management," Valahian Journal of Economic Studies, vol. 4, no. 2, pp. 81-88, 2013.

[28] R. Johnston and X. Kong, "The customer experience: A road-map for improvement," Managing Service Quality: An International Journal, vol. 21, no. 1, pp. 5-24, 2011.

[29] H. Yen-Hao and S.-T. Yuan, "An application of technology-based design for exhibition services," International Journal of Quality and Service Sciences, vol. 8, no. 4, pp. 498-515, 2016. https://doi.org/10.1108/IJQSS-01-2016-0004

[30] M. M. Batra, "Designing a holistic customer experience program," Competition Forum, vol. 16, no. 1, pp. 73-81, 2018.

[31] M. M. Batra, "Customer experience: Trends, challenges, and managerial issues," Journal of Competitiveness Studies, vol. 27, no. 2, pp. 138-151, 2019.

[32] D. Siriguppi and J. Nair, "Tata aia life leverages digital technologies to create a superior customer experience," IUP Journal of Marketing Management, vol. 20, no. 4, pp. 400-411, 2021.

[33] F. V. Ordenes, B. Theodoulidis, J. Burton, T. Gruber, and M. Zaki, "Analyzing customer experience feedback using text mining: A linguistics-based approach," Journal of Service Research, vol. 17, no. 3, pp. 278-295, 2014.

[34] M. M. Batra, "Strengthening customer experience through artificial intelligence: An upcoming trend," Competition Forum, vol. 17, no. 2, pp. 223-231, 2019.

[35] I. R. Hodgkinson, T. W. Jackson, and A. A. West, "Customer experience management: Asking the right questions," The Journal of Business Strategy, vol. 43, no. 2, pp. 105-114, 2022. https://doi.org/10.1108/JBS-07-2020-0158

[36] H. O. S. Jorge, G. H. S. Mendes, P. A. Cauchick Miguel, M. Amorim, and T. Jorge Grenha, "Customer experience research: Intellectual structure and future research opportunities," Journal of Service Theory and Practice, vol. 31, no. 6, pp. 893-931, 2021. https://doi.org/10.1108/JSTP-08-2020-0193

[37] B. Neuhofer, B. Magnus, and K. Celuch, "The impact of artificial intelligence on event experiences: A scenario technique approach," Electronic Markets, vol. 31, no. 3, pp. 601-617, 2021. https://doi.org/10.1007/s12525-020-00433-4

[38] I. Anica-Popa, L. Anica-Popa, C. Rădulescu, and M. Vrîncianu, "The integration of artificial intelligence in retail: Benefits, challenges and a dedicated conceptual framework," Amfiteatru Economic, vol. 23, no. 56, pp. 120-136, 2021. https://doi.org/10.24818/EA/2021/56/120

[39] S. Bharwani and D. Mathews, "Techno-business strategies for enhancing guest experience in luxury hotels: A managerial perspective," Worldwide Hospitality and Tourism Themes, vol. 13, no. 2, pp. 168-185, 2021. https://doi.org/10.1108/WHATT-09-2020-0121

[40] S. Moore, S. Bulmer, and J. Elms, "The social significance of ai in retail on customer experience and shopping practices," Journal of Retailing & Consumer Services, Article vol. 64, pp. 1-8, 2022. 10.1016/j.jretconser.2021.102755

[41] M. Tiutiu, "Technologies that facilitate direct interactions with customers," Revista Economică, Article vol. 75, no. 3, pp. 89-94, 2023. 10.56043/reveco-2023-0030

[42] A. Pappas, E. Fumagalli, M. Rouziou, and W. Bolander, "More than machines: The role of the future retail salesperson in enhancing the customer experience," Journal of Retailing, Article vol. 99, no. 4, pp. 518-531, 2023. 10.1016/j.jretai.2023.10.004

[43] S. Robinson, C. Orsingher, L. Alkire, A. De Keyser, M. Giebelhausen, K. N. Papamichail *et al.*, "Frontline encounters of the ai kind: An evolved service encounter framework," Journal of Business Research, Article vol. 116, pp. 366-376, 2020. 10.1016/j.jbusres.2019.08.038

[44] S. Puntoni, R. W. Reczek, M. Giesler, and S. Botti, "Consumers and artificial intelligence: An experiential perspective," Journal of Marketing, Article vol. 85, no. 1, pp. 131-151, 2021. 10.1177/0022242920953847

[45] Y. Qian, J. Lu, Y. Miao, W. Ji, R. Jin, and E. Song, "Aiem: Ai-enabled affective experience management," Future Generation Computer Systems, Article vol. 89, pp. 438-445, 2018. 10.1016/j.future.2018.06.044

[46] M. Dini, S. Splendiani, L. Bravi, and P. Tonino, "In-store technologies to improve customer experience and interaction: An exploratory investigation in italian travel agencies," TQM Journal, vol. 34, no. 7, pp. 94-114, 2022. https://doi.org/10.1108/TQM-08-2021-0230

[47] H. Tseng-Lung, S. Mathews, and C. Y. Chou, "Enhancing online rapport experience via augmented reality," The Journal of Services Marketing, vol. 33, no. 7, pp. 851-865, 2019. https://doi.org/10.1108/JSM-12-2018-0366

[48] R. Chen, P. Perry, R. Boardman, and H. McCormick, "Augmented reality in retail: A systematic review of research foci and future research agenda," International Journal of Retail & Distribution Management, vol. 50, no. 4, pp. 498-518, 2022. https://doi.org/10.1108/IJRDM-11-2020-0472

[49] T. Hilken, J. Heller, M. Chylinski, D. I. Keeling, D. Mahr, and R. Ko de, "Making omnichannel an augmented reality: The current and future state of the art: An international journal," Journal of Research in Interactive Marketing, vol. 12, no. 4, pp. 509-523, 2018. https://doi.org/10.1108/JRIM-01-2018-0023

[50] B. Romano, S. Sands, and J. I. Pallant, "Virtual shopping: Segmenting consumer attitudes towards augmented reality as a shopping tool," International Journal of Retail & Distribution Management, Article vol. 50, no. 10, pp. 1221-1237, 2022. 10.1108/IJRDM-10-2021-0493

[51] W. D. Hoyer, M. Kroschke, B. Schmitt, K. Kraume, and V. Shankar, "Transforming the customer experience through new technologies," Journal of Interactive Marketing, Article vol. 51, pp. 57-71, 2020. 10.1016/j.intmar.2020.04.001

[52] C. Flavián, S. Ibáñez-Sánchez, and C. Orús, "The impact of virtual, augmented and mixed reality technologies on the customer experience," Journal of Business Research, Article vol. 100, pp. 547-560, 2019. 10.1016/j.jbusres.2018.10.050

[53] D. Plotkina, J. Dinsmore, and M. Racat, "Improving service brand personality with augmented reality marketing," Journal of Services Marketing, Article vol. 36, no. 6, pp. 781-799, 2022. 10.1108/JSM-12-2020-0519

[54] S. B. Qadri, M. M. Mir, and M. A. Khan, "Exploring the impact of augmented reality on customer experiences and

attitudes: A comparative analysis with websites," International Journal of Management Research & Emerging Science (IJMRES), Article vol. 13, no. 2, pp. 168-192, 2023. 10.56536/ijmres.v13i2.421

[55] S. Cacho-Elizondo, J.-D. L. Álvarez, and V.-E. Garcia, "Exploring the adoption of augmented and virtual reality in the design of customer experiences: Proposal of a conceptual framework," Journal of Marketing Trends (1961-7798), Article vol. 5, no. 2, pp. 91-102, 2018.

[56] T.-L. Huang, S. Mathews, and C. Y. Chou, "Enhancing online rapport experience via augmented reality," Journal of Services Marketing, Article vol. 33, no. 7, pp. 851-865, 2019. 10.1108/JSM-12-2018-0366

[57] S. Wolpert and A. Roth, "Development of a classification framework for technology based retail services: A retailers' perspective," International Review of Retail, Distribution & Consumer Research, Article vol. 30, no. 5, pp. 498-538, 2020. 10.1080/09593969.2020.1768575

[58] N. Vaidyanathan and S. Henningsson, "Designing augmented reality services for enhanced customer experiences in retail," Journal of Service Management, Article vol. 34, no. 1, pp. 78-99, 2023. 10.1108/JOSM-01-2022-0004

[59] A. Hall and C. Wright, "Augmented reality in business classes," Journal of Research in Business Education, Article pp. 4-11, 2020.

[60] M. Kozina and D. Dusper, "Adopting best practices to improve customer experience management," 62nd International Scientific Conference on Economic and Social Development, pp. 288-295, 2020.

[61] S. S. Liu and J. Chen, "Using data mining to segment healthcare markets from patients' preference perspectives," International Journal of Health Care Quality Assurance, vol. 22, no. 2, pp. 117-34, 2009. https://doi.org/10.1108/09526860910944610

[62] H. Miyamoto, "Narita airport's journey for establishing an end-to-end biometric passenger experience," Journal of Airport Management, Article vol. 17, no. 1, pp. 27-43, 2022.

[63] J. Dawes and J. Rowley, "Enhancing the customer experience: Contributions from information technology," Management Decision, vol. 36, no. 5, pp. 350-357, 1998. https://doi.org/10.1108/00251749810220568

[64] C. Ja-Shen, Y. L. Tran-Thien, and F. Devina, "Usability and responsiveness of artificial intelligence chatbot on online customer experience in e-retailing," International Journal of Retail & Distribution Management, vol. 49, no. 11, pp. 1512-1531, 2021. https://doi.org/10.1108/IJRDM-08-2020-0312

[65] K. Sidaoui, M. Jaakkola, and J. Burton, "Ai feel you: Customer experience assessment via chatbot interviews," Journal of Service Management, vol. 31, no. 4, pp. 745-766, 2020. https://doi.org/10.1108/JOSM-11-2019-0341

[66] I. Lubbe and N. Ngoma, "Useful chatbot experience provides technological satisfaction: An emerging market perspective," South African Journal of Information Management, vol. 23, no. 1, pp. 1-8, 2021. https://doi.org/10.4102/sajim.v23i1.1299

[67] A. Abdulquadri, E. Mogaji, T. A. Kieu, and N. Nguyen Phong, "Digital transformation in financial services provision: A nigerian perspective to the adoption of chatbot," Journal of Enterprising Communities, vol. 15, no. 2, pp. 258-281, 2021. https://doi.org/10.1108/JEC-06-2020-0126

[68] J.-S. Chen, T.-T.-Y. Le, and D. Florence, "Usability and responsiveness of artificial intelligence chatbot on online customer experience in e-retailing," International Journal of Retail & Distribution Management, Article vol. 49, no. 11, pp. 1512-1531, 2021. 10.1108/IJRDM-08-2020-0312

[69] H. Shin, I. Bunosso, and L. R. Levine, "The influence of chatbot humour on consumer evaluations of services," International Journal of Consumer Studies, Article vol. 47, no. 2, pp. 545-562, 2023. 10.1111/ijcs.12849

[70] J. Trivedi, "Examining the customer experience of using banking chatbots and its impact on brand love: The moderating role of perceived risk," Journal of Internet Commerce, Article vol. 18, no. 1, pp. 91-111, 2019. 10.1080/15332861.2019.1567188

[71] C. Bălan, "Chatbots and voice assistants: Digital transformers of the company–customer interface—a systematic review of the business research literature," Journal of Theoretical & Applied Electronic Commerce Research, Article vol. 18, no. 2, pp. 995-1019, 2023. 10.3390/jtaer18020051

[72] S. Fosso Wamba, C. Guthrie, M. M. Queiroz, and S. Minner, "Chatgpt and generative artificial intelligence: An exploratory study of key benefits and challenges in operations and supply chain management," International Journal of Production Research, Article pp. 1-21, 2023. 10.1080/00207543.2023.2294116

[73] H. Hsu, "Facing the era of smartness – delivering excellent smart hospitality experiences through cloud computing," Journal of Hospitality Marketing & Management, Article pp. 1-27, 2023. 10.1080/19368623.2023.2251144

[74] P. Kumar, A. K. Mokha, and S. C. Pattnaik, "Electronic customer relationship management (e-crm), customer experience and customer satisfaction: Evidence from the banking industry," Benchmarking, vol. 29, no. 2, pp. 551-572, 2022. https://doi.org/10.1108/BIJ-10-2020-0528

[75] J. Teixeira, L. Patrício, N. J. Nunes, L. Nóbrega, R. P. Fisk, and L. Constantine, "Customer experience modeling: From customer experience to service design," Journal of Service Management, vol. 23, no. 3, pp. 362-376, 2012. https://doi.org/10.1108/09564231211248453

[76] G. Timokhina, L. Prokopova, Y. Gribanov, S. Zaitsev, N. Ivashkova, R. Sidorchuk *et al.*, "Digital customer experience mapping in russian premium banking," Economies, vol. 9, no. 3, pp. 1-24, 2021. https://doi.org/10.3390/economies9030108

[77] A. Ludwiczak, "Using customer journey mapping to improve public services: A critical analysis of the literature," Management, vol. 25, no. 2, pp. 22-35, 2021. https://doi.org/10.2478/manment-2019-0071

[78] A. Lao, M. Vlad, and A. Martin, "Exploring how digital kiosk customer experience enhances shopping value, self-mental imagery and behavioral responses," International Journal of Retail & Distribution Management, vol. 49, no. 7, pp. 817-845, 2021. https://doi.org/10.1108/IJRDM-09-2020-0357

[79] Y. Vakulenko, D. Hellström, and P. Oghazi, "Customer value in self-service kiosks: A systematic literature review," International Journal of Retail & Distribution Management, vol. 46, no. 5, pp. 507-527, 2018. https://doi.org/10.1108/IJRDM-04-2017-0084

[80] D. Grewal, S. Benoit, S. M. Noble, A. Guha, C.-P. Ahlbom, and J. Nordfält, "Leveraging in-store technology and ai: Increasing customer and employee efficiency and enhancing their experiences," Journal of Retailing, Article vol. 99, no. 4, pp. 487-504, 2023. 10.1016/j.jretai.2023.10.002

[81] R. N. Bolton, J. R. McColl-Kennedy, L. Cheung, A. Gallan, C. Orsingher, L. Witell, and M. Zaki, "Customer experience challenges: Bringing together digital, physical and social

realms," Journal of Service Management, vol. 29, no. 5, pp. 776-808, 2018. http://dx.doi.org/10.1108/JOSM-04-2018-0113

[82] C. Märtin, B. C. Bissinger, and P. Asta, "Optimizing the digital customer journey—improving user experience by exploiting emotions, personas and situations for individualized user interface adaptations," Journal of Consumer Behaviour, Article vol. 22, no. 5, pp. 1050-1061, 2023. 10.1002/cb.1964

[83] D. Grewal, M. Kroschke, M. Mende, A. L. Roggeveen, and M. L. Scott, "Frontline cyborgs at your service: How human enhancement technologies affect customer experiences in retail, sales, and service settings," Journal of Interactive Marketing, Article vol. 51, pp. 9-25, 2020. 10.1016/j.intmar.2020.03.001

[84] M. C. Tom Dieck and D.-i. D. Han, "The role of immersive technology in customer experience management," Journal of Marketing Theory & Practice, Article vol. 30, no. 1, pp. 108-119, 2022. 10.1080/10696679.2021.1891939

[85] Z. Sander Paul and A. Farhoomand, "The hong kong jockey club: Transforming customer experience through information technology," Communications of the Association for Information Systems, vol. 34, pp. 1115-1132, 2014. https://doi.org/10.17705/1CAIS.03458

[86] C. Gellweiler and L. Krishnamurthi, "Editorial: How digital innovators achieve customer value," Journal of Theoretical and Applied Electronic Commerce Research, vol. 15, no. 1, pp. I-VIII, 2020. https://doi.org/10.4067/S0718-18762020000100101

[87] L. Wolf, "Device ‐ mediated customer behaviour on the internet: A systematic literature review," International Journal of Consumer Studies, Article vol. 47, no. 6, pp. 2270-2304, 2023. 10.1111/ijcs.12925

[88] A. Bonfanti, V. Vigolo, V. Vannucci, and F. Brunetti, "Creating memorable shopping experiences to meet phygital customers' needs: Evidence from sporting goods stores," International Journal of Retail & Distribution Management, Article vol. 51, no. 13, pp. 81-100, 2023. 10.1108/IJRDM-12-2021-0588

[89] M. S. Balaji and S. K. Roy, "Value co-creation with internet of things technology in the retail industry," Journal of Marketing Management, Article vol. 33, no. 1/2, pp. 7-31, 2017. 10.1080/0267257X.2016.1217914

[90] J. M. Matyus, B. T. Mergenthal, Z. Gonzalez, and T. Bilodeau, "Determinants to consumer's shopping preferences," Journal of Behavioral & Applied Management, Article vol. 23, no. 3, pp. 117-124, 2023. 10.21818/001c.90583

[91] D. Buhalis, M. S. Lin, and D. Leung, "Metaverse as a driver for customer experience and value co-creation: Implications for hospitality and tourism management and marketing," International Journal of Contemporary Hospitality Management, Article vol. 35, no. 2, pp. 701-716, 2023. 10.1108/IJCHM-05-2022-0631

[92] V. Vitezic, T. Car, and M. Simunic, "Managing innovative technology in the hotel industry - response to growing consumer preferences," Tourism in Southern and Eastern Europe, vol. 3, pp. 467-478, 2015.

[93] R. Tateson and E. Bonsma, "Shoppinggarden -- improving the customer experience with on-line catalogues," BT Technology Journal, vol. 21, no. 4, pp. 84-91, 2003.

[94] H. Schallehn, S. Seuring, J. Strähle, and M. Freise, "Customer experience creation for after-use products: A product–service systems-based review," Journal of Cleaner Production, Article vol. 210, pp. 929-944, 2019. 10.1016/j.jclepro.2018.10.292

[95] S. Barat, "Rfid – improving the customer experience," The Journal of Consumer Marketing, vol. 28, no. 4, pp. 316-317, 2011. https://doi.org/10.1108/07363761111143501

[96] K. N. Kumar and P. R. Balaramachandran, "Robotic process automation - a study of the impact on customer experience in retail banking industry," Journal of Internet Banking and Commerce, vol. 23, no. 3, pp. 1-27, 2018.

[97] D. Kedziora and H.-M. Kiviranta, "Digital business value creation with robotic process automation (rpa) in northern and central europe," Management (18544223), Article vol. 13, no. 2, pp. 161-174, 2018. 10.26493/1854-4231.13.161-174

[98] B. Pistrui, D. Kostyal, and Z. Matyusz, "Dynamic acceleration: Service robots in retail," Cogent Business & Management, Article vol. 10, no. 3, pp. 1-20, 2023. 10.1080/23311975.2023.2289204

[99] M. Åkesson, B. Edvardsson, and B. Tronvoll, "Customer experience from a self-service system perspective," Journal of Service Management, vol. 25, no. 5, pp. 677-698, 2014. http://dx.doi.org/10.1108/JOSM-01-2013-0016

[100] T. Chellapalli, "Customer experience of banking self-service technologies in india′ s an empirical study," Academy of Marketing Studies Journal, vol. 27, no. 3, pp. 666-675, 2023.

[101] C. Lu, W. Geng, and I. Wang, "The role of self-service mobile technologies in the creation of customer travel experiences," Technology Innovation Management Review, vol. 5, no. 2, pp. 24-32, 2015.

[102] M. Stone, "The death of personal service: Why retailers make consumers responsible for their own customer experience," Journal of Database Marketing & Customer Strategy Management, vol. 18, no. 4, pp. 233-239, 2011. http://dx.doi.org/10.1057/dbm.2011.29

[103] H. Shin and B. Dai, "The efficacy of customer's voluntary use of self-service technology (sst): A dual-study approach," Journal of Strategic Marketing, Article vol. 30, no. 8, pp. 723-745, 2022. 10.1080/0965254X.2020.1841269

[104] C. Liu and K. Hung, "Improved or decreased? Customer experience with self-service technology versus human service in hotels in china," Journal of Hospitality Marketing & Management, Article vol. 31, no. 2, pp. 176-204, 2022. 10.1080/19368623.2021.1941475

[105] C. C. Ugwuanyi and E. C. Idoko, "Effects of self-service technologies' attributes on bank customers' experience, relationship quality and re-use intention: Insights from a developing economy," Vision, Article pp. 1-15, 2022. 10.1177/09722629221110035

[106] A. Ferreira, G. M. Silva, and Á. L. Dias, "Determinants of continuance intention to use mobile self-scanning applications in retail," International Journal of Quality & Reliability Management, Article vol. 40, no. 2, pp. 455-477, 2023. 10.1108/IJQRM-02-2021-0032

[107] L. Gonçalves, L. Patrício, T. Jorge Grenha, and N. V. Wünderlich, "Understanding the customer experience with smart services," Journal of Service Management, vol. 31, no. 4, pp. 723-744, 2020. https://doi.org/10.1108/JOSM-11-2019-0349

[108] S. Kabadayi, F. Ali, H. Choi, H. Joosten, and C. Lu, "Smart service experience in hospitality and tourism services: A conceptualization and future research agenda," Journal of Service Management, vol. 30, no. 3, pp. 326-348, 2019. https://doi.org/10.1108/JOSM-11-2018-0377

[109] V. Chang, L. M. T. Doan, Q. Ariel Xu, K. Hall, Y. Anna Wang, and M. Mustafa Kamal, "Digitalization in omnichannel healthcare supply chain businesses: The role of smart wearable devices," Journal of Business Research, Article vol. 156, pp. 1-20, 2023. 10.1016/j.jbusres.2022.113369

[110] S. K. Tewari, "Information technology: A tool to drive customer experience," Academy of Marketing Studies Journal, suppl. Special Issue 1, vol. 26, pp. 1-14, 2022.

[111] U. Ramanathan, N. Subramanian, and G. Parrott, "Role of social media in retail network operations and marketing to enhance customer satisfaction," International Journal of Operations & Production Management, vol. 37, no. 1, pp. 105-123, 2017. http://dx.doi.org/10.1108/IJOPM-03-2015-0153

[112] Z. Tingting, J. Kandampully, and A. Bilgihan, "Motivations for customer engagement in online co-innovation communities (occs)," Journal of Hospitality and Tourism Technology, vol. 6, no. 3, pp. 311-328, 2015. https://doi.org/10.1108/JHTT-10-2014-0062

[113] R. Jain and S. Bagdare, "Determinants of customer experience in new format retail stores," Journal of Marketing & Communication, vol. 5, no. 2, pp. 34-44, 2009.

[114] P. Echeverri, "Video-based methodology: Capturing real-time perceptions of customer processes," International Journal of Service Industry Management, vol. 16, no. 2, pp. 199-209, 2005. https://doi.org/10.1108/09564230510592315

[115] M. Simoni, A. Sorrentino, D. Leone, and A. Caporuscio, "Boosting the pre-purchase experience through virtual reality. Insights from the cruise industry," Journal of Hospitality and Tourism Technology, vol. 13, no. 1, pp. 140-156, 2022. https://doi.org/10.1108/JHTT-09-2020-0243

[116] G. Pleyers and I. Poncin, "Non-immersive virtual reality technologies in real estate: How customer experience drives attitudes toward properties and the service provider," Journal of Retailing & Consumer Services, Article vol. 57, pp. 1-9, 2020. 10.1016/j.jretconser.2020.102175

[117] T. Stevens and A. May, "Improving customer experience using web services," BT Technology Journal, vol. 22, no. 1, pp. 63-71, 2004. http://dx.doi.org/10.1023/B:BTTJ.0000015496.47894.cb

[118] J. J. Turner and A. Szymkowiak, "An analysis into early customer experiences of self-service checkouts: Lessons for improved usability," Engineering Management in Production & Services, Article vol. 11, no. 1, pp. 36-50, 2019. 10.2478/emj-2019-0003

Capability and Applicability of Measuring AI Model's Environmental Impact

Rui Zhou, Tao Zheng, Xin Wang, Lan Wang
Orange Innovation China
Beijing, China
e-mail: {rui.zhou, tao.zheng, xin2.wang, lan.wang}@orange.com

Emilie Sirvent-Hien, Nathalie Charbonniaud
Orange Innovation
Châtillon, France
e-mail: {emilie.hien, nathalie.charbonniaud}@orange.com

Abstract - **More and more of Artificial Intelligence (AI) systems have been adopted by Information and Communication Technology (ICT) solutions to make effective digital transformation. In recent years environmental impact of AI systems has been investigated and methodologies have been developed to calculate their cost. In this paper, we survey, analyze, and evaluate three types of tools for counting the energy consumption/CO2 emission (CO2e) of AI systems. By verifying them in sets of experiments, including centralized and distributed on devices architecture, we compare ease of use of tools, simulation result vs real measurement and finally bring advice to help AI developers to take into account environmental cost of AI models with measurement tools. Finally, we developed a measurement tool for AI model environment impact based-on our experiments on the power consumption of AI models and applied our tool on AI model to verify optimization results.**

Keywords - AI Environmental Impact; CO2e; FLoating point of OPerations (FLOPs); Power Usage Effectiveness (PUE); Pragmatic Scaling Factor (PSF); Thermal Design Power (TDP); Multiple Object Tracking Accuracy (MOTA).

I. INTRODUCTION

This paper extends our earlier work [1] presented at The Fourteenth International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking (COMPUTATION TOOLS 2023).

The global average temperature in the past decades has increased more than 1°C compared to the pre-industrial baseline (1850-1900) [2]. Such climate change has caused more extreme weather events, rising seas, reduction of biodiversity, and negative impact to global health and safety. The global warming is a critical issue facing all mankind. Paris agreement sets a global objective for the temperature increase below 2°C. Many nations, regions, industries, companies, and individuals have put in place climate actions on their agendas. The current rise is more rapid primarily as the result of greenhouse gas emissions by burning fossil fuels for energy used in industry, transport, building, etc., which took 73.2% of global greenhouse gas emissions according to the data obtained in the year 2016 [3].

Using Communication Technology (ICT) solutions in these sectors can have a calculated potential to reduce greenhouse gas emissions by up to 15% [4]. However, their own contribution to greenhouse gas emissions should not be ignored, for example, it accounted for around 700 MtCO2e in 2020, equivalent to around 1.4% of global GHG emissions [5]. Our focus is on Artificial Intelligence (AI) as more and more of them have been adopted by ICT solutions to make the effective digital transformation. It is expected that the Artificial Intelligence industry will be worth $190 billion by 2025, with global spending in AI systems reaching $57 billion by 2021 already [6]. The demand for computing these AI systems is growing exponentially. The intensive computation nowadays not only takes place in datacenters but also in a huge amount of edge devices closed to consumers and enterprises to support AI applications to process big data with low latency and large bandwidth requirements.

Scientists and researchers have started to investigate the environmental impact of AI systems in recent years and have developed methodologies to calculate their costs. For example, Schwartz and Doge et al. [7] refer to the AI systems that focus on accuracy without any estimation on the economic, environmental, or social cost of reaching the claimed accuracy as Red AI. They have proposed a simplified estimation of the cost of an AI which grows linearly with the cost of processing a single example, the size of the training dataset, and the number of hyperparameter experiments. OpenAI [8] has pointed out that among the three factors driving the advance of AI: algorithmic innovation, data, and the amount of computing available for training, computing is unusually quantifiable. The number of FLOPs (adds and multiplies) in the described architecture per training example can be counted. If there is not enough information to directly count FLOPs, Graphics Processing Unit (GPU) training time, how many GPUs used and a reasonable guess at GPU utilization can be used to estimate the number of operations performed. Strubell et al. [9] have quantified the computational and environmental cost of training several popular Natural Language Processing (NLP) models. The total power required at a given instance during training is related to the average PUE for datacenter multiplying the sum of average power draw from all Central Processing Unit (CPU) sockets, average power draw from all Dynamic Random Access Memory (DRAM) (main memory) sockets, and average power draw of a GPU during training multiplied by the number of GPU. The greenhouse gas emission equivalent per kilowatt-hour is then calculated based on data provided by the U.S. Environmental Protection Agency (EPA). Google research team R. So et al. [10] have evaluated Large Transformer models, which have been central to recent advances in NLP and have developed a more efficient variant with a smaller training cost than the

original transformer and other variants for auto-regressive language modelling. Patterson et al. [11] have calculated the energy use and carbon footprint of several recent large models and found that large but sparsely activated Deep Neural Networks (DNNs) can consume less energy than the large, dense DNNs without sacrificing accuracy despite using as many or even more parameters. The geographic location and specific data center infrastructure matters to reduce the greenhouse gas emission equivalent of Machine Learning (ML) workload. Patterson et al. [12] have shared interesting information that the inference represents about 3/5 of total ML usage at Google across three years, due to the many billion-user services that use ML. The combined emissions of training and serving need to be minimized. Lacoste et al. [13] have developed a tool called "Machine Learning Emissions Calculator" for ML community to approximate the environmental impact of training ML models. Ligozat and Luccioni [14] have proposed a practical guide to quantifying carbon emissions for ML researchers and practitioners. To analyse the carbon impact of ML, besides the ML model emissions of greenhouse gas due to the power consumption incurred by the equipment at the running time, other dimensions of model impact should be considered such as model preparation overhead at deployment stage, static consumption of the equipment, infrastructure, as well as the overall life cycle analysis of the equipment. The authors also have suggested the most important steps to take for practitioners and institutions for example, as an institution, deploying computation in low-carbon regions, providing institutional tools for tracking emissions, capping computational usage, carrying out awareness campaigns, and facilitating institutional offsets. Recent work has also tried to apply the life cycle analysis of the entire ML development and deployment cycle and consider the complexity involved in deploying, scaling, and maintaining ML models in practice and in real-time [15].

Standardizations have begun to tackle the subject, and for example, a new work item proposal has been under discussion in the Joint Technical Committee on Artificial Intelligence (JTC 21) of European Committee for Standardization (CEN) and European Committee for Electrotechnical Standardization (CENELEC). The new work item is about "Green and sustainable AI", which will establish a framework for quantification of the environmental impact of AI and its long-term sustainability and encourage AI developers and users to improve the efficiency of AI use [16]. The "CEN/CENELEC standardization landscape for energy management and environmental viability of green datacenters [17]" defines Key Performance Indicators (KPIs) that address energy and environmental control. However, these KPIs are focused on datacenters and currently do not address the distributed or IoT energy and environmental control. In Dec. 2023, AFNOR(French Standardization Association) published ISO/IEC 42001, which specifies the requirements and provides guidance for establishing, implementing, maintaining and continually improving AI management system within the context of an organization. This document is intended to help the organization develop, provide, or use AI systems responsibly [18]. And more frugal AI initiatives are advocated.

As discussed before, quantifying greenhouse emissions for any AI system is very important and several simplified and applicable methods have been developed in recent studies. Some open-source tools are available applying and integrating these methods. These tools have been classified into three major categories: priori measurement tools, which usually calculate operational points in training and inference; on-the-fly measurement tools, which measure power consumption, etc., when an AI system is running on hardware; posteriori measurement tools refer to these tools to approximate greenhouse gas emissions for a given computation. In our experiments, we deep dive into PowerAPI [19], PyJoules [20], and other open-source tools, such as Keras-flops [21], Torchstat [22], torchsummaryX [23], Flops-counter [24], JouleHunter [25], Jtop [26], CarbonAI [27], MLCO2 [13] and Green Algorithm [28], in order to have first-hand experience and to understand their capabilities and limitations.

Most of the recent research work focuses on the environmental impact of ML models at the training stage. As [10] mentioned, the energy consumed at the inference stage was more than the energy consumed at training for a given few years. So our idea is to set up a framework to evaluate the AI systems at both the training and inference stages. Considering large scale of AI systems is running on the edge side, we set up a heterogenous edge platform with various device types where we can use the proper orchestration tool that we have evaluated to deploy ML model on these different edge devices, which somehow can simulate distributed AI applications deployed in real scenarios. Both X86-based and ARM-based hardware are used in the platform. We have designed methodologies to perform sets of experiments to measure the power consumption of the ML model incurred on hardware for the training stage and inference stage. Unlike a data center, the power consumed by these edge devices is mainly coming from CPU/GPU computation and memory usage with very limited overhead for cooling components if they have any. Even so, we have measured the static power consumption of edge devices to get a more precise measurement of AI model power consumption by subtracting the static power consumption. We also select various types of ML models for one AI use case and compare the power consumption of these ML models when they are running on edge devices in addition to their performance.

Our objectives for the studies are to verify the measurement tools and improve them; to obtain greenhouse gas emissions of various ML models; to benchmark performance vs environmental impact; and to develop greener ML models. These experimental results can provide useful information and recommendations for organizations to build an institutional toolbox to track greenhouse gas emissions of AI systems and offer responsible AI applications to our customers. The scope and key point of our analysis are the comparison of training and inference energy/CO2 consumptions among different AI models solving the same problem, not aiming to compare their

environmental impacts when running on centralized server and running on edge devices.

Green House Gas (GHG) Protocol is an international standard for corporate accounting and reporting emissions, categorizing greenhouse gasses into Scope 1, 2 and 3 based on the source [29]. This paper mainly focuses on scope 2.

The organization of this paper is as follows. Sections II to IV analyze the three categories of measurement tools respectively; Section V provides our experiments and results analysis, Section VI introduces our measurement tool, and Section VII gives the conclusion.

II. PRIORI MEASUREMENT TOOLS ANALYSIS

To evaluate the ML model's power consumption by comparing the model's calculation amount, the priori measurement tools are employed. They are used to evaluate AI models and algorithms through computing the flops/mult-adds/other parameters.

There are two usages of priori measurement tools:

- as an inline module to measure the AI model, for example, first install as a python module, and then call some tools' functions in the model source program to get the model's related computing information.
- as a Command Line Interface (CLI) tool to handle the model's source program, for example, first install the tool, and execute the tool to process the model source program to get the model's related computing information.

Because priori measurement tools handle the source code of AI programs, they always process one specific framework and support a subset of types of layers.

For testing priori measurement tools, a test environment was built, and related AI frameworks and some candidate priori measurement tools were installed first; then, constructed some demo AI models with/without special layers based on relevant frameworks; finally, computed and compared these demo AI models' Flops/Mult-Adds and other related measurements using candidate priori measurement tools and evaluated them through these computed results.

We launched five tests for four priori measurement tools: keras-flops, torchstat, torchsummaryX and flops-counter.

Keras-flops as a python module can calculate the FLOPs of neural network architecture written in Tensorflow. Test1 verifies keras-flops' support for Conv2dTranspose layer. In this test, we constructed a model including Conv2dTranspose layer to test this tool's capability with two python programs (with/without Conv2dTranspose Layer). The difference value of flops means that Conv2dTranspose layer is supported by keras-flops. Test2 verifies keras-flops' support for Conv3dTranspose layer. According to the supporting table, Conv3dTranspose layer is not supported by keras-flops. In this test, we constructed a model including Conv3dTranspose layer and test the tool's capability with two python programs (with/without Conv3dTranspose layer). The same value of flops means that Conv3dTranspose layer is not supported by keras-flops.

TABLE I. SUMMARY OF FOUR PRIORI MEASUREMENT TOOLS

priori measurement tools	support framework	outputs
keras-flops	Tensorflow	FLOPs
torchsummaryX	PyTorch	FLOPs, Multi-Add, memory, total params
torchstat	PyTorch	Multi-Add, total params
flops-counter	PyTorch	Multi-Add, total params

Torchstat is a lightweight neural network analyzer based on PyTorch. Its usage is as a python module to measure an AI model or as a CLI tool to handle a python program including an AI model. Test3 verifies Torchstat's support for Conv2d layer and ConvTranspose2d layer. In this test, we constructed Convolutional Neural Network (CNN) models including Conv2d layer and ConvTranspose2d layer to test the tool's capability.

TorchsummaryX is also a tool based on the Pytorch framework. This tool can handle Recurrent Neural Network (RNN), Recursive Neural Network, or models with multiple inputs. In the test4, we constructed two models with Conv2d layer and ConvTranspose2d layer respectively. We can find Mult-Adds remains unchanged. It means that Convtranspose2d layer is supported for Mult-Adds by torchsummaryX.

Flops-counter is based on the PyTorch framework and designed to compute the theoretical number of multiply-add operations in CNNs. It can also compute the number of parameters and print the per-layer computational cost of a given network. In the test5, we constructed two models with Conv2d layer and ConvTranspose2d layer respectively. We can find the computational complexity (i.e., number of multiply-add operations) remains unchanged. It means that Convtranspose2d layer is supported for Mult-Adds by torchsummaryX.

Torchstat, torchsymmaryX, and flops-counter are all based on the PyTorch framework, but their outputs are different. Torchstat outputs numbers of parameters, amount of Multiply+Adds, number of flops, and memory usage. torchsummaryX and flops-counter just provide numbers of parameters and amount of multiply+adds. According to some feedback from github [30], the results of torchstat's MAdd and FLOPs are wrong, which should be swapped. The summary of the four tools is shown in the following Table I.

Through our five tests, we can find that the effectiveness of priori measurement tools relies on their detailed implementation. The application of priori measurement tools is limited. The tools we tested just support one special framework (TensorFlow or PyTorch) and a subset of types of model layers. In practice, most AI models usually include some specific layers (e.g., 3D ConvTranspose layer), which cannot be calculated by our tested priori measuring tools.

III. ON-THE-FLY MEASUREMENT TOOLS ANALYSIS

The most direct and precise way to measure an AI program's power consumption is to measure it in real-time while the process is going on. We named this type of tool the "on-the-fly tool". For this purpose, we have carried out the

research and study of relevant tools and later carried out the comparison test and verification of their real use situation.

After preliminary selection, we chose the following three measurement tools as candidates, they are PowerAPI series (JouleHunter, PyJoules), CarbonAI, CodeCarbon and Jtop. They have their own methods and application scenarios, and following their official instructions and guidance documents, we conducted a series of tests and applications on them.

The first one is the PowerAPI, the goal of this project is to provide a set of tools to go forward greener computing, the idea is to provide software-defined power meters to measure the power consumption of the program, the core of this project is the PowerAPI toolkit for building such power meters [19].

PowerAPI is a middleware toolkit for building software-defined power meters. Software-defined power meters are configurable software libraries that can estimate the power consumption of software in real-time. A power meter built on PowerAPI normally has two components -- the sensor and the formula. The sensor is also a software, which worked like the physical world sensor, queries the hardware's (host machine) data, and collects raw data correlated with the power consumption of the software. All data will be stored in an external database to make the data available to the formula. For the other component, the formula is a computational module that uses the collected data to determine power consumption. Both are connected by a database that is used to transfer information. The global architecture of a power meter is represented in Figure 1 below [31].

For convenience and quick use, PowerAPI has provided several useful components. As Hwps-Sensor (Hardware Performance Counter), is a tool using the Running Average Power Limit (RAPL) technology to monitor the Intel CPU performance counter and power consumption of the CPU. Also, some matched formulas like "SmartWatts Formula" used for physical Linux machine, "VirtualWatts" used for a virtue machine, etc.

PowerAPI also packages up (with sensor and formula) a set of ready-to-use tools for diverse needs. Here Joule Hunter and PyJoules are the two we selected and used for our research.

JouleHunter runs on Linux machines with Intel RAPL support. This technology has been available since the Sandy Bridge generation [32].

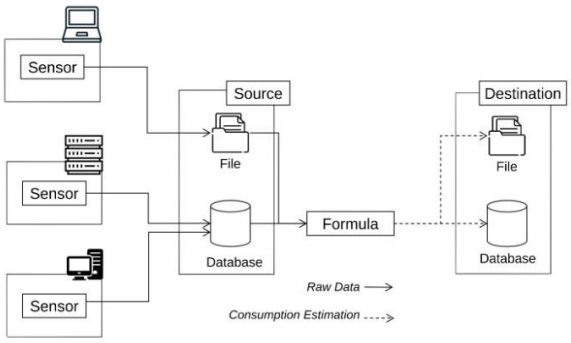


Figure 1. The global architecture of a power meter

JouleHunter can show what part of your program code is consuming considerable amounts of energy in detail. JouleHunter works similarly to pyinstrument [33], as it forked the repo and replaced time measuring with energy measuring. This tool can be easily installed and used with one or two command line(s), two key components of hardware the intel CPU and ram's power consumption can be printed out. However, from its official documentation and real test we can see that JouleHunter this tool has its limitation, such as it only worked with Linux OS and no calculation for GPU power consumption.

Another software toolkit from PowerAPI is the PyJoules, which can be used to measure the energy footprint of a host machine along with the execution of a piece of Python code. Except for Intel CPU socket package and RAM (only for Intel server architectures), it also can monitor the energy consumed by the GPU of the host machine, supporting both for Intel integrated GPU (for client architectures) and Nvidia GPU (Uses the Nvidia "Nvidia Management Library" technology to measure the power consumption of Nvidia devices. The energy measurement Application Programming Interface (API) is only available on Nvidia GPU with Volta architecture 2018) [34]. PyJoules can only work with AI program coding on Python, and it should be installed and imported as a function into the target main project python file. It will report the total power consumption during the code is running. Its results contain not only the target project's power consumption, thus including the OS and other applications running at the same time if have. That means it calculates the global power consumption of all the processes running on the machine during this period. With PyJoules, to get the closest measure to the real power consumption of the measured program, we need to try to eliminate any extra programs (such as graphical interface, background running task, etc.) that may alter the power consumption of the host machine and keep only the code under measurement. Same as JouleHunter, PyJoules currently can only work on GNU/Linux, and does not support on Windows and MacOS.

CodeCarbon and CarbonAI are two other projects, which aim to raise AI the developers' awareness of the AI's carbon footprint. The two have similar methodology and formula, while using different data sources (carbon intensity, hardware references, etc.). Firstly, like PyJoules they provide a python package that allows developer to monitor power consumption. Then based on the measurement results, CodeCarbon and CarbonAI will do a transition between power consumption and CO2 emissions, to provide a more intuitive understanding of how much our AI development is doing to the environment. For example, training an AI model for 100 rounds is the equivalent of driving from Paris to Marseille. Also, the power consumption results of CodeCarbon or CarbonAI are given as a CSV file, which includes most key devices of a host machine, like CPU, GPU, and RAM, but also the name of the country where the package was used (based on the IP or what the user set). And CodeCarbon also provides some information like Cloud provides and the region selected for a AI programs running on a Cloud server. So, the amount of CO2 emitted by the

usage will be depends on the country and the energy mix used by the country to produce electricity. For combability, a different form that PyJoules only support Linux OS, CodeCarbon and CarbonAI packages are compatible with most platforms (Linux, Windows, and MacOS) with the varying installation process.

At present, apart from x86 architecture servers and devices, ARM-based devices are also widely used in various fields of AI. However previous tree tools only work well on x86 hardware platforms, all of them will get several issues or bugs. For ARM platforms, Nvidia provides their official tool Jtop for the Jetson series, a platform designed for AI development and use cases. Jtop is one of jetson-stats, a package for monitoring and controlling NVIDIA Jetson (Xavier NX, Nano, AGX Xavier, TX1, TX2) Works with all NVIDIA Jetson ecosystems. Jtop can be run independently and show the real-time usage data of CPU, GPU, and RAM and also the actual frequency of the hardware. With its built-in graph user interface, we can easily read the results, but only the immediate frequency, so for the final power consumption we need manually calculate the Total power consumption using w=p*t, and "t" is the duration of the target AI program. Compared to other tools, although the result of power consumption cannot be directly obtained, Jtop can provide the usage rate of each device.

All those tools are not very difficult to install and use, some of them can be installed with several command lines, like JouleHunter, CodeCarbon, CarbonAI, and Jtop; and for PyJoules, we can add it into our application code just like a function. For compatibility, except CarbonAI supports Linux, Windows and MacOS (we only use it on Linux machines), other tools currently can only be used on Linux. Most tools currently only support Intel CPU and RAM, for GPU's power consumption, we need external components or 3rd part tools. For the programming language, most tools are built up as a python package, so they only worked with AI apps coded with python.

For the usage, the reported power consumption is not only the power consumption of the code you are running. This includes the global power consumption of all the processes running on the machine during this period, thus including the OS and other applications. So, we need to eliminate any extra programs and get the value of the devices when idling as the base level if possible. This will give the closest measure to the real power consumption of the measured code.

IV. POSTERIORI MEASUREMENT TOOLS ANALYSIS

AI researchers also proposed to estimate carbon emissions of the AI computation by posteriori tools, i.e., ML CO2 Impact and Green Algorithms. The key methodology of the tools is to estimate the power consumption after the computation process and achieve the carbon emissions from power consumption and related carbon intensity.

As shown in Table II, ML CO2 Impact tool is designed to estimate the carbon emissions produced by training ML models. The inputs include the geographical zone of the server, the type of GPU, and the training time, and the output is the approximate amount of CO2e.

TABLE II. ML CO2 IMPACT AND GREEN ALGORITHMS

	ML CO2 Impact	Green Algorithms
Energy consumption	runtime * power draw for GPU	runtime * (power draw for cores * usage + power draw for memory) * PUE * PSF
Hardware type	Mainly GPU type	GPU, CPU, CPU/GPU co-existing case, number of cores, memory
Usage factor	100% by default	100% by default and configurable
Other factors	/	Power Usage Effectiveness: the extra energy needed to operate the data center (cooling, lighting , etc.) Pragmatic Scaling Factor: multiple identical runs (e.g., for testing or optimization)

The inventors collected available public data for the computation including the TDP of the hardware, the location of the hardware, and the related carbon intensity (CO2e emissions per kWh).

Green Algorithms tool aims to estimate the carbon footprint of any computational task. Compared with ML CO2 Impact tool, it requires extra inputs of memory size, real usage factor of the processing core, PUE, and PSF.

Different from the on-the-fly measurement tools, the power consumption model of both tools uses TDP and runtime to achieve the power consumption, which means in the calculation the usage of cores is 100% by default. Green Algorithms tool allows to configure the real usage of cores and takes more quantifiable elements into consideration, i.e., memory power, PUE, and PSF, allowing users to estimate the power consumption more flexibly.

Considering carbon intensity, it is known that fossil fuels have the highest carbon footprints, for example, coal emits 820g of CO2e per kWh of electricity produced [35], while electricity generated by wind, solar, hydro, or nuclear power emits lower amounts of carbon footprints, i.e., 12g CO2e /kWh for wind, and 27~48g CO2e /kWh for all types of solar. In different countries and regions, even different electric power companies, the energy structure differs from others, and various energy sources would be used to generate electricity. The location matters as all servers are connected to local grids and they will have different amounts of CO2e emissions when consuming or generating the same amount of electricity, for example, 174g CO2e emission per kWh of electricity for France and 741g CO2e /kWh for Germany (at 12:00 PM on December 1, 2022) [36].

Both tools refer to public data for carbon intensity. They provide data reference of data centers including Google Cloud Platform, Amazon Web Services, and Azure. However, we can see clearly that there is no unified data source, location scope, and effective time due to differences in the data sources of the two tools.

From the preliminary analysis of the above two tools, we know that when evaluating carbon emission impact, both power consumption and carbon intensity should be considered. Parameters like hardware type, PUE, the usage of the core, and memory will contribute to the energy consumption. For carbon intensity, the location matters because of the different energy mix in different countries and

regions, but there are various data sources that may provide quite different location scopes and effective time. Also, we notice that the goal of these tools is to make people aware of the carbon emission impact, to provide a quick tool to evaluate the carbon emission during machine learning work and to recommend carbon reduction actions like selecting the cloud provider or server location wisely, buying carbon offsets, choosing clean energy, and improving AI algorithms to be green.

V. EVALUATION EXPERIMENTS

Our objectives are to verify the measurement tools and have some comments and suggestions proposed for measurement tools for AI models through our experiments.

We have developed a systematic methodology to carry out our experiments. First, a cloud-edge platform was set up with heterogenous hardware either X86-based, or ARM-based. These hardware devices have similar levels of computation capabilities to commercial end devices such as LiveBox, controllers on vehicles, etc. Then, we selected an AI application, in our case, we choose Person Re-Identification (Re-ID), which is the task of associating the same person taken from different cameras or from the same camera on different occasions [37]. Person Re-ID have wide usage in smart building and smart city scenario. Many Person Re-ID open-source models are accessible using various AI architectures, such as CNN, Transformer, or Long Short Term Memory (LSTM). Based on criteria, such as performance, release date, accessibility, etc., we have selected several Person Re-ID models with different AI model architectures. After that, we measured their power consumption during the training stage and inference stage when they are running on various types of hardware in rounds of experiments.

We selected four different Re-ID models: Fast-ReID (CNN), st-ReID (CNN), DeepPerson (LSTM), and Trans-ReID (Transformer).

We collected each model's detail information related energy consumption, total energy consumption, epoch times, training time, achieved performance, duration time/epoch, and energy consumption/epoch. Four models based-on three model types are trained and evaluated by balancing performance and energy consumption.

Figure 2 presents the total energy consumption with epoch times of 4 models. The orange columns present the energy consumption. It is found that the Fast-ReID has the lowest energy consumption and the Trans-ReID has the most. The st-ReID and deep-person are middle and about the same. And the top blue line stands for epoch times.

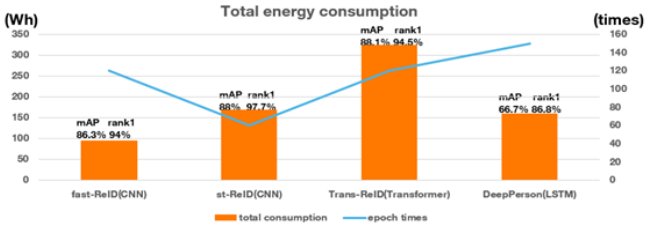


Figure 2. The comparation of total energy consumption with epoch times

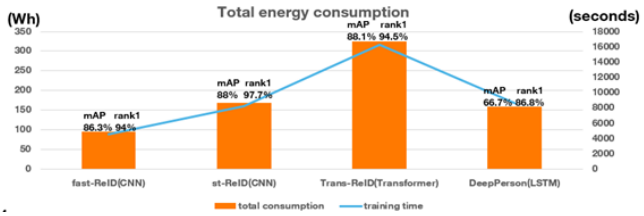


Figure 3. The comparation of total energy consumption with the training time

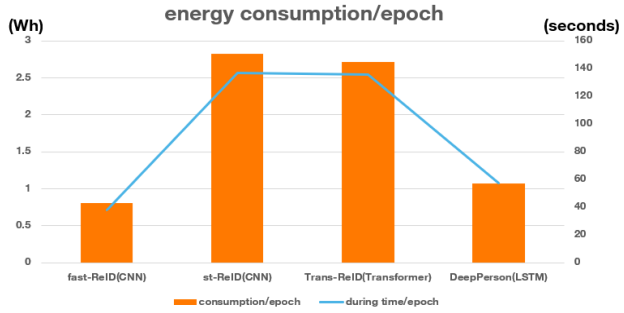


Figure 4. The comparation of energy consumption per epoch with the training time/epoch

Figure 3 presents the total energy consumption with the total training time of 4 models. The orange columns present the energy consumption as Figure 1. And the bottom blue line stands for the minimal training time to achieve the best performance. The change of training time is same as the change of total energy consumption.

Next, in Figure 4, the orange columns present the energy consumption per epoch and the blue line stands for the during time/epoch. An epoch refers to one cycle through the full training dataset. The change of during time per epoch is same as the change of energy consumption per epoch. The st-ReID and trans-ReID are about the same. The Fast-ReID is the lowest one.

After analyzing the information, we collected in the training phase, we got some conclusions. Total training power consumption is determined by the training algorithm and training time. Because GPU consumption is the largest part of energy consumption in the training phase and it worked at all speed and a constant power, training power consumption is in proportion to the training time in general. With the measurement tool, it can quantify power consumption in order to make more accurate assessment for different AI models.

Regardless of performance, for total power consumption, the Trans-ReID has the most one, the Fast-reid has the lowest one. The st-ReID and DeepPerson have the middle ones and are about the same. If considering the performance, First, according to our performance criterion (mAP great than or equal to 80% and Rank1 great than or equal to 90%), the DeepPerson is dropped. Second, st-ReID and Trans-ReID have the similar performance, but Trans-ReID consumed twice of energy than st-ReID. For st-ReID and Fast-ReID, st-

ReID raised about 2-4% as compared with Fast-ReID, but the st-ReID's energy consumption is 70 percent more than Fast-ReID.

Training power consumption/epoch is in proportion to the training time/epoch too. And the Trans-ReID and st-ReID are about the same and have the most consumptions. DeepPerson has the middle one and the Fast-ReID has the still lowest consumption.

Considering to performance and energy saving:

- The training program runs on GPU generally, and in the training phase Fast-ReID is the best choice for a balanced requirement of performance and energy saving.
- Trans-ReID is the newest model probably without optimized for energy consumption.

Evaluating the performance and power consumption for inference of different types of AI models, we designed three scenarios to compare the performance of the AI model under different computing requirements: Single case -- only one person in the video; Multi case: there are always more than two people in the video and Mixed case: dynamic picture, sometimes one person, sometimes many people, and sometimes no one. And in terms of the choice of energy consumption calculation, we also designed two different dimensions: Fixed time scenario -- processing AI application for 500s on different devices; Fixed task scenario -- processing a same 5mins video as input, stop application until all frame finished. In this way, we can get the influence results of AI's energy consumption from the two perspectives, the host machines, and the model itself. Don't like training part, for KPI of AI model inference performance we only take FPS (frames per second) to evaluate the processing speed of the AI applications and accuracy to evaluate the accuracy of video processing (identifying people for our ReID case).

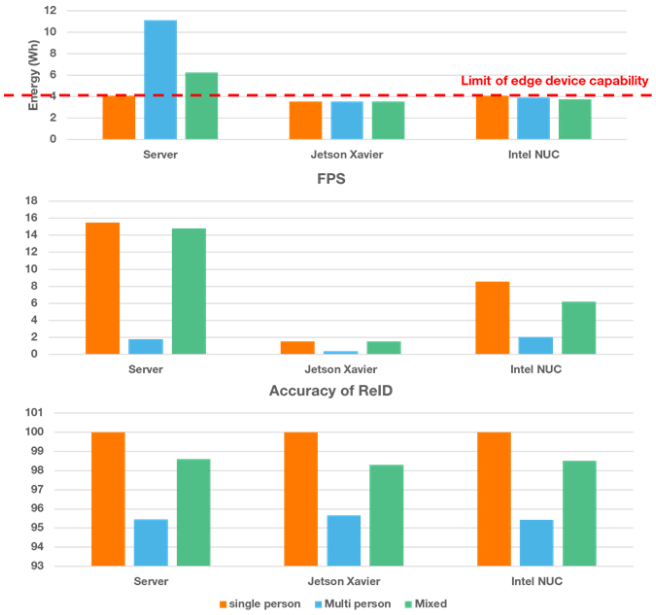


Figure 5. 500s fixed time experiment for Fast-ReID inference on tree different devices

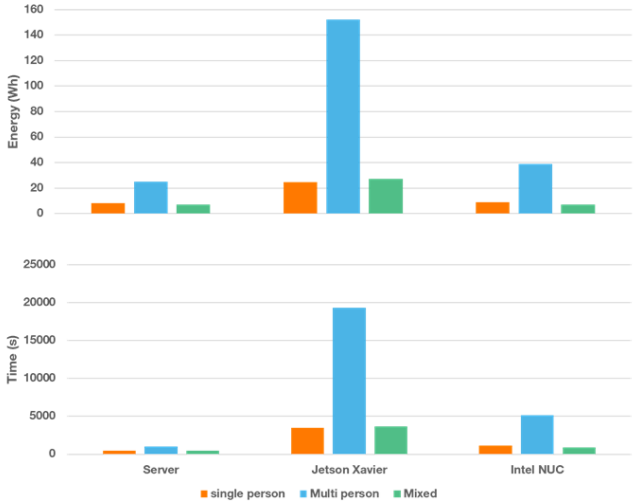


Figure 6. 5mins fixed task experiment for Fast-ReID inference on tree different devices

To minimize the error and contingency of experiment results, we carried out the same experiment five times separately and took the average value of them (there were no abnormal values). Each time, we firstly measured the energy consumption of the devices when idling as the base level, and then get the values while target AI application is running. Results of the AI application power consumption we take were the difference between them: $W_{final}=W_{total}-W_{idling}$.

Figure 5 above shows the fast-ReID model inference results on tree different host devices: edge server (Intel Xeon E5-2678 v3, GeForce GTX 1080 Ti), Intel NUC (Intel NUC8i7BEH) and Jetson AGX Xavier, we compared its total power consumption and performance using same input videos (three 500s videos with one person, more than two people and mixed cases).

From the chart we can see that the energy consumption of AI applications is positively correlated with the complexity of application scenarios when the hardware capability allows. On edge devices with limited computing power, AI application energy consumption is relatively fixed, close to the maximum energy consumption of the device. There is even an overload situation, which leads to overheating of the CPU and frequency of CPU will reduce. It will cause a lower energy consumption but with a poor performance (the FPS, processing speed, no influence for the accuracy as it depends on the models themselves). For the same architecture x86, although the server has a powerful computing capability, but in some cases, if the ability requirements and accuracy are not so high, small edge devices will be more reasonable.

Figure 6 shows the results of the fixed task scenario. This case, we used fast-ReID to finish processing of a same 5mins input video, we find that the energy consumption per unit time is indeed consistent with the parameters (CPU, GPU, RAM type etc.) of the device itself. But the total power

consumption depends on the computing capability of the hardware. Because we know that less capable hardware takes longer to complete the whole process, the longer AI process takes, the more energy consumption goes up. As normal sense, more powerful devices with stronger CPU and GPU frequencies, which is indicative of higher power consumption, but from our measurement results we find, it doesn't necessarily mean that their total power consumption for processing a same AI task will be higher. The time device taking to process the whole AI workflow is also an important impact point for the power consumption. For green AI trend, choosing the right host machine (the computing power it can provide) according to the requirements of the AI application, can optimize the energy consumption of AI applications.

Except CNN model fast-ReID, we also do the inference experiments of transformer type ReID model, Trans-ReID. The two are compared as shown in Figure 7. First, the two are close in accuracy, with no significant difference. As transformer is the latest model type for ReID use-case, unlike the CNN model type, which has been used and promoted for a long time. Trans-ReID seams still need some model optimization, especially on the edge devices. It's also cause of a more complex network of Transformer model. We can see that to process the same task program, Trans-ReID will take longer than Fast-ReID model. From the energy consumption result of a fixed running time of 500s, Fast-ReID is slightly lower than Trans-ReID. Which means that the energy consumption level per unit time of these two model types is very close. However, in a use-case of a fixed task, which is closer to the real AI program usage scene. Fast-ReID (CNN) model will have much lower power consumption than Trans-ReID, as more time are needed for Trans-ReID. Especially for multi person scenario, to finish processing a 5mins input video, Trans-ReID will take around 1700 Wh (out of order and no value in the chat) and the FPS of this model in the multi person case is very low, around 0.2.

We build a benchmark to compare the results of the on-the-fly and posteriori measurement tools. In the first experiment, the power consumption of Fast-ReID (CNN) model is measured by processing AI inference on a 500s video of a single person. The results of PyJoules and Jtop tools are selected as a baseline of the real-time measured power consumption. MLCO2 Impact and Green Algorithms tools are used to estimate the power consumption afterward, respectively. As is shown in Table III, three types of hardware have been evaluated: a server with GeForce GTX 1080 Ti, Intel Xeon E5-2678 v3 and a memory of 64GB, and two edge devices – one with Intel i7-8559U and a memory of 16GB, and the other with NVIDIA Jetson AGX Xavier, ARMv8 Processor rev 0 (v8l) and a memory of 32GB. For Green Algorithms tool, there is a default CPU usage of 100% and a configurable CPU usage that can be estimated based on the observation of the AI processing experiment. The PUE used in the calculation is 1 because we use local private infrastructure instead of cloud services and ignore the power consumption of cooling or lighting. The memory power draw only depends on the size of memory available (0.3725 W per GB).

Figure 7. Experiment results comparation of Fast-ReID and Trans-ReID on edge server

In the second experiment, as is shown in Table IV, four different Re-ID models are evaluated at the training stage. Since both GPU and CPU cores will be used in the AI training process, power consumption should be considered for both.

With default configuration of 100% usage of cores, the results of the two estimation tools are 1~3x of data measured by "on the fly" tools. If applying estimated usage of cores based on observation in Green Algorithms tool, for example, 30% for servers and 70% for edge devices in AI inference experiment, and 66% for GPU and 10% for CPU in AI training experiment, the results are close to real-time measured ones.

In Figure 8 and Figure 9, we can see clearly that for servers, in both AI inference and training case, the "green" estimated results of Green algorithms tool, are closer to the on the fly measured results, due to the use of estimated real usage of cores. On the other hand, for resource constrained edge devices in AI inference case, all estimated values are close to experimental ones, because that the real CPU usage tends to be full of use on these edge devices.

Figure 8. Fast-ReID inference Experiment vs. Estimation (Server & Edge devices)

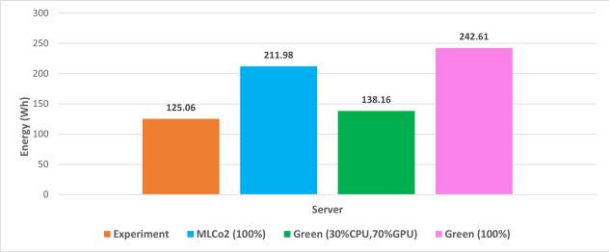


Figure 9. Fast-ReID training Experiment vs. Estimation (Server)

VI. Our Measurement Tool

Based-on the state of art measurement tools that we analyzed and experimented on the power consumption of AI models, we developed an integrated measurement tool by asking ourselves what kind of tool an AI practitioner or user want? We found that they may like real time feedbacks, with carbon emission results, with more detailed information, more visualized. The tool can work well on different hardware, and easy to use, etc. Our proposition is to develop a new AI carbon footprint measurement tool that could satisfy their needs.

According to Sections II, III, and IV, these types of tools have their own pros and cons. The priori tools, which can be used to calculate the model operation in terms of FLOPs, but they only support simple models. The second type on-the-fly, which can measure electricity consumption during computation, but they have poor compatibility and visualization; the third type--posteriori tool, which can estimate CO2 eq for a give computation, but the accuracy is based on input parameters, and it is not in real time. Our tool has taken reference from the good elements from these tools and provide wider support, good compatibility and real-time measurement.

With our tool, we have verified AI model's optimization, which can reduce the number of operations or the model size.

In our experiment, the Multiple Object Tracking (MOT) [38] case was chosen. And we used an optimization method named Knowledge Distillation [39] to reduce model size without much performance degradation. Knowledge Distillation refers to the process of transferring the knowledge from a teacher model to a smaller student model. Our tool is used to verify the optimization results on the aspect of carbon footprint, which are showed in Table V. From the results we can find that after model optimization, the size of student model is reduced by 75% compared with that of teacher model and the power consumption and carbon footprint are reduced by 71% at training phase and 60% at inference phase with less than 10% loss of model performance based-on MOTA.

VII. Conclusion

We have been carrying out series of experiments to verify the measurement tools. For different types of measurement tools, we found that:

The effectiveness of priori measurement tools relies on their detailed implementation. The application of priori measurement tools is limited. The tools just support one special framework and a subset of types of model layers.

The on-the-fly tools can be used during the processes of AI programs; however, they are limited. PyJoules or JouleHunter can be used to get power consumption (CPU, GPU, RAM) of large AI programs on different x86 architectures devices, while for architectures ARM devices, only Jtop supported. Ideally, it is better to develop and use the same cross-platform tool. However, the comparison of experimental and estimated results shows that the error of the on-the-fly measurement tools is acceptable.

The posteriori measurement tools can be used for power consumption estimation after the AI processing by knowing the runtime and the parameters of hardware (CPU, GPU, memory, etc.). For resource-constrained edge devices, the resources usually tend to be nearly full of use, and the tools with a default configuration are able to make a quick estimation of the power consumption. For servers that have more resources and stronger processing capabilities, if extra information can be given, for example, the real usage of the cores, the Green Algorithms tool will be optimized to make close estimations to real-time measured power consumption. Both tools can provide different CO2e emissions due to different locations where the AI computation is processed. The researchers aim to remind people to carefully select the cloud providers and locations for AI services when carbon impacts should be taken into consideration.

We have selected an AI use case: Re-ID, which can be realized by various types of AI architecture: CNN, LSTM, and Transformer. Once the specific AI model for each type is selected, the power consumption of the selected AI models is measured during the training and inference stages when they are running on different edge devices. The experimental results show that the total training power consumption of the AI model is determined by the training algorithm and training time. Training power consumption is in proportion to the training time in general. With the measurement tool, it can quantify the power consumption to make a more accurate assessment for different AI models. The power consumption of AI applications is positively correlated with the complexity of application scenarios when the hardware capability allows.

Finally, we developed our own measurement tool for AI model carbon footprint measurement and verify the tool in an AI model's optimization use case. We hope data scientist community would propose new optimization techniques to evaluate the impact by using such kind of tool. In the future, we plan to extend our measurement tool that can be run on the cloud.

References

[1] R. Zhou, T. Zheng, X. Wang, L. Wang, and E. Sirvent-Hien, "Capability and Applicability of Measurement Tools for AI Model's Environmental Impact", COMPUTATION TOOLS 2023, pp. 1-8, 2023.

[2] P. M. Forster et al., "Indicators of Global Climate Change 2022: annual update of large-scale indicators of the state of

the climate system and human influence", Earth System Science Data. 15 (6): 2295–2327, 2023.

[3] H. Ritchie, "Sector by sector: where do global greenhouse gas emissions come from?". [Online]. Available from: https://ourworldindata.org/ghg-emissions-by-sector/ [retrieved: 01, 2024].

[4] Ericsson, "ICT's potential to reduce greenhouse gas emissions in 2030". [Online]. Available from: https://www.ericsson.com /en/reports-and-papers/research-papers/exploring-the-effects-of-ict-solutions-on-ghg-emissions-in-2030/ [retrieved: 01, 2024].

[5] L. H Kaack et al., "Aligning artificial intelligence with climate change mitigation", Nature Climate Change, pp. 1–10, 2022.

[6] S. Matleena, "Amazing AI Statistics (2022): Stunning Growth of AI". [Online]. Available from: https://zyro.com/blog/ai-statistic/ [retrieved: 01, 2024].

[7] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI", Communications of the ACM, vol. 63, No. 12, pp. 54-63, Dec. 2020.

[8] OpenAI, http://openai.com/ [retrieved: 01 2024].

[9] E. Strubell, A. Ganesh, and A. McCallum, "Energy and Policy Considerations for Deep Learning in NLP", 57th Annual Meeting of the Association for Computational Linguistics (ACL). Florence, Italy. Jul. 2019, doi: 10.18653/v1/P19-1355.

[10] D. R. So et al., "Primer: Searching for Efficient Transformers for Language", 35th Conference on Neural Information Processing Systems (NeurIPS 2021), virtual, 2021, doi: 10.48550/arXiv.2109.08668.

[11] D. Patterson et al., "Carbon Emissions and Large Neural Network Training", 2021, doi: 10.48550/arXiv.2104.10350.

[12] D. Patterson et al., "The Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink", Computer, vol. 55, pp. 18-28, Jul 2022.

[13] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, "Quantifying the Carbon Emissions of Machine Learning", 2019, doi: 10.48550/arXiv.1910.09700.

[14] A. Ligozat and S. Luccioni, "A Practical Guide to Quantifying Carbon Emissions for Machine Learning researchers and practitioners", [Online]. Available from: https://hal.archives-ouvertes.fr/hal-03376391/document/ [retrieved: 01, 2024].

[15] A.-L. Ligozat, J. Lefèvre, A. Bugeau, and J. Combaz, "Unraveling the hidden environmental impacts of AI solutions for environment", arXiv:2110.11822, 2021.

[16] CEN-CENELEC JTC 21 "Artificial Intelligence", [Online]. Available from: https://www.cencenelec.eu/areas-of-work/cen-cenelec-topics/artificial-intelligence/ [retrieved: 01 2024].

[17] CEN/CENELEC, "Standardization landscape for energy management and environmental viability of green data centres", [Online]. Available from: ftp://ftp.cencenelec.eu/EN/EuropeanStandardization/HotTopics/ICT/GreenDataCentres/GDC_report_summary.pdf [retrieved: 01 2024].

[18] ISO/IEC 42001:2003 Afnor editions.

[19] PowerAPI, http://www.powerapi.org/ [retrieved: 01 2024].

[20] PyJoules, https://github.com/powerapi-ng/pyJoules/ [retrieved: 01 2024].

[21] Keras-flops, https://github.com/tokusumi/keras-flops/ [retrieved: 01 2024].

[22] Torchstat, https://github.com/Swall0w/torchstat/ [retrieved: 01 2024].

[23] torchsummaryX, https://github.com/nmhkahn/torchsummaryX/ [retrieved: 01 2024].

[24] flops-counter, https://github.com/sovrasov/flops-counter.pytorch/ [retrieved: 01 2024].

[25] JouleHunter, https://github.com/powerapi-ng/joulehunter/ [retrieved: 01 2024].

[26] Jtop, https://pypi.org/project/jetson-stats/ [retrieved: 01 2024].

[27] CarbonAI, https://github.com/Capgemini-Invent-France/CarbonAI/ [retrieved: 01 2024].

[28] L. Lannelongue, J. Grealey, and M. Inouye, "Green algorithms: quantifying the carbon footprint of computation". Advanced Science, vol 8, issue 12, 2021, doi: 10.48550/arXiv.2007.07610.

[29] GreenHouse-Gas-protocol, https://csanr.wsu.edu/the-basics-of-carbon-markets-and-trends/ [retrieved: 01 2024]

[30] Issues of torchstat from gihub, https://github.com/Swall0w/torchstat/issues/12/ [retrieved: 01 2024].

[31] Global architecture picture, https://powerapi.org/reference/overview/ [retrieved: 01 2024].

[32] Intel Sandy Bridge core, https://www.cpu-world.com/Cores/Sandy%20Bridge.html [retrieved: 01 2024].

[33] Pyinstrument, https://pyinstrument.readthedocs.io/ [retrieved: 01 2024].

[34] NVIDIA Volta architecture, https://www.nvidia.com/en-us/data-center/volta-gpu-architecture/ [retrieved: 01 2024].

[35] https://www.world-nuclear.org/ [retrieved: 01 2024].

[36] https://app.electricitymaps.com/map/ [retrieved: 01 2024].

[37] Person-re-Identification, https://paperswithcode.com/task/person-re-identification [retrieved: 01 2024].

[38] Z. W. Pylyshyn and R. W. Storm, "Tracking multiple independent targets: Evidence for a parallel tracking mechanism" Spatial Vision. 3 (3): 179–197, 1988, doi:10.1163/156856888X00122.

[39] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network", 2015, arXiv:1503.02531.

TABLE III. FAST-REID, SINGLE PERSON, RUNNING TIME = 500S, AI INFERENCE

Measurement tools	**GPU/CPU type**	**Power consumption**				**Carbon emissions[a]**
		CPU (W)	***Usage***	***Memory (GB)***	***Total (Wh)***	***CO2e(mg)***
1 On the fly – PyJoules	Server: GPU: GeForce GTX 1080 Ti CPU: Intel Xeon E5-2678 v3 Memory: 64GB				7.2	
2 A posteriori - MLCO2 Impact		120	100%		16.67	633.46
3 A posteriori – Green Algorithms		120	30%/100%	64	8.31/19.98	426.18/ 1002
1 On the fly - PyJoules	Intel machine II:				4.1	

2 A posteriori - MLCO2 Impact	CPU: Intel i7-8559U Memory: 16GB	28	100%		3.89	147.82
3 A posteriori – Green Algorithms		28	70%/100%	16	3.55/4.72	182.04/ 241.86
1 On the fly - Jtop	ARM: NVDIA Jetson AGX Xavier CPU: ARMv8 Processor rev 0 (v8l) Memory: 32GB				3.8	
2 A posteriori - MLCO2 Impact		30	100%		4.17	158.46
3 A posteriori – Green Algorithms		30	70%/100%	32	4.57/5.82	234.45/ 298.55

a. The reference location is Europe, France.

TABLE IV. FAST-REID, ST-REID, DEEPPERSON, AND TRANS-REID, AI TRAINING

Measurement tools	**Test ML model**	**Running time(s)**	**Power consumption**						**Carbon emissions[a]**
			GPU (W)	***Usage***	***CPU (W)***	***Usage***	***Memory (GB)***	***Total (Wh)***	***CO2e(g)***
1 On the fly – PyJoules	Fast-ReID, CNN	4625						125.06	
2 A posteriori - MLCO2 Impact			120	100%	45	100%		211.98	8.06
3 A posteriori – Green Algorithms			120	66%/100%	45	10%/100%	64	128.16/242.61	7.08/12.44
1 On the fly - PyJoules	st-ReID, CNN	7988						212.26	
2 A posteriori - MLCO2 Impact			120	100%	45	100%		366.12	13.91
3 A posteriori – Green Algorithms			120	66%/100%	45	10%/100%	64	238.62/419.01	12.24/21.49
1 On the fly - PyJoules	DeepPerson, LSTM	8326						201.74	
2 A posteriori - MLCO2 Impact			120	100%	45	100%		381.61	30.35
3 A posteriori – Green Algorithms			120	66%/100%	45	10%/100%	64	248.72/436.74	26.69/46.87
1 On the fly - PyJoules	Trans-ReID, Transformer	17426						451.7	
2 A posteriori - MLCO2 Impact			120	100%	45	100%		798.69	30.35
3 A posteriori – Green Algorithms			120	66%/100%	45	10%/100%	64	520.55/914.09	26.69g/46.87

a. The reference location is Europe, France.

TABLE V. MODEL OPTIMAZATION RESULTS WITH OUR MEASUREMENT TOOL

	Model size	**Gflops**	**Model performance (MOTA)**	**Training power consumption (Wh)/carbon footprint(gCO2e) [a]**	**Inference power consumption (Wh)/carbon footprint(gCO2e) (example video: 5460 frames)**	
					GPU: NVIDIA RTX 3080	**GPU: NVIDIA RTX 2080**
Teacher model	750M	793.21	61.80%	57.43Wh/epoch	12.39Wh	16.34Wh
				4.88gCO2e/epoch	1.05gCO2e	1.38gCO2e
Student model	190M	207.35	56.20%	6.48Wh/epoch	5.72Wh	7.02Wh
				0.55gCO2e/epoch	0.48gCO2e	0.59gCO2e

a. The reference location is Europe, France.

Adaptive Transmission Range for Decentralised Foraging Robots Using Autonomic Pulse Communications

Liam McGuigan, Roy Sterritt, Glenn Hawe
School of Computing, Faculty of Computing, Engineering and the Built Environment
Ulster University
Jordanstown, N. Ireland
email: mcguigan-l8@ulster.ac.uk, r.sterritt@ulster.ac.uk, gi.hawe@ulster.ac.uk

Abstract— **A robot swarm which is to be deployed without the need for regular human input is required to be autonomous, capable of the self-management needed for operation in distant, complex, or changing environments. Communication between the individual robots is an essential facet of the swarm's ability to cooperate and adapt, and use of a fixed transmission range may result in issues with connectivity, inefficiency, or lead to constraints on robot movement. In this research, an Autonomic Pulse Communications system is developed for a simulated robot swarm, adaptively selecting a suitable transmission range based on local measurements of swarm density. The system is able to successfully share data around the swarm within a fixed time period, even with low density swarms and with a high robustness to communications loss. Further, the APC system is used in a simulated foraging task, performing as well as a previous decentralised autonomic system, but without the need for prior selection of a suitable transmission range.**

Keywords- Swarm robotics; Self-adaptation; Autonomic Computing; Swarm communication; Simulation.

I. INTRODUCTION

This paper is an extended version of the work published in [1], extending those results and presenting further research.

Swarm robotics, the study of how individual behaviours within a group of robots may combine through local interactions to create a more complex set of behaviours [2], has potential applications in fields such as space exploration [3], precision agriculture [4], and disaster response [5], where many small, simple robots can cover a much larger area than a single monolithic craft.

The size of the swarm, its decentralised nature, and the conditions in which it may potentially operate mean that a swarm should be able to act on its own, adjusting its behaviour according to a changing situation without the need for any external guidance [6]. Autonomic Computing concepts [7][8] can assist in achieving swarm self-adaptation, making use of a Monitor, Analyse, Plan and Execute loop, with a shared Knowledge base, known as MAPE-K, as described in [7] to assess the situation, identify any changes necessary, and implement them.

As swarms are decentralised, their ability to adapt depends on their cooperation through sharing information on which to base decisions and come to an agreement on actions to be taken. When the swarms are reliant on local communication with neighbouring robots, the effective range of that communication matters. Too small, and robot behaviour may need to be constrained to maintain communication links with other members of the swarm. Too large, and it may be an inefficient use of battery power, lead to communication interference, or even be detrimental to overall performance.

In previous work, a decentralised swarm made use of an autonomic system to help adjust a range over which robots would broadcast for help in a foraging task [9]. This worked by using a fixed range pulse message between robots to help estimate the density, but it was found that the range of this pulse message needed to be set for differing swarm densities. If this is not initially known, performance would be degraded.

The objective of this work is to implement an adaptive system for setting the range over which a robot broadcasts information, according to the local density of the swarm, detected at run-time. This will then be used in a simulation of foraging robots to resolve the requirement for a pre-set pulse range.

The rest of this paper is structured as follows. Section II discusses related work in swarm self-adaptation and autonomic systems used to develop the Autonomic Pulse Communication (APC) system presented. Section III discusses the design of the APC system and how it estimates local density in order to determine a suitable broadcast range. Section IV describes the data sharing task designed to test the APC's ability to maintain communication in the swarm, Section V introduces the test scenarios used, and Section VI presents the results. Section VII puts the APC system to work in a simulation of foraging robots, comparing the results against the performance of the previous decentralized system. Section VIII discusses the results, and Section IX concludes the paper with a summary, and directions for future research.

II. RELATED WORK

In the context of a robot swarm, a distinction can be made between the adaptation of individual robots, and that of the swarm as a whole. This can be related to the idea of *self-expression* [10][11], in which the swarm at large can be reconfigured. Such swarm-level adaptation can then take advantage of wider knowledge to make changes to swarm composition [12], or cooperative strategies [13].

To achieve swarm-level adaptation, however, cooperation and communication becomes essential. Individuals must share data in order to collectively recognize the need to adapt, and then to decide on the new course of action. Consensus problems, typified in swarm research as the best-of-*n* problem [14], in turn require some means of communicating the currently held opinion of any one robot to neighbours.

Direct communication between neighbours requires a degree of connectivity between the robots in the swarm. All-time connectivity uses approaches such as control laws to balance both the task at hand and the need for connectivity [15][16]. Such approaches necessarily restrict the movement of individual robots, and may be detrimental to performance [17]. Relay approaches may help with this, by delegating the job of providing connectivity to only some portion of the swarm [18][19].

Relaxing the need for all-time connectivity, path planning approaches [17] or ferries [20] may allow for an intermittent approach, but add complexity to swarm behaviour and require some or all robots to halt their task periodically.

The absence of explicit attempts to maintain communications links may be described as opportunistic, with robots transferring data to others in range when their paths happen to cross. This is the least restrictive approach and does not require dedicated roles or periodic rendezvous, but at the expense of guaranteed connectivity.

A crucial factor, regardless of the approach taken, is the communication range. The further apart any two robots may be when maintaining a communication link between them, the freer the robots are to move, and the fewer the number of robots that may be critical to network connectivity. As higher ranges may require more power and result in network interference [21], and lower ranges may decrease connectivity, finding a suitable broadcast range becomes desirable.

The mechanism for achieving this, described in the next section, is based on the existing concept of Pulse Monitoring (abbreviated to PBM due to its extension of Heart Beat Monitoring, HBM) [22], in which a periodic heartbeat message has a pulse encoded within it, allowing a component in a system to indicate its current health status. The concept has been explored in applications such as personal computers [23], telecommunications [24], and cluster management [25]. In order to support a reflexive reaction by minimising the processing required by a recipient, health-related data may be included in the message [24].

Pulse monitoring may be applied to a robot swarm, such as in [26], where it may be a means for a ruler craft during the Prospecting Asteroid Mission to monitor the health of workers under their control. However, another perspective may be used. In a dynamic swarm, where there is a need for scalability, it may be undesirable for one robot to track another's health over a significant period of time, and it cannot be expected that any one robot would rely upon another *specific* robot to assist in a task. Instead, pulses received during a small interval may represent the health of the local neighbourhood, allowing a robot to determine if its own status is abnormal, or provide early-warning of danger by noting problems developing in neighbouring robots.

Pulse monitoring is typically concerned with reporting on the health of whatever aspect is being monitored, as a form of failure management. In this paper, the concept is adapted to allow an individual robot to measure the local density of the swarm through the receipt of pulse messages from neighbouring robots that contain information about the source robots' positions. In this way, the "I am healthy" signal is replaced with one saying, "I am here". The design of the APC system is described in the next section.

III. AUTONOMIC PULSE COMMUNICATIONS

The goal of the APC system described in this paper is to provide a mechanism for the adaptive adjustment of the transmission range used for inter-robot communication, in order to avoid the pitfalls that come with needing to set the range used at the start of the mission.

To achieve this, the concept of PBM described in the previous section is adapted to repurpose the regular signal sent by each robot. In the Decentralised Autonomic Manager (DAM) described in [9], robots used periodic pulses to determine the local density of the swarm, but the pulse required a fixed transmission range used by each robot. If different transmission ranges were to be used, the density could not be easily calculated.

This problem is resolved by having each pulse also contain the position of the sending robot, allowing the distance from the pulse origin to the receiving robot to be calculated. Alternatively, situated communication [27] may be used to derive distance information from the received signal. Whichever approach is taken, the distance may be used to estimate the local density.

Fig. 1 (a) shows a case in which Robot A has a number of neighbours, all broadcasting pulse messages at different ranges, each of which is transmitted far enough to reach the robot. To simplify the example, all robots are shown to be sending their messages simultaneously, but the same process applies as long as all messages are received within the same short period of time. Each pulse contains the position of its sending robot.

By totalling the measured ranges of the received pulses, the APC system is able to calculate the average distance of pulse messages received. The local density, ρ, is then calculated as:

$$\rho = n / \pi \bar{d}^2, \quad (1)$$

where n is the number of received pulses in the time period, and $\bar{d}$ is their mean distance.

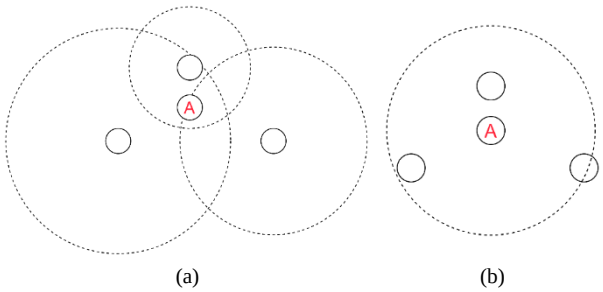


Figure 1. A robot receives pulse messages from neighbours (a), and uses the encoded distance information to calculate a suitable range for its own pulse message.

Given a density, the APC system may then use a density-pulse range relationship provided in its knowledge base. This is tailored for the assigned task such that the ideal range for this task can be determined. In Fig. 1 (b), Robot A sends out its own pulse, with the range determined by that relationship, enabling its pulse message to reach its neighbours.

The APC system is only able to calculate a suitable local density if it receives pulse messages during the period between sending its own pulses. In Fig. 2 (a), the nearby robots are not sending pulse messages with sufficient range to reach Robot A. If none are received, the robot is considered to be isolated from the rest of its swarm. Its current pulse range may not be sufficient to reach its own neighbours, as in Fig. 2 (b), and so it gradually increases its broadcast range on subsequent pulses. This increases the chance that the robot will later reconnect with the other robots, in turn influencing future selections of the transmission range.

In addition to the distance information required by the APC system, pulse messages may also share arbitrary data, sent on each broadcast, for the purpose of spreading information throughout the swarm. In this work, the data packet is small and does not grow with size, so a simple strategy of sharing data with neighbouring robots is used, in which no individual robot needs to care about which robots receive a broadcast. This approach scales with the swarm size, as the underlying behaviour of the robots does not need to change for larger swarms.

IV. Data Sharing From A Single Robot

This research employs a time-stepped simulation of a homogeneous swarm of robots, tasked with sharing a piece of data throughout the swarm. In this simulation, elapsed time is measured in simulation ticks, while distances are in arbitrary units defining the simulation space, hereafter referred to simply as "units". The robots are represented by a position only, with no physical size or robot-robot collisions. The purpose of this task is to determine how well a swarm of robots may share a single piece of information, initially held by only one robot in the swarm, with the rest of the members.

The swarm of robots, each using an APC system configured with a pulse period of 10 simulation ticks, and a fixed pulse range of 10 units, is placed in a circular map. Each robot stores a Boolean flag, initially set to false. At the start of each run, a robot is selected at random from the swarm and their flag is set to true. Any robot whose flag is true will share this data via the APC system. Receipt of the flag will cause a robot to set its own flag to true, and commence its own sharing.

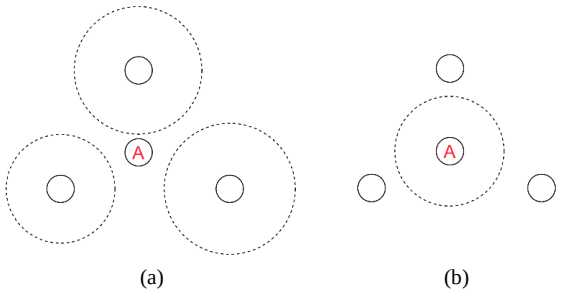


Figure 2. Nearby robot pulses are not strong enough to reach Robot A (a), which must gradually increase its own range to compensate (b).

During the test, the robots may wander freely throughout the map. Each tick of the simulation, a robot picks a random direction in two dimensions. If the robot is able to move forward one unit distance without leaving the map, the robot moves to that location, otherwise it will not move in this simulation update.

The test is left to run for 250 simulation ticks, and at the end, the success of the swarm in sharing the data is scored by the percentage of robots with their flag set to true. The test duration used will impact the density-range calculation, as the ideal range data used will be that which enables the swarm to reliably share the data with all members within 250 ticks.

All tests were run with the APC system set to stagger pulse times, rather than having all robots pulse simultaneously. This removes any requirement of the APC system to synchronise robot behaviour, while also avoiding flooding the available bandwidth with messages sent simultaneously.

V. Test Scenarios

The following subsections describe the particular test scenarios run. Each test was run 50 times, and the results averaged across all runs.

A. Density-Pulse Range Relationship

To determine the relationship between the swarm density and the ideal pulse range to use, a set of simulations was run, for swarm sizes of 50, 100, 200, 500 and 1,000 robots, and maps with radii of 25, 50, 75 and 100 units.

The ideal pulse range for a given combination was determined by taking the lowest pulse range for which over 99.5% of the swarm, on average, received the data.

B. Pulse Period

This test explores how the APC pulse period affects the ability of the swarm to share the data. A map with a radius of 100 units was used, with the pulse range fixed at 10 units. The test was repeated with the five swarm sizes from the previous test, and pulse periods of 2, 5, 10, 15, 20 and 25 ticks. Each combination of swarm size and pulse period was tested, and the scores from each scenario are compared to evaluate the effects.

C. Test Duration

This test explores how the APC pulse period affects the ability of the swarm to share the data. A map with a radius of 100 units was used, with the pulse range fixed at 10 units. The test was repeated with the five swarm sizes from the previous test, and pulse periods of 2, 5, 10, 15, 20 and 25 ticks. Each combination of swarm size and pulse period was tested, and the scores from each scenario are compared to evaluate the effects.

D. Adaptive Pulse Range

The equation relating density and pulse range derived from the previous test is now used in the APC system to adaptively adjust the pulse range, based on the local swarm density. This test looks at the ability of this adaptive APC system to set an appropriate pulse range, and therefore share the data throughout the swarm.

The maps and robot counts are the same as those listed from the Density – Pulse Range tests. Each APC system starts with a pulse range of one unit, and uses a period of 10 ticks. The score for each combination of map and swarm size is measured, and compared against the best performing fixed range communication established in the previous test.

E. Communications Loss

To explore the impact of communications no longer being guaranteed to arrive, a swarm of 200 robots is tested on a map with a radius of 100 units. The simulation is configured with a probability of any robot receiving a broadcast range, and the test is run with probabilities of 20%, 15%, 10%, 5%, 4%, 3%, 2% and 1%, together with a test of the fixed range communications with a probability of communication success set to 5%. Every 10 ticks, the number of robots that have the flag set to true are recorded, and the results compared.

VI. Results

The following subsections discuss the results of the tests described above.

A. Density-Pulse Range Relationship

Table I shows the best performing ranges and their respective scores for each combination of map radius and swarm size, while Fig. 3 shows the relationship between swarm density and best performing pulse range.

Fitting a trend line to the plot leads to an equation for determining the pulse range to use, given the density of the swarm:

$$r = 0.5884 \times \rho^{-0.652}, \quad (2)$$

where r is the pulse range, and ρ is the swarm density.

B. Pulse Period

Fig. 4 shows the performance for each size of swarm, as the pulse period is increased. Increasing the period results in a drop in the score achieved, which is less prominent in the largest swarms, and is most clearly seen with a swarm of 200 robots.

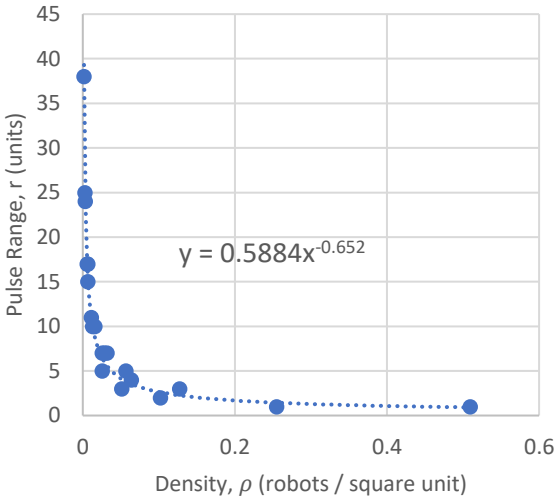


Figure 3. Plot of ideal pulse range against swarm density, for which ≥ 99.5% of the swarm received data shared starting with a single robot.

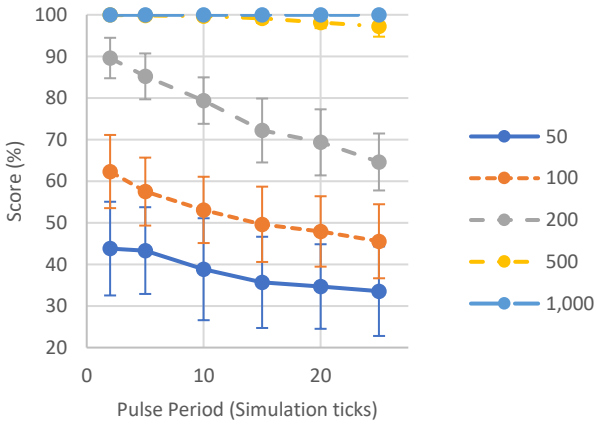


Figure 4. Score achieved by the swarm in sharing a data starting with a single robot, for the given pulse periods.

C. Test Duration

Fig. 5 shows the performance of each swarm size over time. It can be seen that denser swarms more quickly reach

TABLE I. Performance of swarm in sharing data using ideal pulse ranges for each combination of map and swarm size

Swarm Size	Map Radius							
	25		50		75		100	
	Range	Score	Range	Score	Range	Score	Range	Score
50	5	99.84%	15	99.72%	25	99.67%	38	99.88%
100	3	99.82%	10	99.62%	17	99.54%	24	99.60%
200	2	99.97%	7	99.94%	11	99.57%	17	99.53%
500	1	99.96%	4	99.94%	7	99.90%	10	99.74%
1,000	1	100%	3	100%	5	99.95%	7	99.87%

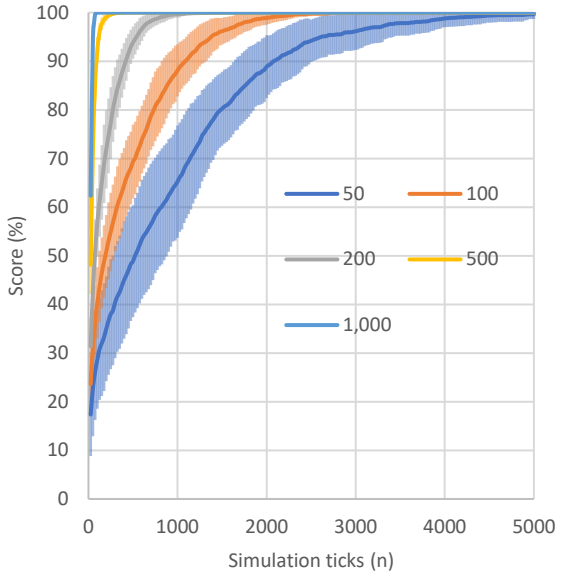


Figure 5. Score improvement during the data sharing scenario for swarms of different sizes in a 100-unit radius map. Shaded areas indicate one standard deviation.

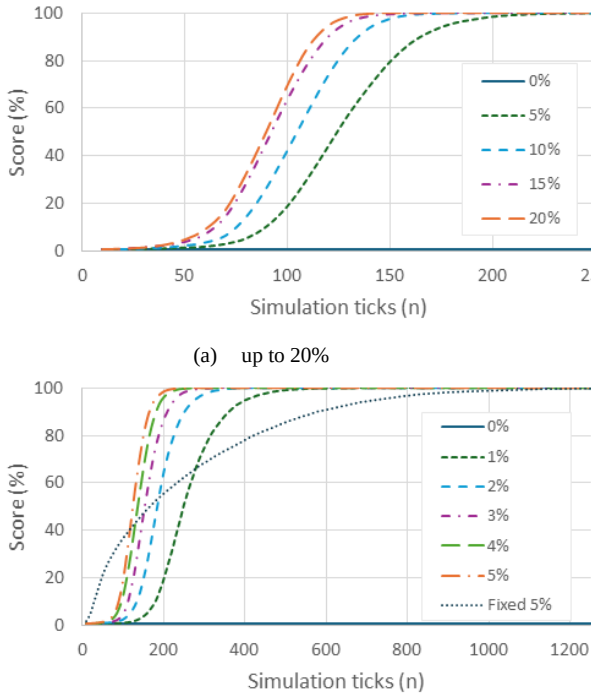
the point where all robots have received the information, but less dense swarms may require much more than the 250 ticks used as a standard in other tests.

D. Adaptive Pulse Range

Table II shows the performance of the swarm, and average pulse range used, for each combination of map radius and swarm size. All scenarios achieved greater than the 99.5% score used as a benchmark in the fixed range tests, and all but three of the scenarios received a perfect score. The average pulse range used by the swarm can be compared against the ideal fixed ranges shown in Table I, and shows that higher density swarms make use of shorter-range pulses on average.

E. Communications Loss

Fig. 6 shows the performance of the swarm of 200 robots on a map with a 100-unit radius, in scenarios where the probability of a communications broadcast being received by a robot was 20% or lower. In addition, the chart shows the performance of the APC system running with a fixed pulse range, where communications have a 5% probability of succeeding.



(a) up to 20%

(b) up to 5%, with fixed pulse range

Figure 6. Performance of swarm in sharing data originating with a single robot, for given chances of a successful communication. At 0%, no messages are successfully received.

VII. Foraging Robots with APC

The APC system was inspired by the results of previous work on foraging robots [9], showcasing the need for an adaptive pulse range. That scenario is revisited here, employing the APC system to fulfil that requirement.

This research makes use of a time-stepped simulation of a heterogeneous swarm of agents, tasked with foraging for items within a square arena, as presented in previous work [9][28]. In this task, robots and items are placed within a grid at random, as shown in Fig. 7. Each robot and item may be either of two possible types, denoted by their colour. A single cell contains only one item, but may contain any number of

TABLE II. Average pulse ranges and performance for swarm sharing data using Autonomic Pulse Communication

Swarm Size	Map Radius							
	25		50		75		100	
	Range	Score	Range	Score	Range	Score	Range	Score
50	7.28 ± 0.32	100%	16.27 ± 0.33	100%	23.34 ± 0.47	99.96%	29.42 ± 0.51	99.64%
100	4.61 ± 0.14	100%	10.68 ± 0.21	100%	16.50 ± 0.18	100%	21.63 ± 0.26	99.98%
200	3.04 ± 0.07	100%	7.14 ± 0.13	100%	11.06 ± 0.17	100%	14.87 ± 0.19	100%
500	1.89 ± 0.02	100%	4.02 ± 0.08	100%	6.35 ± 0.07	100%	8.98 ± 0.11	100%
1,000	1.38 ± 0.00	100%	2.65 ± 0.03	100%	4.18 × 0.05	100%	5.82 ± 0.06	100%

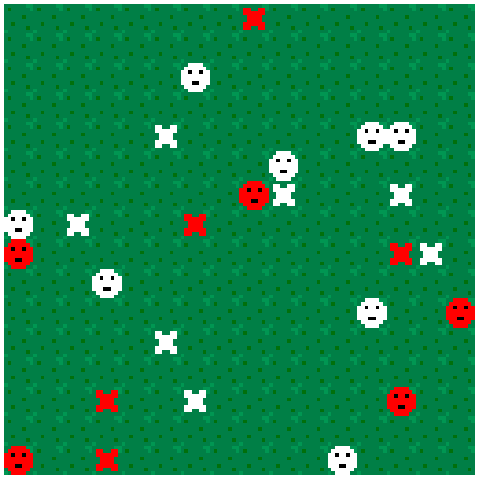


Figure 7. A portion of the world state during a simulation. The colour of a robot (face) or item (cross) indicates its type.

robots, with potential collisions between robots ignored by the simulation as each cell may be considered much larger than any one robot.

The simulation is updated in a time step manner, with each robot updated in turn for each tick of the simulation. The behaviour of the robots is based on the particular cooperation strategy they are using, as presented in [28]. In this work, the Help Recruitment and Blackboard strategies are used.

In the Help Recruitment strategy, as shown in Fig. 8, a robot begins in the Explore state. In this state, the robot moves to an adjacent cell in search of an item every tick of the simulation. If it finds an item, it moves to the Forage state, otherwise it will continue to Explore in the next tick.

In the Forage state, the robot determines the type of the item at that location. If the robot and item share a type, the robot is able to successfully forage the item, and so returns to the Explore state. However, if the robot and item are of different types, the robot must cooperate with neighbours. To do so, it broadcasts a recruitment message at a given range, containing the location and type of the item to be foraged, then moves to the Wait for Offers state.

Nearby robots in the Explore state that receive a help request will inspect the item's type, and if they are able to help, will send a response offering help, before moving to the Wait for Assignment state.

The original robot requesting help will wait for a short period and receive any offers. If found, the nearest responding robot to the item is selected and assigned the task, and the original robot can resume its previous behaviour. Robots which have offered help remain in the Wait for Assignment state for a short period before returning to Explore, however if they receive an assignment, they'll enter the Respond state in which they move directly towards the item to forage.

As can be seen in Fig. 8, a robot in the Respond state may find items en route that they are unable to forage, and they will send out help messages of their own before resuming their journey to their assigned item. When they reach the location, they will forage the item if found. If the item has been foraged by another robot in the intervening period, the responding robot will resume exploration.

The APC's ability to share data throughout the swarm presents an opportunity to employ the Blackboard strategy in which each robot maintains a list of known items while following the behaviour shown in Fig. 9. When in the Explore state, before a robot moves to a random adjacent cell, it first checks its knowledge base to see if there is a nearby item of the same type that it may move towards. If so, the robot will enter the Respond state in order to forage that item. During exploration or responding, if a robot finds an item it cannot forage, it will add it to its knowledge base.

To facilitate cooperation, knowledge is periodically broadcast to neighbouring robots, which synchronise the incoming data with their own knowledge. Each item is recorded with its position, type and forage status. By storing the status of an item, a robot is able to inform neighbours when an item has been foraged, thus spreading that information through the swarm, and preventing robots from moving to forage items which no longer exist. To balance

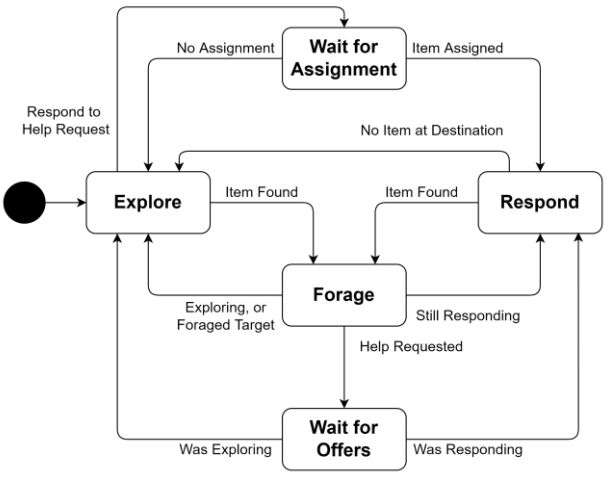


Figure 8. State Machine for the Help Recruitment cooperation strategy for use in the foraging task.

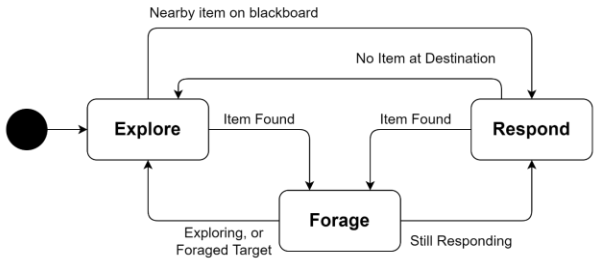


Figure 9. State Machine for the Blackboard strategy for use in the foraging task.

robot behaviour between exploration and responding to known items, a maximum response distance is used, under the assumption that the swarm is better served by ignoring distant items as other swarm members may be better placed.

As reported in [28], engaging in cooperative behaviour allows the swarm to find and remove all items more quickly. Further, for the Help Recruitment strategy to perform at its best, it was found in [9] that an autonomic system that manages the range of each help broadcast can improve performance. The Decentralised Autonomic Manager presented there works by using a fixed pulse range between robots for the purpose of estimating the local density, with the proviso that the appropriate pulse range must be set ahead of time.

In this paper, each robot makes use of the APC system to share their robot type and current position alongside the APC pulse messages. By receiving messages from neighbouring robots, each robot may deduce the numbers of each robot type in their neighbourhood, and subsequently calculate an appropriate density and select a suitable range. As in Section IV, the robots must make use of a relationship between the density of the swarm and the desired pulse range.

When using the DAM, every robot uses the same pulse range, and so it was possible to define the area from which pulses were received by treating it as a circle using the pulse range as the radius. With the APC, each robot may use a different pulse range, so the average distance of all received pulses is used instead.

A. Methods

To test the performance of the APC system, first it is necessary to determine the density-pulse range relationship for the task. To achieve this, swarms of between 16 and 320 robots, rising in 16-robot increments, were tested with fixed help broadcast ranges of 4 units, and 8-64 units, rising in 8-unit increments, and no APC system active.

Following that, swarm sizes of 32, 64, 128 and 256 robots, equally split between the two types, are deployed. Each configuration is run 50 times with a different initial position of robots and items, and the performance is measured as the number of simulation ticks taken to forage all items.

For the Help Recruitment strategy, these tests are carried out with the DAM set with fixed pulses of 8-64 units, rising in 8-unit increments, and again with the APC system, both making use of the density-pulse range relationship to determine a suitable range for help broadcasts. The performance of the APC is then compared against the best performing DAM configuration.

The Blackboard strategy is employed using the APC only, with performance compared against the performance of the Help Recruitment strategy in both DAM and APC configurations. As the Blackboard strategy requires a parameter dictating the maximum range at which a robot responds to a nearby item, this strategy is tested with maximum ranges of 8, 16, 24 and 32 units.

B. Results

Fig. 10 shows the relationship between swarm density and help broadcast range, with a trend line fitted. The resulting density-pulse range relationship for the foraging robots task is:

$$r_{help} = 1.4615 \times \rho^{-0.501}, \quad (3)$$

where r_{help} is the help broadcast range, and ρ is the swarm density calculated using (1).

Table III shows the performance of the two cooperation strategies implemented using the APC system, compared against the best-performing DAM configuration as determined by how quickly each configuration completes the foraging task.

For the Help Recruitment strategy, there is no statistical difference between the results of the DAM and the APC system at $p < 0.05$. The Blackboard strategy, on the other hand, shows a statistical difference with some values for the response range parameter, performing worse than the DAM in those cases. An exception is the case with 256 robots and a response range of 8 units, in which the APC system outperforms the DAM.

VIII. Discussion

The results show that a relationship may be established between the performance of the swarm and the pulse range used for transmitting the data, as seen in Fig. 3. This relationship is specific to the task employed, in this case the sharing of data to at least 99.5% of the swarm within 250 ticks. Different tasks, with different requirements for success,

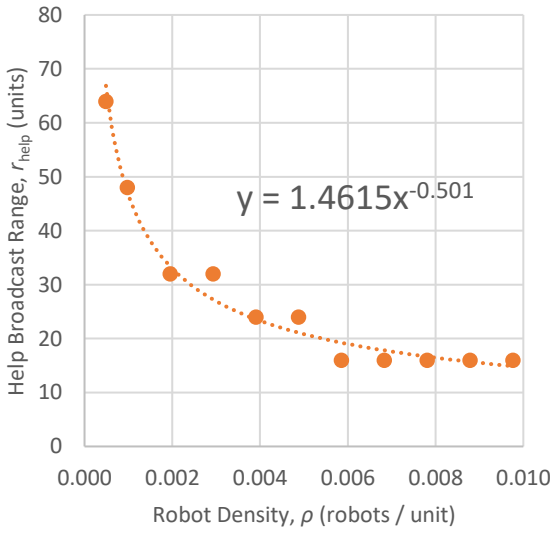


Figure 10. Plot of ideal broadcast range against swarm density, with a best fit trend line, for communications within a swarm of robots engaged in a foraging task.

TABLE III. COMPARISON OF SWARM STRATEGY PERFORMANCE IN A FORAGING TASK USING THE DECENTRALISED AUTONOMIC MANAGER AND AUTONOMIC PULSE COMMUNICATION

Swarm Size	DAM Ticks			APC Ticks			t-statistic	p-value
	Range	Mean	SD	Strategy	Mean	SD		
32	24	9022.78	2692.76	Help	8863.28	2406.95	0.312	0.755
				Board 8	10164.52	2561.04	-2.172	0.032
				Board 16	9291.96	2710.00	-0.498	0.619
				Board 24	9452.50	2780.54	-0.785	0.434
				Board 32	9440.04	3786.79	-0.635	0.527
64	24	4331.44	1121.37	Help	4728.30	1604.44	-1.434	0.155
				Board 8	4799.04	1430.61	-1.819	0.072
				Board 16	4052.30	1278.41	1.161	0.249
				Board 24	4240.56	1130.31	0.404	0.687
				Board 32	4801.76	1429.02	-1.831	0.070
128	16	2073.22	667.62	Help	2127.40	577.34	-0.434	0.665
				Board 8	2354.30	869.60	-1.813	0.073
				Board 16	2076.66	481.86	-0.030	0.976
				Board 24	2333.58	810.23	-1.754	0.083
				Board 32	2414.86	640.94	-2.610	0.010
256	8	876.34	180.73	Help	1059.82	676.66	-1.852	0.067
				Board 8	696.38	239.71	-2.191	0.031
				Board 16	1130.18	371.97	-4.340	< 0.001
				Board 24	1182.20	315.82	-5.944	< 0.001
				Board 32	1477.02	445.47	-8.835	< 0.001

will necessarily result in a different relationship being established.

Increasing the pulse period has a detrimental effect on swarm performance, although it would reduce the energy used as fewer pulses would be sent. Balancing the performance needs of the swarm with the energy cost is an important factor, so a pulse period of 10 ticks was chosen for the adaptive APC and communications loss tests. Halving the period to 5 ticks would double the expected energy usage for only a small gain in performance, as seen in Fig. 4. Any performance decrease from using a longer period can be balanced through pulse range selection in the adaptive APC system.

Fig. 5, showing how the knowledge of the swarm improves with time, indicates that the larger the swarm, the faster the data is shared. As the test uses a fixed pulse range, the results are explained by noting that a larger swarm has a greater density, and so more robots are likely to be reached with each pulse. At the smallest size, some pulses may not be received by any robot. When the swarm contains 200 or fewer robots, it was not able to share data with all members within 250 ticks.

It is the responsibility of the adaptive APC system to address this problem, and the results in Table II show that the system, when starting with an initial pulse range of just one unit, is able to determine an appropriate range for a robot to broadcast at and enable the sharing of the data throughout the swarm within the allotted 250 ticks. This is a large improvement over the performance seen in Fig. 5, where the swarm of 50 robots still hasn't reached that knowledge level after 5,000 ticks.

When comparing the average pulse range in Table II to the best fixed ranges in Table I, the adaptive APC system is found to have a slightly higher range on average in lower density swarms, but in higher density swarms it can reduce the average pulse range, allowing the swarm to expend less energy. In the denser swarms, not every robot will detect the same local density, so the APC system enables the robots to reduce their pulse range while in higher density areas.

The APC system was also found to be extremely robust to communications loss, being able to successfully share the data within 250 ticks even when the probability of a successful message is as low as 5%, and it performs much better than the fixed pulse range at that level. A lower number of pulses being successfully received will result in a lower density estimate being made by the APC system, and a corresponding increase in the pulse range to reach more robots. While this system balances, increasing pulse ranges will increase energy usage.

It may be preferable for the swarm in cases of extremely high message loss to recognise the problem and find an alternative solution, perhaps contracting the swarm or temporarily increasing the period between pulses. Adaptive adjustment of the pulse period may help reduce energy usage overall, and this may be a topic for future work.

With the APC shown to be capable of allowing robots to adjust their pulse range in reaction to the perceived local density of the swarm, the next scenarios investigated the system's use in a foraging task. The results here show that the APC is capable of matching the performance of the DAM when used for the Help Recruitment strategy in the foraging task. By adaptively adjusting the pulse range based on the

density of the swarm, the system does not require prior knowledge of the swarm size, making it useful in situations where the swarm may change due to robot loss, or the addition of reserves.

However, it is not perfect. While not statistically significant, the swarm of 256 robots appears to take longer with the APC system than with the DAM. This may be down to the difficulty in calculating the area around the robot from which the local density is derived. In the DAM, the fixed pulse range may be used as a radius. In the APC system, an average distance approach may be used, changing the density calculation.

The Blackboard, when implemented using the APC system, is also capable of matching performance in some cases, and in one case exceeding it. However, it requires an appropriately configured response range in order to do so, and an incorrect setting may negatively impact performance. This may itself be a candidate for adaptive adjustment based on the environment, using information such as the number of known items and the composition of the swarm.

Further, the Blackboard strategy has much higher data transfer requirements, increasing with every item known rather than the fixed size used for the Help Recruitment strategy. This may be mitigated by limiting the data sent in some way, perhaps using timestamps to favour recent data, or only sharing items nearby. Any advantage conferred by the Blackboard strategy should be balanced against the strategy's requirements.

IX. Conclusion and Future Work

This research presented a system for adaptively adjusting the range of communications between robots based on the density of the swarm, by adapting the existing concept of Pulse Monitoring. By replacing the "I am healthy" message with one saying, "I am here", a receiving robot can use the aggregate data presented by multiple received pulses to estimate the local density of the swarm.

In a task to share a piece of data with the rest of the swarm, the Autonomic Pulse Communications system was able to adaptively determine the pulse range to use to achieve excellent results, ensuring that 100% of the swarm received the data within the allotted time in all but three scenarios. The results show the system selecting shorter pulse ranges when the swarms are denser, and compare favourably with the best performing fixed pulse ranges used to establish the relationship between density and pulse range that the system uses. Further, the APC system was shown to be extremely robust to communications loss, as the system adapts to a decrease in the number of received messages by increasing the pulse range, thus increasing the chances of the message being received by some robots.

The APC system was then used to implement both the Help Recruitment and Blackboard strategies for a swarm of foraging robots. The performance was shown to be comparable to that of the best performing DAM which required a pulse range to be set prior to the mission. The APC lifts that restriction, successfully enabling the swarm to adapt the pulse range according to the measured swarm density.

The APC system therefore shows promise, allowing a swarm to maintain communication links between its members while imposing fewer restrictions on the behaviour of the robots. Should the swarm suffer loss of robots over the course of the mission, the resulting lower density of the swarm may be compensated for automatically by the system.

This work was carried out exclusively using simulation, which may suffer from what is termed the "reality gap" [29], where results obtained in simulation are not replicated when the same experiment is run in reality. The abstract nature of the simulations used here means there are several steps that can be taken to close the gap, however the ideal test environment would use real physical hardware.

Individual pulse messages used in this work were simplified, by considering them to be atomic actions. Larger amounts of data may take longer to broadcast than small packets, and this will impact the ability of a robot to successfully receive all of the data in a single broadcast. The motion of the robots may result in a recipient moving out of range before the transmission is completed. Additionally, communications failure was simply modelled as a random chance of failure, not taking into account the operating conditions or physical obstructions in the path.

Future work may investigate the impact of those aspects on the system, as well as applying the APC system to other tasks such as collective decision-making. Another avenue of interest may be the mechanism by which data is shared. As information grows in complexity, it may be desirable to selectively share only a portion of data in order to minimise the time and energy costs of data transfer, keeping the pulse messages short.

Further work may also investigate the impact of other factors in the ability of the swarm to share data. In this work, the data to be shared was fixed, so a changing data set that requires frequent reporting should be investigated. Also of note is the movement of the swarm, which supports data sharing through changing the set of neighbours receiving a robot's pulse. Different robot speeds, more limited mixing, and the absence of motion altogether may impact the performance of the system.

References

[1] L. McGuigan, R. Sterritt, and G. Hawe, 'Autonomic Pulse Communications for Adaptive Transmission Range in Decentralised Robot Swarms', Sep. 2023, pp. 15–21, Accessed: Jan. 22, 2024. [Online]. Available: https://www.thinkmind.org/index.php?view=article&articleid=emerging_2023_1_30_50021.

[2] E. Sahin, 'Swarm robotics: From sources of inspiration to domains of application', in *Swarm Robotics*, vol. 3342, E. Sahin and W. M. Spears, Eds. 2005, pp. 10–20.

[3] M. G. Hinchey, R. Sterritt, and C. Rouff, 'Swarms and Swarm Intelligence', *Computer*, vol. 40, no. 4, pp. 111–113, Apr. 2007.

[4] V. Trianni, J. IJsselmuiden, and R. Haken, 'The SAGA concept: Swarm Robotics for Agricultural Applications', Technical Report, 2016. Accessed: Dec. 21, 2022. [Online].

Available: http://laral.istc.cnr.it/saga/wp-content/uploads/2016/09/saga-dars2016.pdf.

[5] L. Abraham, S. Biju, F. Biju, J. Jose, R. Kalantri, and S. Rajguru, 'Swarm Robotics in Disaster Management', in *2019 International Conference on Innovative Sustainable Computational Technologies (CISCT)*, Oct. 2019, pp. 1–5.

[6] G. Beni, 'From Swarm Intelligence to Swarm Robotics', in *Swarm Robotics*, Berlin, Heidelberg, 2005, pp. 1–9.

[7] J. O. Kephart and D. M. Chess, 'The vision of autonomic computing', *Computer*, vol. 36, no. 1, pp. 41–50, Jan. 2003.

[8] E. Vassev, R. Sterritt, C. Rouff, and M. Hinchey, 'Swarm Technology at NASA: Building Resilient Systems', *IT Prof.*, vol. 14, no. 2, pp. 36–42, Mar. 2012.

[9] L. McGuigan, R. Sterritt, G. Wilkie, and G. Hawe, 'Decentralised Autonomic Self-Adaptation in a Foraging Robot Swarm', *Int. J. Adv. Intell. Syst.*, vol. 15, no. 1 & 2, pp. 12–23, 2022.

[10] M. Puviani, G. Cabri, and L. Leonardi, 'Enabling Self-Expression: The Use of Roles to Dynamically Change Adaptation Patterns', in *2014 IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems Workshops*, Imperial College, London, United Kingdom, Sep. 2014, pp. 14–19.

[11] F. Zambonelli, N. Bicocchi, G. Cabri, L. Leonardi, and M. Puviani, 'On Self-Adaptation, Self-Expression, and Self-Awareness in Autonomic Service Component Ensembles', in *2011 Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops*, Ann Arbor, MI, USA, Oct. 2011, pp. 108–113.

[12] J. Zelenka, T. Kasanický, and I. Budinská, 'A Self-adapting Method for 3D Environment Exploration Inspired by Swarm Behaviour', in *Advances in Service and Industrial Robotics*, Cham, 2018, pp. 493–502.

[13] C. Saunders, R. Sterritt, and G. Wilkie, 'Autonomic Cooperation Strategies for Robot Swarms', in *Adaptive 2016: The Eighth International Conference on Adaptive and Self-Adaptive Systems and Applications*, Rome, Italy, Mar. 2016, pp. 20–27.

[14] G. Valentini, E. Ferrante, and M. Dorigo, 'The Best-of-n Problem in Robot Swarms: Formalization, State of the Art, and Novel Perspectives', *Front. Robot. AI*, vol. 4, 2017, Accessed: Jan. 03, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2017.00009.

[15] B. Capelli and L. Sabattini, 'Connectivity Maintenance: Global and Optimized approach through Control Barrier Functions', in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 5590–5596.

[16] B. Capelli, H. Fouad, G. Beltrame, and L. Sabattini, 'Decentralized Connectivity Maintenance with Time Delays using Control Barrier Functions', in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 1586–1592.

[17] Y. Kantaros, M. Guo, and M. M. Zavlanos, 'Temporal Logic Task Planning and Intermittent Connectivity Control of Mobile Robot Networks', *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4105–4120, Oct. 2019.

[18] N. Majcherczyk, A. Jayabalan, G. Beltrame, and C. Pinciroli, 'Decentralized Connectivity-Preserving Deployment of Large-Scale Robot Swarms', in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 4295–4302.

[19] V. S. Varadharajan, D. St-Onge, B. Adams, and G. Beltrame, 'Swarm Relays: Distributed Self-Healing Ground-and-Air Connectivity Chains', *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5347–5354, Oct. 2020.

[20] P. Smith, R. Hunjet, A. Aleti, and J. Barca, 'Data Transfer via UAV Swarm Behaviours', *J. Telecommun. Digit. Econ.*, vol. 6, pp. 35–57, Jun. 2018.

[21] P. Gupta and P. R. Kumar, 'The capacity of wireless networks', *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[22] R. Sterritt, 'Pulse monitoring: extending the health-check for the autonomic grid', in *IEEE International Conference on Industrial Informatics, 2003. INDIN 2003. Proceedings*, Banff, AB, Canada, 2003, pp. 433–440.

[23] R. Sterritt and S. Chung, 'Personal autonomic computing self-healing tool', in *Proceedings. 11th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, 2004*, May 2004, pp. 513–520.

[24] R. Sterritt, D. Gunning, A. Meban, and P. Henning, 'Exploring autonomic options in a unified fault management architecture through reflex reactions via pulse monitoring', in *Proceedings. 11th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, 2004*, Brno, Czech Republic, 2004, pp. 449–455.

[25] W. Truszkowski, M. Hinchey, and R. Sterritt, 'Towards an Autonomic Cluster Management System (ACMS) with Reflex Autonomicity: Workshop on Reliability and Autonomic Management in Parallel and Distributed Systems (RAMPDS-05) at ICPADS-2005', *Unkn. Host Publ.*, pp. 478–482, Jul. 2005.

[26] E. Vassev and M. Hinchey, 'Self-Awareness in Autonomous Nano-Technology Swarm Missions', in *2011 Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops*, Ann Arbor, MI, USA, Oct. 2011, pp. 133–136.

[27] K. Støy, 'Using Situated Communication in Distributed Autonomous Mobile Robotics', *SCAI*, vol. 1, pp. 44–52, Feb. 2001.

[28] L. McGuigan, C. Saunders, R. Sterritt, and G. Wilkie, 'Cooperation Strategies for Swarms of Collaborating Robots: Analysis of Time-Stepped and Multi-Threaded Simulations', *Int. J. Adv. Syst. Meas.*, vol. 14, no. 3 & 4, pp. 44–58, 2021.

[29] N. Jakobi, P. Husbands, and I. Harvey, 'Noise and the reality gap: The use of simulation in evolutionary robotics', in *Advances in Artificial Life*, vol. 929, F. Morán, A. Moreno, J. J. Merelo, and P. Chacón, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 704–720.

Optimized Hardware Procurement for High Performance Computing Systems

Scott Hutchison
Department of Computer Science
Kansas State University
Manhattan, KS 66505, USA
email: scotthutch@ksu.edu

Daniel Andresen
Department of Computer Science
Kansas State University
Manhattan, KS 66505, USA
email: dan@ksu.edu

William Hsu
Department of Computer Science
Kansas State University
Manhattan, KS 66505, USA
email: bhsu@ksu.edu

Mitchell Neilsen
Department of Computer Science
Kansas State University
Manhattan, KS 66505, USA
email: neilsen@ksu.edu

Benjamin Parsons
High Performance Computing Modernization Program
Engineering Research and Development Center
Vicksburg, MS 39180, USA
email: ben.s.parsons@erdc.dren.mil

***Abstract*—When faced with upgrading or replacing High Performance Computing or High Throughput Computing systems, system administrators can be overwhelmed by hardware options. Servers come with various configurations of memory, processors, and hardware accelerators, like graphics cards. Differing server capabilities greatly affect their performance and their resulting cost. For a fixed budget, it is often difficult to determine what server package composition will maximize the performance of these systems once they are purchased and installed. This research uses simulation to evaluate the performance of different server packages on a set of jobs, and then trains a machine learning model to predict the performance of un-simulated server package compositions. In addition to being orders of magnitude faster than conducting simulations, this model is used to power a recommender system that provides a precision@50 of 92%. This model is further evaluated using 24 days throughout the calendar year, and it achieves a precision@50 of 88%.**

***Index Terms*—HPC; Procurement Optimization; Recommender system; XGBoost.**

I. Preface

This research was originally presented at The Seventeenth International Conference on Advanced Engineering Computing and Applications in Sciences [1]. The results presented there have been expanded upon and further clarified for this publication.

II. Introduction

When faced with upgrading or expanding a High Performance Computing (HPC) or High Throughput Computing (HTC) system, administrators of these systems can be overwhelmed by options. It is a challenging task to get the best performance for a fixed budget. Server capabilities (i.e., number and types of processors, amount of memory, and number and types of Graphics Processing Units (GPU) or other hardware accelerators) greatly affect their costs, and for a fixed spending ceiling, it is desirable to get the "best bang for your buck." For an HPC system, an optimal server package composition is dictated by its typical use. For instance, if many users rely upon a GPU-accelerated application or library, a higher GPU count may be desirable, even if this means fewer servers can be purchased. With many factors to consider, HPC administrators often rely upon their preferences, intuition, and experience to inform procurement decisions. This research uses historical job data from an HPC system, a discrete event simulator (DES), and a machine learning model to power a recommender system, which can help inform a hardware procurement decision. These techniques provide additional information to HPC system administrators about which set of budget-constrained hardware minimizes wait time for users' jobs, and provides quantifiable support for procurement decisions when upgrading or expanding existing HPC infrastructure. The contributions of this work can be summarized as follows:

1) A data set consisting of roughly 12,700 HPC scheduling simulations, each with a different HPC server set
2) An optimized XGBoost regression model for predicting average wait time when given a composition of servers
3) A recommender system with precision@50=92%, which can inform hardware procurement decisions

This paper is laid out as follows: Section III provides additional background on the problem and describes similar work done by others, Section IV provides the methodology and some implementation details, Section V provides details of formulas for metric calculations, Section VI provides the results of the experiments, and Section VII provides additional details on how the recommendations of the system were evaluated across a wider time frame. Section VIII evaluates the performance of a model trained using all available data, and Section IX provides our final conclusions.

III. Background and Related Works

The Open Science Grid (OSG) [2] [3] is a worldwide collaboration that offers distributed computing for scientific

research. In the central United States, one of the organizations contributing resources to the OSG is the Great Plains Augmented Regional Gateway to the Open Science Grid (GP-ARGO) [4]. In part, GP-ARGO receives funding through governmental grants. These grants are often used to procure new equipment to expand or improve the capabilities of GP-ARGO's participating organizations. Consequentially, there is a fixed budget ceiling for HPC equipment procurement, and the administrator's goal is to purchase new equipment that will maximize computational performance for our typical applications while ensuring costs remain under the fixed grant budget. The research question for this work is as follows: for a planned HPC expansion, can experimental simulation provide an optimal set of hardware under a given budget that will minimize job wait time?

The challenge of optimal hardware procurement is not exclusive to our organization. Similar work was done by Evans et al. [5]. They collected benchmarks for various software applications on different hardware to optimize the ratio of Central Processing Unit (CPU) and GPU architectures for HPC jobs. Their work is similar to ours, but we took a different approach by using a scheduling simulator to evaluate the performance of a set of jobs that were actually submitted to an HPC system. We are solving a very similar problem as Evans et al., but using a different approach to arrive at an optimal hardware configuration.

Other researchers have attempted to optimize for a particular application, such as the work Kutzner et al. [6] did to improve the utilization of GPU nodes when using GROMACs. Although these techniques are not without their merits for HPC systems that run a large number of homogeneous applications, users of the GP-ARGO HPC systems run a wide variety of jobs and applications. A more broad scheduler-based optimization was more appropriate for our application.

Various public HPC workloads exist [7], and have been used by HPC researchers in the past. However, as we are attempting to identify and evaluate new hardware for a specific HPC system, log data from that HPC system was utilized as the workload for this research.

Different scheduling applications like SLURM, HTCondor, or PBS, operate on HPC systems and perform the function of assigning HPC resources to jobs. This job-to-machine-assignment task is as an extension of the online bin packing problem [8]. For the bin packing problem, the goal is to pack a sequence of items with sizes between 0 and 1 into as few bins of size 1 as possible. Each job specifies the resources requested (the object sizes), and each HPC machine has a certain amount of available resources (the bins with their respective sizes). The scheduler is given the task to meet job requirements by assigning them to HPC nodes (pack the objects into the available bins) as efficiently as possible. This is an online problem as new jobs are submitted over time to the scheduler. The best fit bin packing (BFBP) algorithm has been shown by Dosa and Sgall [9] to use at most $\lfloor 1.7OPT \rfloor$ bins, ensuring this algorithm will provide a reasonably close to optimal average wait time when it is used as an HPC job scheduling algorithm.

Algorithm 1 Best Fit Bin Packing Scheduling

```
1: while The simulation is incomplete do
2:     if Some job in the queue can be executed on some machine then
3:         Find the (job, machine) pairing that results in the fewest remaining resources for some machine. Begin executing that job on that machine.
4:     else
5:         Advance simulation time until a new job is submitted or a running job ends, whichever is sooner.
6:         Queue submitted jobs and stop ending jobs.
7:     end if
8: end while
```

Fig. 1. Pseudocode for the best fit bin packing algorithm

Since scheduling algorithms vary between applications, most being highly customizable, and others being proprietary, a discrete event simulator utilizing the BFBP algorithm served as a stand-in for our scheduling application in an attempt to make it more universally applicable. The BFBP scheduling algorithm is described in Figure 1.

Although various HPC simulators have been used for similar research, such as SimGrid [10], GridSim [11], or Alea [12], this experiment needed a simple discrete event simulator using the BFBP scheduler. The simulators mentioned above were either deemed overly complex for our purposes, or they failed to allow for the three limiting resources (memory, CPUs, and GPUs) we were interested in investigating. An HPC scheduler simulator was also considered, such as the Slurm simulator developed at SUNY University in Buffalo [13]. Although this option was investigated further, scaling a job's actual duration from the log data to the new machine once it is assigned to a machine was challenging. As such, a custom discrete event simulator was developed and utilized for this research. The simulator allows for three resource constraints in each machine: memory, CPUs, and GPUs. It is fairly lightweight, fast, and easy to understand.

A significant consideration when evaluating new server hardware is the performance increase newer technology or architectures can provide. Using log data, we know how long a job took on a machine with known hardware. Since the specifications for the new hardware under consideration are also known, the actual duration of the jobs from the historic log data was scaled using base performance of the processor as reported by SPEC CPU2017 benchmark, second quarter, 2023 [14].

Knowing how a particular job performed on one set of hardware and estimating how it will perform on some other hypothetical set of hardware is challenging. Sharkawi et al. [15] successfully used a similar SPEC benchmark to estimate the performance projections of HPC applications. Other researchers, like Wang et al. [16] have pointed out that these

benchmarks fail to account for all the variables affecting job resource utilization and should be avoided. Although CPU performance is not the only factor by that we could have scaled job duration, and perhaps it is not the best factor by which to scale, it worked well for our purposes. The discrete event simulator was implemented such that the scaling factor could be easily changed if other researchers should find a different factor more relevant to their situation.

Various metrics are typically used when evaluating the performance of HPC scheduling algorithms. Some of these are average wait time, HPC utilization, average turnaround time, makespan, throughput, etc. Which metric is used depends on the application and function of the HPC system, and different organizations may value one metric over another. The metric used for this research was average wait time, or the average number of seconds each job spent waiting in the job queue for execution on HPC resources. We presume that the same techniques could be applied by other researchers using a different metric, should they prefer a different one.

This research relied upon a regression model where: given the total CPUs, total memory, and the total GPUs for a composition of servers, the regression model will predict the average jobs wait time for the representative set of jobs. Various regression techniques were tried, but Extreme Gradient Boosting (XGBoost) [17] was the most effective of those tried. XGBoost is a scalable, distributed, gradient-boosted, decision tree machine learning library. It relies upon supervised machine learning, decision trees, ensemble learning, and gradient boosting. Similar to a random forest, multiple decision trees are created for the regression task, and these trees each make predictions of the average wait time given the three inputs (total server package CPUs, memory, and GPUs). The results from the multiple trees are combined via a weighted sum, and they are "boosted" by generatively adding new decision trees. The error of the objective function is minimized by gradient descent during the training process, resulting in quick convergence and accurate prediction results.

Recommender systems power a variety of applications like search engines and music recommendation systems. First, the "hits" for the system must be defined. Hits are the elements from the data set that are relevant to the user's search. Next, the user specifies the number of recommendations, k, that they would like to receive. If the recommender system is precise, a large portion of the k items returned will be hits.

IV. Methodology

The general plan for optimizing a hardware package for our fixed budget can be summarized as follows:

1) Receive vendor quotes with potential server options.
2) Generate potential server combinations to purchase under the specified budget which meet our procurement requirements.
3) Identify a typical set of jobs representing the workloads typically submitted to our HPC system.
4) Conduct simulations using a subset of the server packages to schedule the representative job set and compute metrics to determine their performances.
5) Use machine learning to train and refine a regression model that can predict the performance of un-simulated server combinations.
6) Develop a recommender system using the machine learning model and quantitatively evaluate its performance
7) Subjectively evaluate the recommended server packages and make a more informed procurement decision.

This pipeline is illustrated by Figure 2. First, we generate all possible combinations of servers we can purchase under our budget. Next, we uniformly sample 10% of these by selecting every tenth server combination and we use the DES to simulate the execution of a chosen set of jobs. We then use XGBoost to develop and train a regression model that will map a sever package's total CPUs, memory, and GPUs to the predicted average wait time that these jobs will experience. We use the regression model to predict the average wait time of the sampled server combinations, sort them by the predicted wait time, and return the top k recommended server sets to the user. These recommendations can be quantitatively evaluated, as the actual average wait time has been simulated. Next, we can use the same regression model and recommender system to make predictions on the entire set of servers, and we can summarize them and subjectively evaluate them, as 90% of the have not be simulated and their actual average job wait time is unknown.

Finally, the recommendations of the system were simulated using workloads from 24 days across a calendar year to determine it the recommended server sets continued to be effective when faced with the varied workloads the HPC system experienced throughout the year.

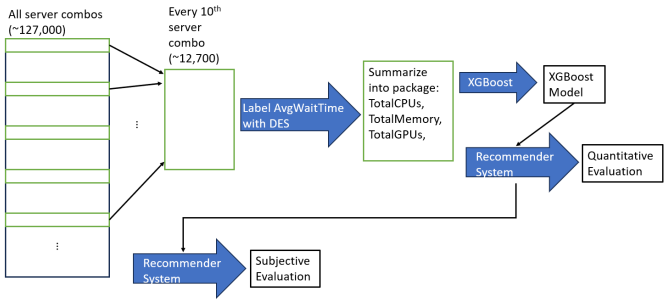


Fig. 2. A pictorial representation of the methodology for this research.

A. Generate Server Options

To begin, we received several vendor quotes specifying the costs and capabilities of 21 potential servers to purchase. When considering upgrade options, we typically separate servers into one of three categories: compute nodes, big memory nodes, or GPU nodes. A compute node typically has a large number of processor cores, a moderate amount of memory, and no GPU. A big memory node will have a large amount of

TABLE I. SERVER CAPABILITIES AND COSTS UNDER INVESTIGATION

Node type	*Distinct nodes consid-ered*	*Memory range per node*	*CPUs range per node*	*GPUs per node*	*Cost range per node*
Compute	4	256-512 Gb	24-64 cores	0 GPUs	$6k-$10k
Big memory	2	1024 Gb	24-64 cores	0 GPUs	$11k-$13k
GPU	15	256-1024 Gb	24-64 cores	1-8 GPUs	$14k-$100k

memory with a moderate amount of CPU cores and no GPU. A GPU node is any node that has a GPU. Table I lays out the options we received from several different vendors. The procurement budget was fixed at $1 million, and all possible server combinations were generated in the following way:

- Separate servers into three categories: compute nodes, big memory nodes, and GPU nodes.
- Choose all combinations of one node from each category.
- Determine all quantities of the three node types under a given budget such that there is at least one GPU node and there is not enough funding remaining to purchase another node.

In our selected job set, many jobs requested GPUs as a resource. These jobs would automatically fail if at least one GPU node were not included in a potential server package. Roughly 127,000 different server combinations met these requirements. Table II provides an illustrative example of how the server combinations were generated. Many server options and packages were omitted from the table for the sake of brevity.

B. Identify a Representative Set of Jobs

One typical days' worth of submitted jobs (roughly 16,000 jobs) was subjectively pulled from the log data of the local HPC system. As with most HPC systems, jobs were submitted in a bursty manner, and variety of resources were requested. Figure 3 and Table III display some descriptive statistics and information about the jobs used for this portion of this research.

C. Job Duration Scaling

The submitted jobs were scaled using the base performance of the processor on the SPEC CPU2017 benchmark suite. The requested duration was not modified, but the actual duration of each job was calculated using the following formula:

$$\text{New duration} = \frac{\text{logged duration} * \text{logged processor performance}}{\text{new processor performance}}$$

D. Discrete Event Simulator

Since there are many different applications for scheduling jobs on HPC systems, the discrete event simulator using the BFBP scheduling algorithm acted as a generic substitute for the scheduling application for our HPC system. What was needed was a method for determining the average job wait

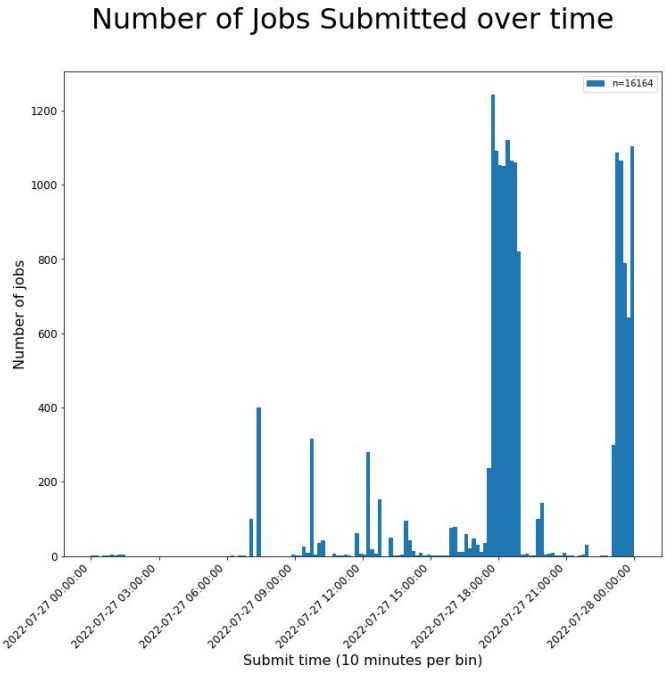


Fig. 3. The number of jobs submitted over time for the selected day

time for selected jobs on specified hardware. Other simulators could have been used, but for this research, a discrete event simulator was implemented in Python that provides the following functionality:

- A global clock to keep track of simulation time.
- Several queues, priority queues, or lists to track jobs as they progress through the execution process: future jobs, queued jobs, running jobs, completed jobs, and unrunnable jobs.
- Jobs and machines are specified using comma separated value (csv) files, which is loaded prior to the simulation.
- Machines have three limiting resources: available memory, CPUs, and GPUs.
- Jobs are specified with the following attributes: submit time, actual duration, and requested duration, memory, CPUs, and GPUs. Jobs track their start time and end time as the simulation progresses to allow for metric calculation.
- Job end time is set when the job starts running as the job start time plus the job actual duration.
- When a job starts running on a machine, that machine's available resources are decremented by the resources requested by the job. Conversely, when a job completes, the machine executing it has its available resources increased by the amount requested by the ending job.
- Jobs with a submit time greater than the current global clock reside in the future jobs priority queue.
- Jobs with a submit time less than or equal to the current global clock, but not yet assigned to a machine, reside in the job queue.
- Jobs that have begun their execution and have an ending

TABLE II. GENERATED SERVER COMBINATIONS

ComputeNode1, $6,960 ea.	*BigMemNode1, $11,112 ea.*	*GPUNode1, $14,730 ea.*	...	*Package Cost*	*Funds Remaining*
141	0	1	...	$996,090	$3,910
139	1	1	...	$993,282	$6,718
138	2	1	...	$997,434	$2,566
⋮	⋮	⋮	⋱	⋮	⋮
0	1	67	...	$998,022	$1,978

TABLE III. DESCRIPTIVE STATISTICS FOR THE POOL OF SELECTED JOBS

	Requested Mem (in Gb)	*Requested CPUs*	*Requested GPUs*	*Requested Duration (in hours)*	*Actual Duration (in hours)*
Mean	5.12	4.75	0.002	2.82	2.27
Std Dev.	16.73	3.33	0.055	1.02	13.67
Min	1	1	0	0	0
Max	800	64	4	11.20	11.20

time less than the current global clock, reside in the running jobs priority queue.

- Jobs with an ending time less than or equal to the current global clock reside in the completed jobs list.
- If no node in the cluster has adequate resources to run a particular job, that job is moved to the unrunnable jobs list.
- In the event that no queued jobs can run on available resources, the simulation time "fast forwards" to the next event: either job submission or job ending.
- Jobs in the job queue are run as soon as there are available resources and are chosen using the best fit bin packing scheduling algorithm described in Algorithm 1.
- Actual job duration from logged job data can be scaled to allow for hardware improvement with newer hardware.

E. Machine Learning

Although each simulation completed fairly quickly, requiring no more than 30 minutes each, this particular combination of server quotes yielded roughly 127,000 combinations that need to be evaluated. To reduce the computational requirement, every tenth line from the file with the server combinations was sampled, and roughly 12,700 simulations for these server packages were completed in parallel using HPC resources. By sampling from the generated server packages uniformly, various quantities of each server under consideration were included in the simulated data. Each server package was summarized into the package total memory, total CPUs, and total GPUs, by summing the resources of every machine comprising the package. The average wait time for the simulation served as the label for each package. Using five fold cross validation, an XGBoost regression model was trained using training data. The regression model was evaluated using root mean squared error (RMSE) on the test data. An accurate regression model enabled the prediction of the average wait time for unsimulated server combinations and saved countless hours of additional simulation.

F. Recommender System

In our case, a hit was defined as a server combination with an average wait time in the lowest 5% of simulated combinations (or 632 hits out of the ∼12,700 simulated server combinations). The value of k was varied to evaluate the performance of the recommender system. Then, once confidence was gained that our recommender system was functioning properly, it was used to recommend systems from the entire server combination pool of 127,000 server combinations. The recommendations were summarized and evaluated subjectively before arriving at a final procurement decision.

G. Simplifying Assumptions

The current nodes comprising the HPC system were not added to the set of nodes simulating the selected jobs. The benefit current nodes would provide to the new servers under investigation would be common to all.

Any additional equipment required to install and operate the new servers (e.g., networking hardware, additional cooling equipment, server racks, power infrastructure, etc.) were not deducted from the total procurement budget. It was thought that these costs would be a relatively fixed regardless of the server package chosen. The same analysis described in this research could be done by reducing the total budget by the cost of additional hardware and then completing the analysis with a reduced budget.

V. EVALUATION

Pearson's Correlation Coefficient [18] determined the extent of the correlation between the total memory, CPUs, and GPUs of a package and the average wait time. This coefficient provides a value between -1 and 1, where values closer to -1 or 1 indicate that the feature and the label are more strongly correlated. A coefficient of 0 indicates no correlation.

Wait time was calculated by analyzing the completed jobs output from each simulation. The wait time for each job was the number of seconds from the time the job was submitted until it began. For N jobs, the average wait time was calculated as follows:

$$\text{AvgWaitTime} = \frac{\Sigma_{i=0}^{N}(\text{Start Time}_i - \text{Submit Time}_i)}{N}$$

Root Mean Squared Error was utilized for regression model evaluation calculated according to the following formula:

$$\text{RMSE} = \sqrt{\frac{\Sigma_{i=0}^{N}(\text{actual wait time}_i - \text{predicted wait time}_i)^2}{N}}$$

The performance of the final recommender system was evaluated using precision@k, recall@k, and F1@k. In general, precision@k is the proportion of recommended items in the top-k set that are relevant, and recall@k is the proportion of relevant items found in the top-k recommendations. F1@k is the harmonic mean of precision@k and recall@k, which simplifies them into a single metric. They were calculated according to the following formulas:

$$\text{Precision@k} = \frac{(\text{\# of recommended items @k that are relevant})}{(\text{\# of recommended items @k})}$$

$$\text{Recall@k} = \frac{(\text{\# of recommended items @k that are relevant})}{(\text{total \# of relevant items})}$$

$$\text{F1@k} = \frac{(2 * \text{precision@k} * \text{recall@k})}{(\text{precision@k} + \text{recall@k})}$$

VI. Results

The correlation of features, the performance of the regression model and the recommender system, and some analysis about the recommended server compositions are described below.

A. Feature Correlation

The correlation between the features and the labels is shown in Table IV. For this set of jobs, the total CPUs in a server package were most strongly correlated to the average wait time. For the chosen jobs, the more CPUs a package had, the lower its average wait time.

Since we are constrained by our available budget of $1 million, choosing to buy one type of node over another is a zero-sum game. The more GPU nodes we purchase, and the more GPUs there are per node, the fewer compute nodes or big memory nodes we are able to afford. This is indicated by the positive correlation between GPUs and the average wait time.

TABLE IV. Pearson Correlation Coefficients

	TotalMem	*TotalCPUs*	*TotalGPUs*	*AvgWaitTime*
TotalMem	1.00	0.14	−0.54	−0.23
TotalCPUs	0.14	1.00	−0.42	−0.70
TotalGPUs	−0.545	−0.42	1.00	0.44
AvgWaitTime	−0.23	−0.70	0.44	1.00

B. Regression Model

The XGBoost regression model had a RMSE = 150.13 seconds, indicating that the total memory, CPUs, and GPU features made excellent predictors for the average wait time for these jobs when simulated with the discrete event simulator. The predicted vs. simulated wait time is shown in Figure 4. If the regression model were perfect, all these points would lie upon the $y = x$ line, and it is clear that this model does a good job at predicting the average wait time for a given composition of servers.

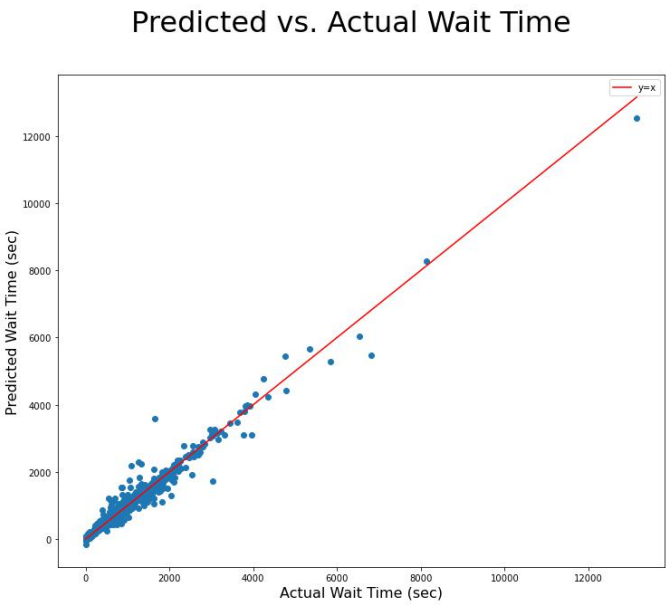


Fig. 4. The predicted vs. simulated wait times showing the accuracy of our regression model.

C. Recommender System

The regression model was used to predict the 12,700 labeled simulations, and their precision@k, recall@k, and F1@k for various values of k are displayed in Table V. The goal was to reduce the number of possibilities from roughly 127,000 different possible combinations of servers down to a reasonable number that could be evaluated by an HPC system administrator and have a large percentage of the recommended server combinations be hits (among the best 5% of server combinations with the lowest average wait times). Although precision@10 was 100%, it is thought that seeing more server package options would allow system administrators a wider variety from which to choose. A system administrator could easily and quickly review up to 50 recommendations ($k = 50$), and more than 46 out of 50 of these recommendations returned by this system (92%) would be top performing server combinations, which is excellent. Recall@k when k is less than the number of total hits (632 hits total) is unfairly penalized, but the recall@k above 632 is also excellent. When $k = 1,000$, the recall@1000 = 91%, meaning the recommender system successfully retrieved 91% of the top 5% performing server packages when returning less than 1% of the 127,000 different options.

TABLE V. Precision@k and Recall@k for Test Data

k value	*Precision@k*	*Recall@k*	*F1@k*
10	1.00	0.02	0.03
50	0.92	0.07	0.13
100	0.81	0.13	0.22
500	0.74	0.59	0.66
632	0.72	0.72	0.72
1000	0.58	0.91	0.71

D. Recommended Compositions

Beyond looking at the individual server compositions recommended, we wanted to draw some conclusion about the types and quantity of nodes that the recommender system returned. The sum of the server quantities for the top 50 recommendations can be found in Table VI. Compute nodes with the larger number of cores were vastly preferred, and the recommender system did not recommend spending additional funds on more memory for the compute nodes. Additionally, the recommender system preferred the cheaper big memory node with fewer cores. Finally, for our typical workload, the recommender system did not recommended purchasing a large number of GPUs per GPU node, instead recommending servers with 2 GPUs per server most often. As shown in Figure 5, the recommender system suggests spending on average 58% of our total budget on compute nodes, 8% on big memory nodes, and 34% on GPU nodes.

TABLE VI. RECOMMENDATIONS DRAWN FROM MODEL PREDICTED RESULTS

Node Type	*Node Description*	*Sum of Servers Across Top 50*
Compute Nodes	Low Cost CPU w/ 256Gb	232
	Low Cost CPU w/ 512Gb	0
	High Cost CPU w/ 256Gb	3,467
	High Cost CPU w/ 512Gb	0
Big Memory Nodes	Low Cost CPU w/ 1024Gb	232
	High Cost CPU w/ 1024Gb	111
GPU Nodes	2 GPUs in one server	732
	4 GPUs in one server	267
	8 GPUs in one server	0

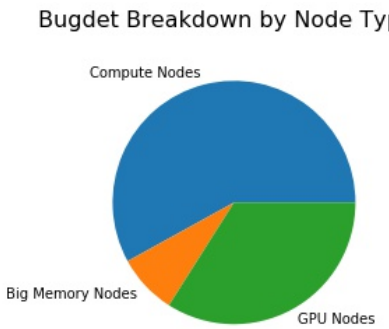


Fig. 5. The recommended budget breakdown by node type.

A boxplot showing the simulated average job wait time of the top 5% of recommended server sets (or k=638) is shown in Figure 6. It is clear that the recommender system was able to retrieve and recommend server sets that performed well on the representative set of jobs.

VII. EVALUATION OF THE GENERALIZATION OF THE APPROACH

The previously described regression model was developed using the results when using a single, representative workload

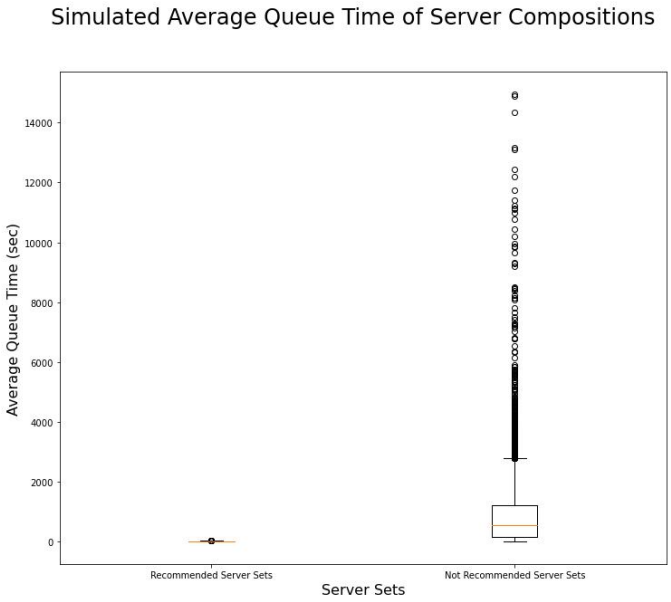


Fig. 6. The average job wait time for the top 5% recommended server sets vs. the bottom 95%.

of one days' worth of jobs from the local HPC system. For this technique to be viable, it must be demonstrated that the recommended server packages would perform well not only for the representative day, but also for many different days of HPC activity. To investigate this further, the following methodology was used:

- Subjectively pull log data an additional 24 days across the year (2 days per month)
- Use the discrete event simulator to simulate the execution of the workloads for the same subset of 12,700 server packages
- Identify the top 10% of server compositions with the lowest average wait time for each day
- Evaluate the performance of the initial recommended server sets using the additional labeled data

This computation was done in parallel using HPC resources and involved over 150,000 CPU hours.

A. Generalized Results

To begin, 24 different days of HPC log data from throughout the year were chosen subjectively (2 days per month). These days were scheduled using the DES using the roughly 12,700 server combinations that were originally used to train the regression model. See Figure 7 for the average wait times for each of the days simulated. Since the job characteristics for each day were different, this caused a re-ordering of the "hits" for each day. For instance, if a day had many GPU jobs, server combinations with more GPUs would have lower average job wait times. Each day's hits were defined as the server set whose average job wait time was in the lowest 10% for that day. By counting the number of days across the year for which that server combination was in the top

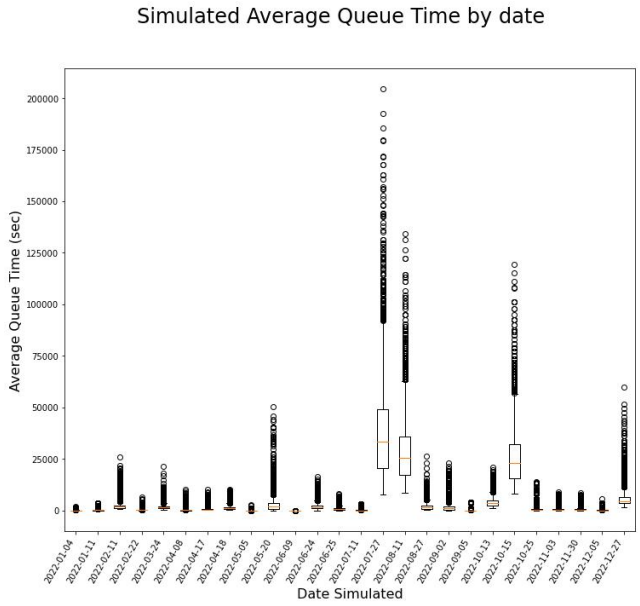


Fig. 7. Boxplots of the average wait time for each of the days simulated throughout the year.

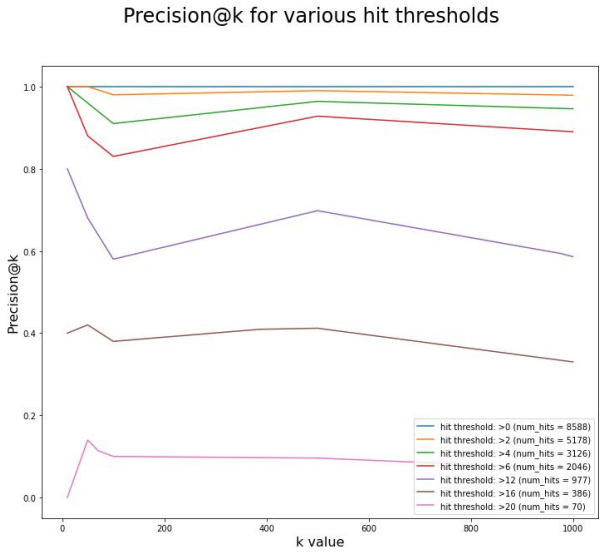


Fig. 8. The precision@k as the hit threshold is varied.

10% of performant server combinations, we were able to determine which server combinations were an overall "hit" for the recommender system. A hit threshold of *num_hits*>6 was found to produce good results, meaning that for 7 or more days of the 25 labeled days throughout the year, the server combination was in the top 10% of performant server packages. The results can be found in Table VII.

TABLE VII. PRECISION@K AND RECALL@K WHEN HIT THRESHOLD >6

Hit Threshold	k	precision@k	recall@k	F1@k
>6	10	1.00	0.005	0.01
>6	50	0.88	0.02	0.04
>6	100	0.83	0.04	0.08
>6	500	0.93	0.23	0.36
>6	1000	0.89	0.43	0.58
>6	2046	0.75	0.75	0.75
>6	5000	0.41	1.00	0.58
>6	10000	0.2	1.00	0.34

If $k = 50$, the recommender system achieves a precision@50=88%, which is slightly lower than the precision@50=92% when the day was evaluated on the same day on which it was trained. Again, 50 recommendations is thought the be an easily human parsable amount which can be compared and evaluated by system administrators for purchase. In other words, given the top 50 recommendations returned to the user, 88% of them would be in the top 10% of performant server sets for 7 out of the 15 days throughout the year which were evaluated.

Increasing the hit threshold reduces the number of total hits that the recommender system can find, and consequentially lowers the precision@k. For instance, the threshold mentioned in Table VII required a server set to be among the top 10% for seven or more days out of the 25 days simulated. In this case, there were 2,046 overall "hits" out of the ~12,700 total server sets whose performance was measured. If the hit threshold is raised to 20, meaning 21 or more days found these server combinations in the top 10% of performing server sets, there are only 70 total hits for the recommender system to find. Figure 8 shows how precision@k degrades when the hit threshold is raised. When the hit threshold is raised to 20, the recall@500 is 67%, meaning that the top 500 results returned by recommender system contained 67% of the 70 hits that were found. Table VIII shows the results at this hit threshold. Though these results are less promising, there were relatively few server sets that performed well for this many days throughout they year. Using the recommender system trained on days' worth of representative jobs saved approximately 150,000 hours worth of computation time conducting the simulations on the various jobs submitted throughout the year.

TABLE VIII. PRECISION@K AND RECALL@K WHEN HIT THRESHOLD >20

Hit Threshold	k	precision@k	recall@k	F1@k
>20	10	0.00	0.00	0.00
>20	50	0.14	0.10	0.12
>20	70	0.11	0.11	0.11
>20	100	0.10	0.14	0.12
>20	500	0.10	0.67	0.17
>20	1000	0.07	1.00	0.13
>20	5000	0.01	1.00	0.03
>20	10000	0.01	1.00	0.01

B. Time Savings for this Technique

Each simulation took around 30 minutes to complete, but they were conducted in parallel using HPC resources. The roughly 12,700 server compositions simulated to train the

regression model took over 6,000 compute hours to complete. Each of the 24 additional days took another 6,000+ hours, for a total of over 150,000 compute hours required to validate the recommendations across a representative sample throughout the year. An exhaustive search of all 127,000 server combinations for a single representative days' worth of jobs would have taken over 60,000 compute hours, and would have yielded a definitive answer on which server combination would have performed best on a single representative days' worth of jobs. Validating this across 24 days across a calendar year would have required over 1.5 million compute hours on HPC resources. The recommender system built using regression from a subset of the possible servers required a 99.96% decrease in the time required for computation while still achieving a precision@50 of 92%. This model achieved a precision@50 of 88% when using the threshold that for 6 or greater days, the server compositions had the lowest 10% average job wait time for each day. The additional validation step of computing 24 days across the year could even be omitted, as the results from the original trained model did quite well across the year.

VIII. Training with All Data

Though we have shown it is sufficient to train using a single day's worth of representative jobs, we obtained simulated average wait times for jobs for 25 days throughout the year. We wanted to explore the performance of a recommender system trained on all data gathered and compare and contrast its performance with the recommender system described above. For each of the 25 days simulated, the average wait time varied depending on the jobs which were submitted on those days. Figure 7 shows boxplots of the average wait time by day depicting the these variations. As such, these values were scaled using min-max normalization prior to regression using the following formula:

$$Normalized\_value = \frac{(actual\_value - min\_value)}{(max\_value - min\_value)}$$

Performing this normalization across each day transforms the average wait time values into a a unitless value between 0 and 1 where values closer to zero represent the best performing server sets with the lowest average wait time. Though the predictions by the regression model will no longer predict the number of seconds of average wait time a server compositions is expected to to have, predicted lower values still represent server packages with lower expected average wait time for jobs. The regression model was not as accurate as when training using a single representative set of jobs, as depicted in Figure 9. Again, if the regression model were perfect, all the predicted vs. actual values would lie upon the $y = x$ line of the graph. Though this model using normalization does not appear to be as good as the previous one, some loss of precision is expected when normalizing in this manner. We can still use the regression model to power a recommender system and evaluate its performance.

Table IX shows the results of the normalized model when the hit threshold is greater than 16. Using the same $k = 50$ value from before, we achieve a precision@50 of 94%, which

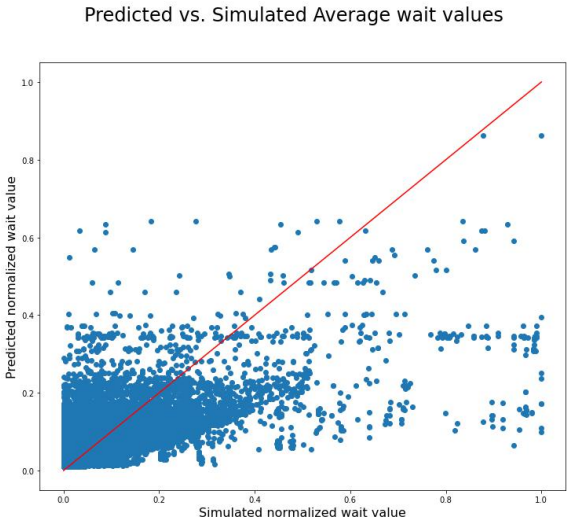


Fig. 9. The predicted vs. simulated normalized average wait time values.

is thought to be excellent. In other words, 47 out of the top 50 recommendations returned by this model will be among the top 10% of performant server combinations for 17 or more days throughout the year. Though this model does slightly better than the model trained on one representative days' worth of jobs, it took 24 times more computation to provide the data to train it. Though the original model trained using a single day achieved a lower precision, it was still able to return good results across the year and required substantially less computation time.

TABLE IX. Precision@k and Recall@k when Hit Threshold >16 for Normalized Model

Hit Threshold	k	precision@k	recall@k	F1@k
>16	10	1.00	0.026	0.05
>16	50	0.94	0.12	0.22
>16	100	0.78	0.20	0.32
>16	386	0.61	0.61	0.61
>16	500	0.54	0.70	0.61
>16	1000	0.32	0.83	0.46
>16	5000	0.08	1.00	0.14
>16	10000	0.04	1.00	0.07

The results of summing the top 50 recommended server sets can be found in Table X. These differ slightly from the recommendations of the model trained using a single representative days' worth of jobs. In both models, the more expensive compute node with more cores was preferred, however the model trained on all the days prefers the compute node with more memory. The model trained on a single day preferred the cheaper big memory node, and the model trained on all the days preferred the more expensive one. Both models preferred GPU nodes with fewer GPUs. Though they differ slightly in the nodes types and quantities they prefer, they are fairly close with their recommendations, and it is thought that the original model using only a single representative day's worth of jobs would provide adequate enough recommendations for a HPC

system administrator to evaluate and arrive at a good server combination to purchase.

TABLE X. RECOMMENDATIONS DRAWN FROM NORMALIZED MODEL PREDICTED RESULTS

Node Type	*Node Description*	*Sum of Servers Across Top 50*
Compute Nodes	Low Cost CPU w/ 256Gb	0
	Low Cost CPU w/ 512Gb	0
	High Cost CPU w/ 256Gb	1,667
	High Cost CPU w/ 512Gb	1,764
Big Memory Nodes	Low Cost CPU w/ 1024Gb	34
	High Cost CPU w/ 1024Gb	751
GPU Nodes	2 GPUs in one server	341
	4 GPUs in one server	108
	8 GPUs in one server	48

IX. CONCLUSIONS

We began with roughly 127,000 different possible server packages that could have been purchased under our budget. We uniformly sampled 10% of these, leaving us with roughly 12,700 different server packages. We chose a representative days worth of jobs from local HPC log data, and simulated the scheduling of these jobs on the 12,700 server packages, so we could calculate the average jobs wait time for a given composition of servers. By developing an XGBoost regression model, we were able to predict the average job wait time for the unsimulated server compositions. Finally, we were able to verify that the recommender system made good recommendations by choosing different sets of jobs spaced throughout the year and simulating the scheduling of different job workloads using those server sets. An administrator considering purchase options has an 88% chance of selecting a top performing server packaged under their budget if they were to choose one from the top 50 recommendations returned by the recommender system. By simulating the performance of a small minority of server packages, our recommender system was able to make excellent recommendations.

The most benefit from using this system comes from the time saved doing simulations. The roughly 127,000 initial server combinations could be effectively summarized by simulating only 10% of them on a single representative set of jobs, and it proved unnecessary to conduct simulations for days spaced throughout the year. This was done for the research in order to evaluate the performance of the recommender system, and explore its feasibility when used to optimize hardware procurement for HPC systems.

This recommender system is not intended to replace the expertise of HPC administrators when it comes to decisions for hardware procurement. It is our hope that this tool can provide a data-driven technique that will help narrow the search space with which administrators are confronted when they make procurement decisions. Returning to the research question: experimental simulation coupled with a regression model enabled a recommender system to return server compositions under a given budget with low average wait times with a precision@50 of 92%. Additionally, the discrete event simulator, job data set, machine learning code, and recommender system code are released under the GPLv3 license should other researchers find it useful (https://github.com/shutchison/Optimal-Hardware-Procurement-for-a-HPC-Expansion).

REFERENCES

[1] S. Hutchison, D. Andresen, W. Hsu, M. Neilsen, and B. Parsons, "Optimized hardware configuration for high performance computing systems," in *Proceedings of the 17th International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2023)*, International Academy, Research, and Industry Association, 2023.

[2] R. Pordes *et al.*, "The open science grid," in *J. Phys. Conf. Ser.*, vol. 78 of *78*, p. 012057, 2007.

[3] I. Sfiligoi, D. C. Bradley, B. Holzman, P. Mhashilkar, S. Padhi, and F. Wurthwein, "The pilot way to grid resources using glideinwms," in *2009 WRI World Congress on Computer Science and Information Engineering*, vol. 2 of *2*, pp. 428–432, 2009.

[4] "The great plains augmented regional gateway to the open science grid." https://gp-argo.greatplains.net/. Accessed 2023-01-18.

[5] R. T. Evans, M. Cawood, S. L. Harrell, L. Huang, S. Liu, C.-Y. Lu, A. Ruhela, Y. Wang, and Z. Zhang, "Optimizing gpu-enhanced hpc system and cloud procurements for scientific workloads," in *International Conference on High Performance Computing*, pp. 313–331, Springer, 2021.

[6] C. Kutzner, S. Páll, M. Fechner, A. Esztermann, B. L. de Groot, and H. Grubmüller, "More bang for your buck: Improved use of gpu nodes for gromacs 2018," *Journal of Computational Chemistry*, vol. 40, no. 27, pp. 2418–2431, 2019.

[7] D. G. Feitelson, D. Tsafrir, and D. Krakov, "Experience with using the parallel workloads archive," *Journal of Parallel and Distributed Computing*, vol. 74, no. 10, pp. 2967–2982, 2014.

[8] S. Martello and P. Toth, *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., 1990.

[9] G. Dósa and J. Sgall, "Optimal analysis of best fit bin packing," in *Automata, Languages, and Programming: 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I 41*, pp. 429–441, Springer, 2014.

[10] H. Casanova, A. Giersch, A. Legrand, M. Quinson, and F. Suter, "Versatile, scalable, and accurate simulation of distributed applications and platforms," *Journal of Parallel and Distributed Computing*, vol. 74, pp. 2899–2917, June 2014.

[11] R. Buyya and M. Murshed, "Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing," *Concurrency and Computation: Practice and Experience*, vol. 14, no. 13-15, pp. 1175–1220, 2002.

[12] D. Klusáček, M. Soysal, and F. Suter, "Alea–complex job scheduling simulator," in *Parallel Processing and Applied Mathematics: 13th International Conference, PPAM 2019, Bialystok, Poland, September 8–11, 2019, Revised Selected Papers, Part II 13*, pp. 217–229, Springer, 2020.

[13] N. A. Simakov, R. L. DeLeon, M. D. Innus, M. D. Jones, J. P. White, S. M. Gallo, A. K. Patra, and T. R. Furlani, "Slurm simulator: Improving slurm scheduler performance on large hpc systems by utilization of multiple controllers and node sharing," in *Proceedings of the Practice and Experience on Advanced Research Computing*, pp. 1–8, 2018.

[14] "Second quarter 2023 spec cpu2017 results," 2023. https://www.spec.org/cpu2017/results/res2023q2, Accessed on June 14, 2023.

[15] S. Sharkawi, D. Desota, R. Panda, R. Indukuru, S. Stevens, V. Taylor, and X. Wu, "Performance projection of hpc applications using spec cfp2006 benchmarks," in *2009 IEEE International Symposium on Parallel & Distributed Processing*, pp. 1–12, IEEE, 2009.

[16] Y. Wang, V. Lee, G.-Y. Wei, and D. Brooks, "Predicting new workload or cpu performance by analyzing public datasets," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 15, no. 4, pp. 1–21, 2019.

[17] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

[18] K. Pearson, "Note on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London*, vol. 58, no. 347-352, pp. 240–242, 1895.

Network Experimental Workflow Leveraging MDE and LLM: Case Study of Wireless System Performance in an α-μ Fading Environment with Selection Diversity Receiver

Dragana Krstic
University of Nis, Faculty of Electronic Engineering
Nis, Serbia
Email: dragana.krstic@elfak.ni.ac.rs

Suad Suljovic
Academy of Applied Technical Studies Belgrade
Belgrade, Serbia
Email: ssuljovic@atssb.edu.rs

Nenad Petrovic
University of Nis, Faculty of Electronic Engineering
Nis, Serbia
Email: nenad.petrovic@elfak.ni.ac.rs

Goran Djordjevic
Academy of Applied Technical Studies Belgrade
Belgrade, Serbia
Email: gdjordjevic@atssb.edu.rs

Devendra S. Gurjar
Department of Electronics and Communications Engineering, National Institute of Technology Silchar
Assam, India
Email: dsgurjar@ece.nits.ac.in

Suneel Yadav
Department of Electronics and Communication Engineering, Indian Institute of Information Technology Allahabad
Email: suneel@iiita.ac.in

***Abstract*—In this paper, a wireless system in the presence of α-μ fading and Co-Channel Interference (CCI) is observed. CCI, a sort of congestion in wireless systems, often has the same distribution as the fading in the observed environment. The α−μ distribution used here is a common model for small-scale fading of THz links. We used diversity receiver with Selection Combining (SC) to mitigate these adverse effects of fading and CCI. For wireless system configured with SC receiver, we derived the Average Bit Error Probability (ABEP) based on the Moment Generating Function (MGF), the level crossing rate (LCR), the average fade duration (AFD), and the channel capacity (CC). The analytical results are presented in a greater number of graphics to highlight the parameters' influence of fading and CCI. Additionally, we propose a workflow for convenient network planning leveraging the synergy of Large Language Models (LLMs) and model-driven engineering (MDE) approach, making use of the previously derived expressions within the evaluation scenario.**

Keywords- α-μ fading; Co-Channel Interference (CCI); Large Language Model (LLM); Model-driven engineering (MDE); Selection Combining (SC).

I. INTRODUCTION

Among the most critical disturbances of signal propagation in wireless channels is fading. Describing and modeling of wireless channels in the presence of fading is of particular importance as for designing the transmission system itself, as well as for the performance analysis. During the development of wireless communications, a large number of different channel fading distribution models have been defined to describe correctly the statistical characteristics of the amplitude and phase of the propagated signal. In the last few years, a general fading distributions are the most popular because other known distributions can be obtained from them. Between them are: α-μ, κ-μ, η-μ, α-κ-μ, α-η-μ, λ-μ, etc. [1]-[7].

In this work, the α-μ distribution is introduced to model a small-scale fading. Usage of α−μ distribution is a common model for small-scale fading of THz links. With this distribution, the nonlinearity of the propagation medium is included since the premise of homogeneity is unrealistic and only approximates the actual transmission medium [2]. As said, α-μ distribution is general distribution. This is generalized Gamma distribution that includes other distributions as are: Gamma (with Erlang as its discrete versions, and also central Chi-squared), Nakagami-*m* (with its discrete version- Chi distribution), Rayleigh, exponential, Weibull, and One-sided Gaussian [3]. That is why it is suitable for an analysis of the performance of wireless systems in the presence of the listed types of fading. The performance obtained for α-μ fading can be reduced to special cases of those fading's distributions obtained for specified values of parameters α and μ.

There are still not many works in the literature that consider this fading distribution, although it is very suitable. Some of them are [8] - [13].

In [8], the Moment Generating Function (MGF) for the Probability Density Function (PDF) of an α-μ wireless fading channel is evaluated for non-integer values of α. By dint of the MGF, the Bit Error Rate (BER) for different modulation techniques is derived. Also, formula for the outage probability (Pout) in the closed form is obtained. All obtained expressions can be reduced to the special cases of Nakagami-*m*, Rayleigh, and Weibull fading channels. The same authors in [9] derived expressions for the amount of

fading and the average channel capacity (CC) for α-μ wireless fading channel.

In [10], the authors proposed a novel MGF for α-μ fading distribution valid for all values of parameter α, as an improvement of [8]. Then, the BER expressions in closed-form are derived for different modulation techniques such as Binary Phase-Shift Keying (BPSK), Binary Frequency Shift Keying (BFSK), Differential Quadrature PSK (DQPSK), Binary Differential PSK (BDPSK), and *M*-ary PSK (MPSK) over α-μ fading channels.

An enriched $\alpha-\mu$ distribution is observed in [11] because it also can be convenient for fading model. Further, in [12], the authors analyzed the complex α-μ fading channel with an application in Orthogonal Frequency-Division Multiplexing (OFDM) systems.

The expressions for the PDF and Cumulative Distribution Function (CDF) of the square ratio of two multivariate exponentially correlated variables with α-μ distribution are determined in [13]. These formulas provide the basis for analyzing system performance in the presence of interference, based on the Signal-to-Interference Ratio (SIR), when using a Selection Combining (SC) receiver to reduce the effects of fading and interference.

The Co-Channel Interference (CCI) also occurs in wireless systems beside fading when more than one device is operating on the same frequency channel. Its influence has to be studied along with the influence of fading [14].

Our group of authors introduce CCI into analysis and made a few papers with this topic. So, in [15], an analysis of outage probability for selection combining (SC) receiver under the influence of α−μ fading and α−μ CCI is presented. The derived PDF and Pout are shown graphically. The fading and CCI parameters impact is highlighted. After, the simulation software environment for modelling and planning of wireless MIMO systems with *L*-branch SC receiver under the influence of α−μ fading and CCI is given in order to minimize the transmission costs and have the best possible Quality of Service (QoS) for defined data transfer scenario.

We performed in [16] the channel capacity of such *L*-branch SC receiver under the α−μ small-scale fading and CCI with the same distribution. The analytical results for the CC was derived in the closed form. Then, some graphs are plotted to highlight the magnitude of the disturbance. In addition, quantum computing-based machine learning approach to service consumer number prediction and Quality of Service (QoS) level estimation leveraging the previously calculated channel capacity value using Qiskit library in Python is introduced. In [17], the Level Crossing Rate (LCR) of MIMO systems with *L*-branch selection combining (SC) receiver in the presence of α-μ fading and α-μ CCI effects during transmission, is derived. After, an accelerated graphics processing unit (GPU) simulation is applied to plan a QoS-efficient 5G mobile network in a smart city. The goal is to optimize the LCR calculation speed for the observed communication system type, by providing efficient planning (reducing costs and maximizing performance) with combined approach of linear optimization and deep learning.

In this paper, we perform different performance of an α-μ fading and CCI environment when SC diversity receiver was used to mitigate the influences of these disturbances. Among them are: a MGF-based calculation of the Average Bit Error Probability (ABEP), the level crossing rate (LCR), the average fade duration (AFD), and the channel capacity. According to our knowledge, the derivation of these performance for the scenario defined here has not been reported in available literature.

Afterwards, an experimental workflow aiming to make network planning faster, relying on model-driven engineering (MDE) – for network model representation and Large Language Models (LLM) – for generating experiment code based on textual description. In this context, the expression derived in the first part of the paper is used for approach evaluation.

This work consists of six sections. After the introduction, in Section II, the SIR-based analysis of the performance of SC receiver in the presence of α-μ fading and CCI is presented, and that: PDF of the output SIR, moment generating function, ABEP for BFSK modulation and BDPSK modulation. In Section III, the second order system performance (LCR and AFD) are obtained, and channel capacity is presented in Section IV. In Section V, LLM-enabled wireless network planning workflow is done, and Section VI concludes the paper.

II. SIR- BASED PERFORMANCE ANALYSIS

We derive in the next parts of the paper the performance of a wireless system in the presence of α-μ fading and CCI. To mitigate the effects of fading and CCI, a SC diversity receiver with *L* branches is utilized. This receiver is shown in Figure 1. The SC receiver works so that transmits to the user the signal from the input with the highest value.

We have labeled the inputs with: x_i, i=1, 2, …, L; $L \geq 2$, and the output signal with x. The CCI input envelopes are y_i, i=1, 2, …, L with output value y. Considering the presence of CCI, performance will be determined on the basis of output SIR, denoted by z. Input SIRs are equal to the ratios of the useful signals and the CCIs at the input antennas ($z_i = x_i / y_i$).

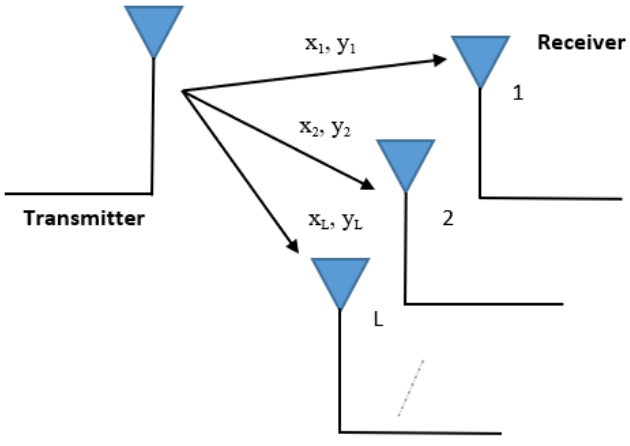


Figure 1. Model of a selection combining diversity receiver.

A. The PDF of the Output SIR

The useful signal has the α-μ distribution [3]:

$$p_{x_i}(x_i) = \frac{\alpha \mu_1^{\mu_1} x_i^{\alpha\mu_1 - 1}}{\Omega_i^{\mu_1}\Gamma(\mu_1)} \mathrm{e}^{-\mu_1 \frac{x_i^{\alpha}}{\Omega_i}} . \quad (1)$$

The parameters are: α (describes the nonlinearity of the propagation environment), μ_j (shows the number of clusters in that environment and the indices are: j=1 for the signal, and j=2 for the CCI), and Ω_i, i=1, 2, …, L, (the mean values of the input signals powers). $\Gamma(\cdot)$ denotes the Gamma function.

The CCI also follows α-μ distribution:

$$p_{y_i}(y_i) = \frac{\alpha \mu_2^{\mu_2} y_i^{\alpha\mu_2 - 1}}{s_i^{\mu_2}\Gamma(\mu_2)} \mathrm{e}^{-\mu_2 \frac{y_i^{\alpha}}{s_i}} , \quad (2)$$

where s_i marks the average powers of the CCI.

The PDFs of the SIRs z_i are given as [18]:

$$p_{z_i}(z_i) = \int_0^{\infty} y_i \, p_{x_i}(z_i y_i) \, p_{y_i}(y_i) \, dy_i . \quad (3)$$

If we substitute (1) and (2) into (3), the PDFs for SIRs are obtained as:

$$p_{z_i}(z_i) = \frac{\alpha \mu_1^{\mu_1} \mu_2^{\mu_2} z_i^{\alpha\mu_1 - 1} \Omega_i^{\mu_2} s_i^{\mu_1} \Gamma(\mu_1 + \mu_2)}{\Gamma(\mu_2)\Gamma(\mu_1)\left(\Omega_i \mu_2 + \mu_1 s_i z_i^{\alpha}\right)^{\mu_1 + \mu_2}} . \quad (4)$$

From the next formula [18], it is possible to derive the expression for CDF of z_i:

$$F_{z_i}(z_i) = \int_0^{z_i} p_{z_i}(t) \, dt . \quad (5)$$

After substitution (4) into (5), the CDF of the SIR z_i is:

$$F_{z_i}(z_i) = \frac{\alpha (\mu_1 s_i)^{\mu_1} (\mu_2 \Omega_i)^{\mu_2}}{\Gamma(\mu_2)} \cdot \frac{\Gamma(\mu_1 + \mu_2)}{\Gamma(\mu_1)} \times$$

$$\times \int_0^{z_i} \frac{z_i^{\alpha\mu_1 - 1}}{\left(\Omega_i \mu_2 + \mu_1 s_i z_i^{\alpha}\right)^{\mu_1 + \mu_2}} dt . \quad (6)$$

The integral from (6) is solved by using Beta function [19]:

$$\int_0^{\lambda} \frac{x^m}{\left(a + b x^n\right)^p} dx = \frac{a^{-p}}{n} \left(\frac{a}{b}\right)^{\frac{m+1}{n}} B_z\left(\frac{m+1}{n}, p - \frac{m+1}{n}\right)$$

$$z = \frac{b\lambda^n}{a + b\lambda^n}, a > 0, b > 0, n > 0, 0 < \frac{m+1}{n} < p . \quad (7)$$

Finally, the CDF of z_i is in the form:

$$F_{z_i}(z_i) = \frac{\Gamma(\mu_1 + \mu_2)}{\Gamma(\mu_2)\Gamma(\mu_1)} B_{\frac{\mu_1 s_i z_i^{\alpha}}{\Omega_i \mu_2 + \mu_1 s_i z_i^{\alpha}}}(\mu_1, \mu_2) . \quad (8)$$

The incomplete Beta function from (6) can be represented by [19; Eq. 8.391]:

$$B_x(p, q) = \int_0^x t^{p-1}(1-t)^{q-1} dt = \frac{x^p}{p} {}_2F_1(p, 1-q; p+1; x) =$$

$$= \frac{x^p}{p} {}_2F_1(a, b; c; z) = \frac{x^p}{p} \sum_{j=0}^{\infty} \frac{a_j b_j}{c_j} \frac{z^j}{j!} , \quad (9)$$

with ${}_2F_1$ beeing the hyper geometric function of the second order.

After a few substitutions, the CDF can be written in the form:

$$F_{z_i}(z_i) = \frac{\Gamma(\mu_1 + \mu_2)}{\mu_1 \Gamma(\mu_1)\Gamma(\mu_2)} \sum_{j=0}^{\infty} \frac{(\mu_1)_j (1-\mu_2)_j}{j!(\mu_1 + 1)_j} \left(\frac{\mu_1 s_i z_i^{\alpha}}{\Omega_i \mu_2 + \mu_1 s_i z_i^{\alpha}}\right)^{j + \mu_1} \quad (10)$$

SC receiver choses the strongest signal from L received signals and processes it (Figure 1). So, the output SIR z is the maximum SIR of all the received SIRs:

$$z = \max(z_1, z_2, ..., z_L) . \quad (11)$$

The PDF of the SIR z at the SC receiver output is [20]:

$$p_z(z) = L p_{z_i}(z_i) \left(F_{z_i}(z_i)\right)^{L-1} . \quad (12)$$

By replacing (4) and (10) into (12), the PDF of the output SIR z becomes:

$$p_z(z) = \frac{L\alpha\mu_2^{\mu_2}}{\mu_1^{L-\mu_1-1}} \cdot \frac{z_i^{\alpha\mu_1 - 1} \Omega_i^{\mu_2} s_i^{\mu_1}}{\left(\Omega_i \mu_2 + \mu_1 s_i z_i^{\alpha}\right)^{\mu_1 + \mu_2}} \left(\frac{\Gamma(\mu_1 + \mu_2)}{\Gamma(\mu_2)\Gamma(\mu_1)}\right)^L \times$$

$$\times \left(\sum_{j=0}^{+\infty} \frac{(\mu_1)_j (1-\mu_2)_j}{(\mu_1 + 1)_j \, j!} \left(\frac{\mu_1 s_i z_i^{\alpha}}{\Omega_i \mu_2 + \mu_1 s_i z_i^{\alpha}}\right)^{j + \mu_1}\right)^{L-1} . \quad (13)$$

B. Moment Generating Function

The MGF is an important statistical function for each distribution. MGF has many advantages, among which is its usefulness in analysis of sums of Random Variables (RVs). Namely, the MGF of RV gives all moments of this RV, which fact gives the name to the moment generating function. Then, if exists, the MGF determines the distribution uniquely. Therefore, if two RVs have the same MGF, they have the same distribution. Thus, if we find the MGF of a RV, its distribution is determined.

The MGF was introduced for easier determination of the system performance of fading channels in the case of complicated PDF.

In reality, the conditional BEP is a nonlinear function of the SNR or SIR. The nonlinearity is a consequence of the modulation/detection scheme. That is why we consider the MGF-based approach to determine ABEP. So, in the theory of probability and statistics, the MGF of a real RV is an alternate feature of its PDF.

The MGF is defined by formula [21; Eq. (6)]:

$$M_z(h)=\overline{e^{zh}}=\int_0^{\infty} e^{-zh}\,p_{z_i}(z)\,dz\,. \qquad (14)$$

By putting (4) into (14), the MGF for our scenario will be:

$$M_z(h)=\frac{L\alpha\mu_2{}^{\mu_2}\Omega_i{}^{\mu_2}}{\mu_1{}^{L-1}\mu_1{}^{\mu_2}s_i{}^{\mu_2}}\cdot\left(\frac{\Gamma(\mu_1+\mu_2)}{\Gamma(\mu_2)\Gamma(\mu_1)}\right)^L\times$$

$$\times\left(\sum_{j=0}^{+\infty}\frac{(\mu_1)_j(1-\mu_2)_j}{(\mu_1+1)_j\,j!}\right)^{L-1}\times$$

$$\times\int_0^{\infty}\frac{z_i{}^{\frac{2^{\frac{\alpha jL-\alpha j+\alpha\mu_1L}{2}}-1}}e^{-hz}}{\left(\left(\sqrt{\frac{\mu_2\Omega_i}{\mu_1s_i}}\right)^2+\left(z_i{}^{\frac{\alpha}{2}}\right)^2\right)^{1-(j-jL-\mu_1L-\mu_2+1)}}dz\,. \qquad (15)$$

By using the shape [19; Eq. 3.389]:

$$\int_0^{\infty}\frac{x^{2v-1}e^{-\mu x}}{\left(u^2+x^2\right)^{1-q}}dx=\frac{u^{2v+2q-2}}{2\sqrt{\pi}\,\Gamma(1-q)}G_{13}^{31}\left(\frac{\mu^2u^2}{4}\middle|\begin{matrix}1-v\\1-q-v,0,\frac{1}{2}\end{matrix}\right), \qquad (16)$$

into (15), where G[·] represents the Meijer's G-function [19; Eq. 9.301], the MGF for output SIR z becomes:

$$M_z(h)=\frac{L\alpha}{2\sqrt{\pi}\mu_1{}^{L-1}}\left(\frac{\Gamma(\mu_1+\mu_2)}{\Gamma(\mu_2)\Gamma(\mu_1)}\right)^L\left(\frac{\mu_2\Omega_i}{\mu_1s_i}\right)^{\frac{\mu_1L(\alpha-2)}{2}}\cdot$$

$$\times\left(\sum_{j=0}^{+\infty}\frac{(\mu_1)_j(1-\mu_2)_j}{(\mu_1+1)_j\,j!}\left(\frac{\mu_2\Omega_i}{\mu_1s_i}\right)^{\frac{j(\alpha-2)}{2}}\right)^{L-1}\times$$

$$\times\frac{1}{\Gamma(jL-j+\mu_1L+\mu_2)}\times$$

$$\times G_{13}^{31}\left(\frac{h^2\mu_2\Omega_i}{4\mu_1s_i}\middle|\begin{matrix}1-\left(\frac{\alpha jL-\alpha j+\alpha\mu_1L}{2}\right)\\\left(\frac{(2-\alpha)(jL-j+\mu_1L)+2\mu_2}{2}\right),0,\frac{1}{2}\end{matrix}\right). \qquad (17)$$

C. Average Bit Error Probability

The ABEP is among the system performance of the first order. It describes the system's behavior on the best way. For that reason, simply determining ABEP is of prime importance.

We determine here the MGF-based approach to determine the ABEP for two types of modulations on an efficient way, without numerical integrations.

Based on (17), the ABEP for non-coherent BFSK and BDPSK modulations are [22]:

$$P_{be}(\Omega_0)=0.5M_z(0.5)\,,\quad \text{for BFSK,} \qquad (18)$$

$$P_{be}(\Omega_0)=0.5M_z(1)\,,\quad \text{for BDPSK.} \qquad (19)$$

Follows, we illustrate the influence of fading and CCI severity on the ABEP based on numerically obtained results. For this purpose, we use the programs Origin and Mathematica to plot some figures.

D. ABEP for Binary Frequency Shift Keying Modulation

Firstly, the case of BFSK modulation is observed. The curves for ABEP versus SIR, $w=\Omega/s$, at the output of the multi-branch SC receiver, when BFSK modulation was used, are shown in Figures 2 and 3. The values for one group of parameters are changed, and the values for the other parameters are kept.

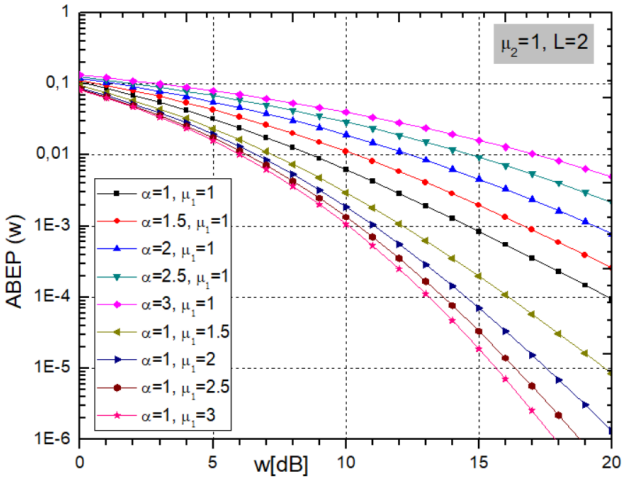


Figure 2. BEP versus SIR for BFSK modulation: parameters α and μ_1 are changing.

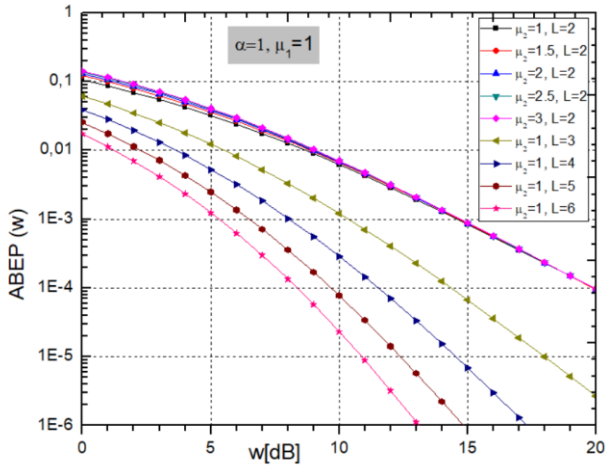


Figure 3. BEP versus SIR for BFSK modulation with variable parameters μ_2 and L.

So, in Figure 2, the ABEP is presented for BFSK modulation and dual branch SC receiver (L=2), with μ_2=1, while parameters α and μ_1 are changing. One can see from Figure 2 that the ABEP increases with an increasing in parameter α. Then, the system performance becomes worse. When the parameter μ_1 increases, the ABEP decreases and the system has better performance.

In Figure 3, the ABEP is presented versus SIR for BFSK modulation when the parameters μ_2 and L are variable. In this figure, the parameters: α=1, and μ_1=1 have maintained the same values. We can conclude that the increase in the parameter μ_2 has no effect on the ABEP. On the other side, with an increase of the number of branches L, the ABEP decreases significantly and the system performance is improved.

E. Binary Differential Phase-Shift Keying Modulation

Now, the case of BDPSK modulation is shown. In Figures 4 and 5, for ABEP for BDPSK modulation is shown versus SIR at the output of SC receiver with L branches. Figure 4 shows graphs for dual branch SC receiver (L=2) for μ_2=1, where parameters α and μ_1 took different values. We can remark that the ABEP grows with the increasing of parameter α, spoils the system performance. When the parameter μ_1 increases, the ABEP is decreasing, improving the system performance.

In Figure 5, the ABEP is presented versus SIR for the BDPSK modulation. Here, the values of parameters μ_2 and L are changeable. The parameters that keep their values all the time are: α=1 and μ_1=1. One can see that the increasing of μ_2 is without significant impact on the ABEP. When L (the number of branches) is increasing, the ABEP decreases significantly, and the system performance are improved, which is in accordance with the theory.

When we compare the last two pairs of graphs, we can conclude that the system has smaller ABEP and better performance when BDPSK modulation is used.

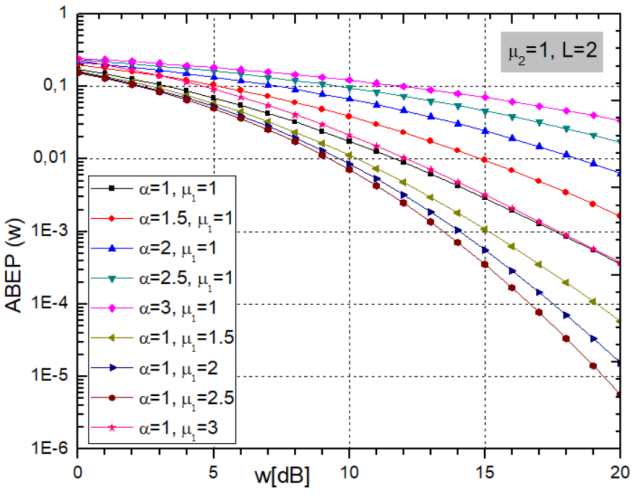


Figure 4. ABEP versus SIR for BDPSK modulation when parameters α and μ_1 are changing.

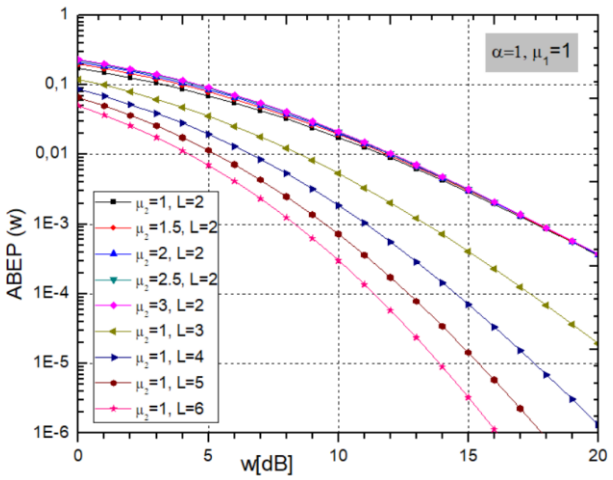


Figure 5. ABEP versus SIR for BDPSK modulation and variable parameters μ_2 and L.

III. SECOND ORDER SYSTEM PERFORMANCE

The LCR is very important the second order performance measure of wireless communication system. It presents the number of crossing the specified level in positive or negative direction. The LCR is being calculated as average value of the first derivative of random process (RP). The AFD is also second order performance measure and is defined as average time that signal envelope is below that specified threshold level. It can be evaluated as the ratio of the Pout and the LCR.

A. Level Crossing Rate

Let now calculate the first derivative of z_i, which we need to calculate the LCR:

$$\dot{z}_i = \frac{1}{y_i}\dot{x}_i - \frac{x_i}{y_i^2}\dot{y}_i \,. \tag{20}$$

The derivative of the α-μ RP is a Gaussian RP, and a linear combination of Gaussian processes is also a Gaussian RP. Then, the conditional Gaussian distributed $\dot{z}_i$, with zero mean, has the variance:

$$\sigma_{\dot{z}_i}^2 = \frac{1}{y_i^2}\sigma_{\dot{x}_i}^2 + \frac{x_i^2}{y_i^4}\sigma_{\dot{y}_i}^2 \quad . \tag{21}$$

The variances relating to the signal and interference are [17]:

$$\sigma_{\dot{x}_i}^2 = \left(\frac{2\pi f_m}{\alpha}\right)^2 \frac{\Omega_i x_i^{2-\alpha}}{\mu_1}, \sigma_{\dot{y}_i}^2 = \left(\frac{2\pi f_m}{\alpha}\right)^2 \frac{s_i y_i^{2-\alpha}}{\mu_2} \tag{22}$$

where f_m denotes the Doppler frequency. Unlike [17], we will show here the LCR in the case of different μ values for the signal and CCI: μ_1 for the signal, and μ_2 for the CCI.

After replacing expressions from (22) into (21), the variance $\dot{z}_i$ becomes:

$$\sigma_{\dot{z}_i}^2 = \frac{1}{z_i^{\alpha-2} y_i^{\alpha}} \left(\frac{2\pi f_m}{\alpha}\right)^2 \left(\frac{\Omega_i}{\mu_1} + z_i^{\alpha} \frac{s_i}{\mu_2}\right) \quad (23)$$

The conditional probability density functions (CPDF) of $\dot{z}_i$ and z_i are [18]:

$$p_{\dot{z}_i}\left(\dot{z}_i \middle| z_i y\right) = \frac{1}{\sqrt{2\pi}\sigma_{\dot{z}_i}} e^{-\frac{\dot{z}_i^2}{2\sigma_{\dot{z}_i}^2}},$$

$$p_{z_i}\left(z_i \middle| y_i\right) = \left|\frac{dx_i}{dz_i}\right| p_{x_i}\left(z_i y_i\right) = y_i p_{x_i}\left(z_i y_i\right) \quad (24)$$

The conditional joint probability density function (CJPDF) of z_i, $\dot{z}_i$ and y_i is [18]:

$$p_{\dot{z}_i z_i y_i}\left(\dot{z}_i z_i y_i\right) = p_{\dot{z}_i}\left(\dot{z}_i \middle| z_i y_i\right) p_{z_i}\left(z_i \middle| y_i\right) p_{y_i}\left(y_i\right) =$$

$$= p_{\dot{z}_i}\left(\dot{z}_i \middle| z_i y_i\right) p_{y_i}\left(y_i\right) y_i p_{x_i}\left(z_i y_i\right) \quad (25)$$

The joint PDF of z_i and $\dot{z}_i$ becomes finally [18]:

$$p_{\dot{z}_i z_i}\left(\dot{z}_i z_i\right) = \int_0^{\infty} p_{\dot{z}_i z_i y_i}\left(\dot{z}_i z_i y_i\right) dy_i \,. \quad (26)$$

The LCR of the SIR at the output of multi-branch SC receiver is actually the mean value of the first derivative of the SIR at the receiver output. So, it is necessary to average the first derivative by an integration [20]:

$$N_{z_i}\left(z_i\right) = \int_0^{\infty} \dot{z}_i p_{\dot{z}_i z_i}\left(\dot{z}_i z_i\right) d\dot{z}_i \quad (27)$$

Substituting the corresponding expressions in (27), we get:

$$N_{z_i}\left(z_i\right) = \int_0^{\infty} dy_i p_{y_i}\left(y_i\right) y_i p_{x_i}\left(z_i y_i\right) \int_0^{\infty} d\dot{z}_i \dot{z}_i \frac{1}{\sqrt{2\pi}\sigma_{\dot{z}_i}} e^{-\frac{\dot{z}_i^2}{2\sigma_{\dot{z}_i}^2}} =$$

$$= \frac{\sqrt{2\pi} f_m z_i^{\frac{2\alpha\mu_1-\alpha}{2}} \left(\mu_2\Omega_i\right)^{\mu_2-\frac{1}{2}} \left(\mu_1 s_i\right)^{\mu_1-\frac{1}{2}} \Gamma\left(\mu_1+\mu_2-1/2\right)}{\left(\mu_2\Omega_i + \mu_1 s_i z_i^{\alpha}\right)^{\mu_1+\mu_2-1} \Gamma\left(\mu_1\right)\Gamma\left(\mu_2\right)} \,. \quad (28)$$

The LCR of the SIR at the output of the multi-branch SC receiver is defined as [23; Eq. (8)]:

$$N_z\left(z\right) = L\left(F_{z_i}\left(z_i\right)\right)^{L-1} N_{z_i}\left(z_i\right) \quad (29)$$

By using equations (10) and (28) in (29), for i=1, 2, ..., L, LCR of SIR z at the SC receiver output becomes:

$$N_z\left(z\right) = L \frac{\sqrt{2\pi} f_m z_i^{\frac{2\alpha\mu_1-\alpha}{2}} \left(\mu_2\Omega_i\right)^{\mu_2-\frac{1}{2}} \left(\mu_1 s_i\right)^{\mu_1-\frac{1}{2}} \Gamma\left(\mu_1+\mu_2-1/2\right)}{\left(\mu_2\Omega_i + \mu_1 s_i z_i^{\alpha}\right)^{\mu_1+\mu_2-1} \Gamma\left(\mu_1\right)\Gamma\left(\mu_2\right)} \cdot$$

$$\cdot \left(\frac{\Gamma\left(\mu_1+\mu_2\right)}{\mu_1\Gamma\left(\mu_2\right)\Gamma\left(\mu_1\right)} \sum_{j=0}^{+\infty} \frac{\left(\mu_1\right)_j}{j!} \frac{\left(1-\mu_2\right)_j}{\left(\mu_1+1\right)_j} \left(\frac{\mu_1 s_i z_i^{\alpha}}{\Omega_i\mu_2 + \mu_1 s_i z_i^{\alpha}}\right)^{j+\mu_1}\right)^{L-1} \quad (30)$$

To explore the influence of fading and CCI severity on the concerned LCR, numerically obtained results in (30) are drawn in a few graphs. Figures 6 and 7 show normalized LCR depending on receiver output SIR z.

From Figure 6 is visible that when the parameter α increases, the curves of LRC narrow and the maximum of LCR curves increase.

Also, it is notable that LCR decreases for bigger values of parameter α. Further, an increasing of the parameter μ_1 for z<0 leads to decreasing of LCR what provides better performance for wireless system. For z>0, the parameter μ_1 slightly affects the LCR value.

The next figure, Figure 7 shows that at positive values of z [dB], the LCR increases as the number of branches L at the receiver input increases, and decreases for negative values of z, when the system has better performance.

When the parameter μ_2 increases for positive values of z, the LCR decreases and the system is improved. The influence of the parameter μ_2 is negligible for negative values of z.

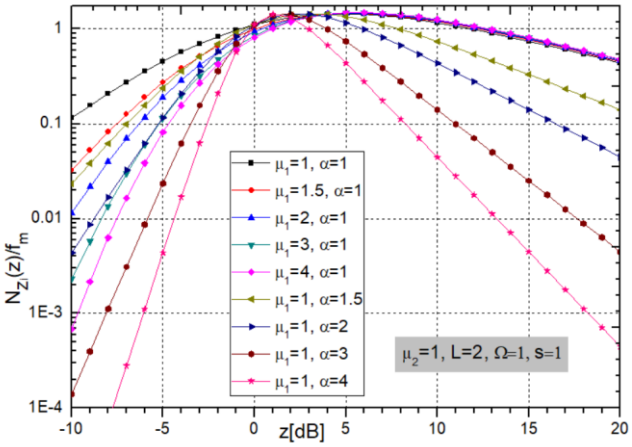


Figure 6. Normalized LCR versus SIR for variable parameters μ_1 and α.

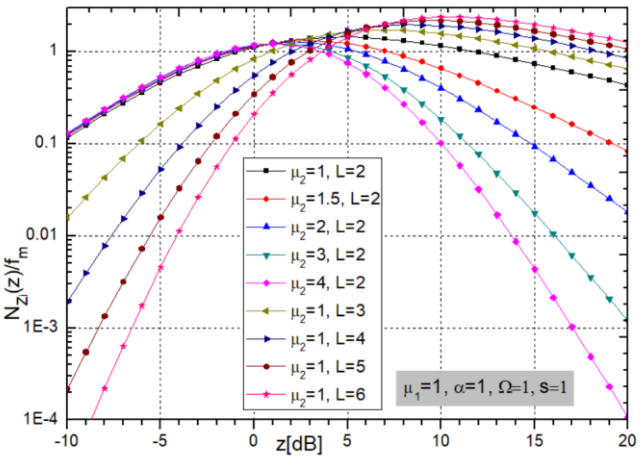


Figure 7. Normalized LCR of defined SC receiver versus SIR for variable parameter μ_2 and number of brunches L.

B. *Average Fade Duration*

For a fading signal, the AFD is defined as the average time over which the signal envelope, or SIR, remains below a certain level. As mentioned above, AFD is equal to the ratio of the Pout and the LCR [24; Eq. 2.106]:

$$AFD = \frac{F_z(z)}{N_z(z)}. \tag{31}$$

Mathematically, Pout is equal to the CDF of SIR z at the output of SC receiver [25]:

$$P_{out} = F_z(z) = \left(F_{z_i}(z_i)\right)^L \tag{32}$$

Since the CDF of z_i, i=1,2, …, L, is given by (10), Pout becomes after replacing:

$$P_{out} = \left(\frac{\Gamma(\mu_1+\mu_2)}{\mu_1\Gamma(\mu_1)\Gamma(\mu_2)} \sum_{j=0}^{\infty} \frac{(\mu_1)_j (1-\mu_2)_j}{j!(\mu_1+1)_j} \left(\frac{\mu_1 s_i z_i^{\alpha}}{\Omega_i \mu_2 + \mu_1 s_i z_i^{\alpha}} \right)^{j+\mu_1} \right)^L \tag{33}$$

The graphics for Pout for the system model described here are presented in [15; Fig. 3] for $\mu_1=\mu_2=\mu$.

Now, by replacing the corresponding expressions for $F_z(z)$ from (32) and $N_z(z)$ from (29), we get AFD:

$$AFD = \frac{\left(F_{z_i}(z_i)\right)^L}{LN_{z_i}(z_i)\left(F_{z_i}(z_i)\right)^{L-1}} =$$

$$= \frac{\Gamma(\mu_1+\mu_2)\left(\mu_2\Omega_i + \mu_1 s_i z_i^{\alpha}\right)^{\mu_1+\mu_2-1} \sum_{j=0}^{\infty} \frac{(\mu_1)_j (1-\mu_2)_j}{j!(\mu_1+1)_j} \left(\frac{\mu_1 s_i z_i^{\alpha}}{\Omega_i \mu_2 + \mu_1 s_i z_i^{\alpha}} \right)^{j+\mu_1}}{L\mu_1\sqrt{2\pi} f_m z_i^{\frac{2\alpha\mu_1-\alpha}{2}} (\mu_2\Omega_i)^{\mu_2-\frac{1}{2}} (\mu_1 s_i)^{\mu_1-\frac{1}{2}} \Gamma(\mu_1+\mu_2-1/2)} \tag{34}$$

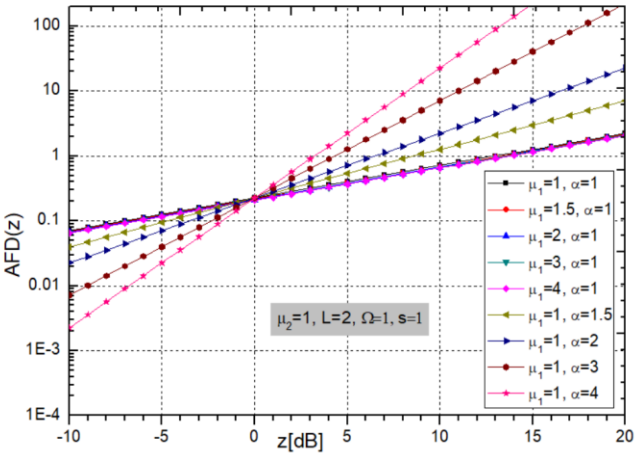


Figure 8. Normalized *AFD* of *L*-branch SC receiver depending on SIR considering different values of fading parameters μ_1 and α.

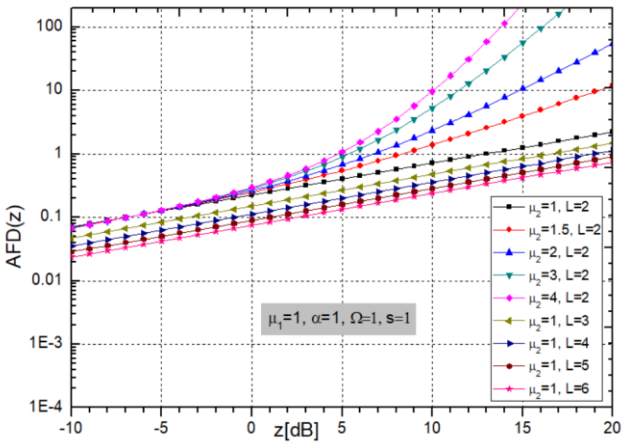


Figure 9. Normalized AFD depending on SIR for variable parameters μ_2 and L.

Because of fluctuation of wireless channel, signal amplitude, or SIR, also fluctuates, and the receiver will experience periods during which the signal cannot be reliably detected. The normalized AFDs are presented for various system parameters in Figures 8 and 9.

When the crossing threshold z is below the average signal level, the AFD is low, and this is generally the regime in which the system operates normally. It is obvious from Figure 8 that the AFD gets smaller for bigger α when z<0. The AFD increases with increasing the parameter α for z>0, and the performance is worse. This figure shows that the size of the parameter μ_1 slightly changes AFD.

From Figure 9 one can conclude that with the growth of L the AFD decreases and system performance is improved. The performance improvement in less severe environments is expected as the number of branches L increases and consequently AFD decreases. On the other side, when the parameter μ_2 increases, in the case of the CCI, the AFD also increases slightly, which is bad for system performance.

IV. CHANNEL CAPACITY

Channel capacity is of great importance between performance metrics of wireless system. CC is defined as [26]:

$$\frac{CC}{B} = \frac{1}{\ln(2)} \int_0^{\infty} \ln(1+z) p_z(z)\, dz, \tag{35}$$

where CC is Shannon capacity (in bits/s), and B is transmission bandwidth (in Hz).

Deriving an expression for CC is shown in [16]. Unlike the paper [16], here we have introduced different values of parameters μ_1 and μ_2. We give new expression obtained by the procedure in [16].

Final form of CC is:

$$CC = \frac{L}{\ln(2)\mu_1^{L-1}}\left(\frac{\Gamma(\mu_1+\mu_2)}{\Gamma(\mu_2)\Gamma(\mu_1)}\right)^L \sum_{j_1=0}^{+\infty}\frac{(-1)^{j_1}}{(j_1+1)!}\left(\frac{\mu_2}{\mu_1}\left(\frac{\Omega_i}{s_i}\right)\right)^{\frac{j_1+1}{\alpha}} \times$$

$$\times\left(\left(\sum_{j_2=0}^{+\infty}\frac{(\mu_1)_{j_2}(1-\mu_2)_{j_2}}{(\mu_1+1)_{j_2}\,j_2!}\right)^{L-1} B\left(\frac{j_1+\alpha j_2(L-1)+\alpha L\mu_1+1}{\alpha},\frac{\alpha\mu_2-j_1-1}{\alpha}\right)\right) \quad (36)$$

Then, in Figures 10 and 11, we show new graphics for the case of changing these different parameters μ_1 and μ_2, as well as parameter α and number of branches at the receiver input, *L*.

Figure 10 shows the normalized channel capacity for different values of fading parameters μ_1 and α. From this figure, it can be seen that the change of the parameter μ_1 has an insignificant effect on the CC, while when the parameter α increases, the CC decreases and the system has worse performance.

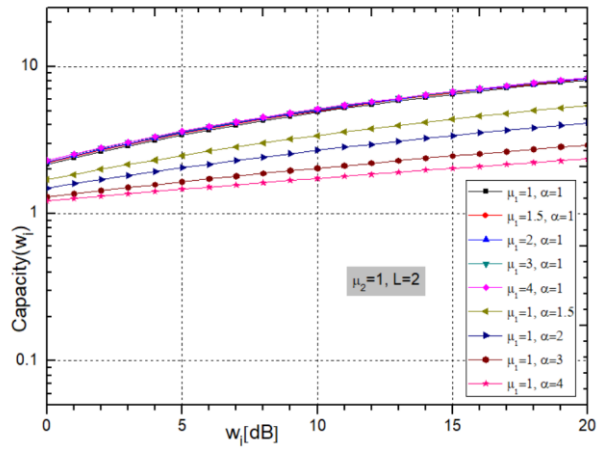


Figure 10. The normalized capacity for the use of SC receiver when the parameters μ_1 and α are changing.

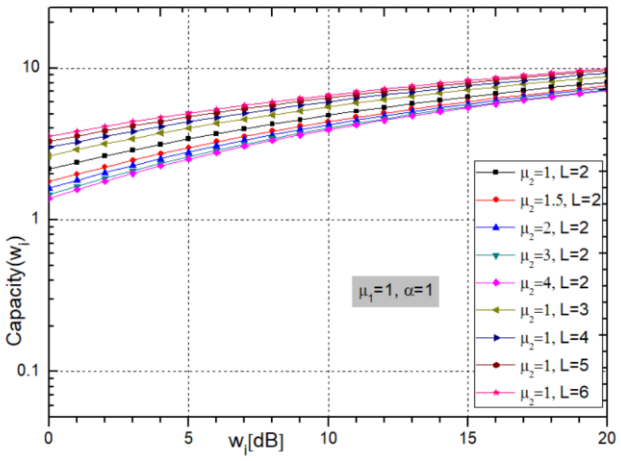


Figure 11. Normalized channel capacity for different values of CCI parameter μ_2 and number of branches *L*.

Figure 11 shows the normalized CC for some values of the CCI parameter μ_2 and variable number of input branches *L*. Other parameters have constant values $\mu_1 = \alpha = 1$. From this figure it can be seen that when μ_2 increases, the channel capacity decreases and the system performs worse. When the number *L* increases, then the channel capacity increases and the system has better performance, which is realistic to expect for a larger number of input branches of the SC receiver.

V. LLM-ENABLED WIRELESS NETWORK PLANNING WORKFLOW

Since the rise of ChatGPT [27] in late 2022, LLMs have drawn large attention, resulting in various innovative usage scenarios beside human-alike question answering – from poetry writing to playing board games. Based on experiments of many enthusiasts and researchers, it was concluded that LLMs show high potential for many use cases in area of software and computer applications [28]. One of them refers to synergy with model-driven engineering, which comes from the summarization power of LLMs [29]. This way, many novel use cases can be achieved [28]-[30]: 1) metamodel creation - domain conceptualization based on free-form text as input, resulting with metamodel as output; 2) model instance creation - using metamodel and text as input, resulting with model instance as outcome; 3) constraint rule extraction – identifying formal logic rules that should hold within the model instances, based on text and metamodel as inputs as well 4) code generation – parametrized model instances and code templates as input are leveraged to generate the target platform executable code.

Taking into account the previously mentioned scenarios, our aim in this paper is to make use of LLM and MDE synergy in order to reduce the cognitive load when it comes to experimentation in area of mobile and wireless networks. Considering the increasing infrastructure complexity, together with large number of devices involved and their heterogeneity, experimentation aiming prototyping and development of next-generation wireless and mobile networks becomes quite challenging [28], [31]. Therefore, we propose an approach based on MDE technologies for domain concept representation (Eclipse Modeling Framework's Ecore [32]) and ChatGPT, - LLM-based service that enables automated creation of model instances and code generation.

The proposed workflow is illustrated in Figure 12. In the first step, user provides free-form text describing the experiment configuration, together with experiment constraints – both related to network design and performance aspects.

After that, considering user input and meta-model on the other side, the prompt for LLM is constructed in the following form:

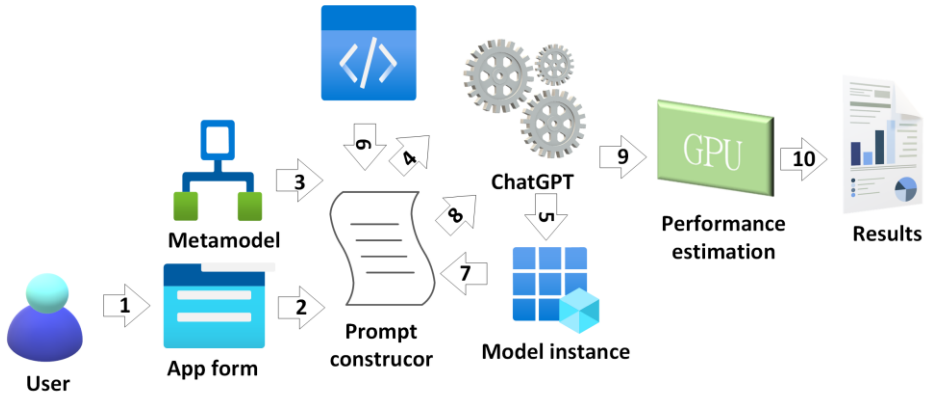


Figure 12. LLM-enabled experimental workflow for next-generation network planning: 1-Experiment definition in free-form text 2-Forwarding user-defined experiment to Prompt constructor 3-Ecore representation of metamodel 4-Prompt1 5-Generated XMI model instance 6-Experiment script Docker template 7-Taking model instance as input for code generation 8-Prompt2 9-Parametrized experiment 11-Performance estimations, such as ABEP.

Prompt1: Based on description {Experiment text} generate Ecore model instance with respect to metamodel {Ecore metamodel}

Prompting engine is written in Python programming language, making use of OpenAI Application Programming Interface (API) for ChatGPT. The outcome of this prompt is model instance representing experiment configuration with respect to the given metamodel.

Moreover, based on the provided model parameters, performance estimation is done with respect to performance calculation formulas taking into account the specified fading environment, such as the ABEP expression derived in this paper. For purpose of calculation acceleration, we make use of GPU-enabled approach which introduces high degree of loop-based calculation parallelization, built upon our works from [33]. In order to achieve this, we construct the prompt whose goal is to parametrize experiment script run inside Docker container (by assigning values to environment variables), taking into account the model instance parameters:

Prompt2: Parametrize template {experiment template} based on model instance {model instance}

Based on performance estimation results, model instance is augmented with performance-related aspects, so it can be checked from perspective of the desired goals as well. The underlying metamodel used for experiments is depicted in Figure 13.

The highest-level concept is network deployment. It consists of service provider infrastructure elements, such as base stations and service consumers, on the other side, leveraging different receiver types.

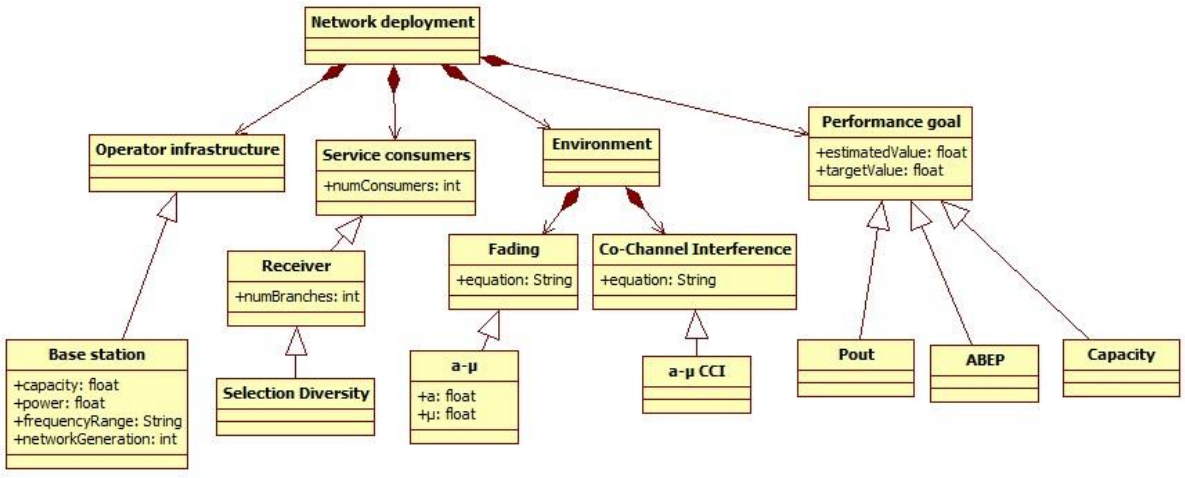


Figure 13. Network experimentation metamodel.

TABLE I. LLM-ENBLED WORKFLOW EVALUATIONAPPROACH EVALUATION

Aspect	Manual efforts	Execution time [s]	Experiment description
Text to model instance	50s – Typing the sentence	7.5	α-μ fading, α-μ CCI, ABEP 1 receiver/ 2 receivers
		12.1	
Model instance to experiment	Entirely automatic	3.3	
		8.2	
Performance estimation	Entirely automatic	1.5	
		2.4	

For telco infrastructure, we take into account their power consumption, frequency range, capacity in terms of user number and targeted network generation (2G-5G). Additionally, environmental aspects are taken into account as well in form of fading and co-channel interference type, where each specific type has distinct parameters. Finally, the model takes also into account performance-related goals which are taken into account, such as boundary values for ABEP, channel capacity or outage probability. The estimated performance value is compared to these goals in the end, so user will be notified whether the proposed deployment satisfies the requirements.

Table I gives summary of the achieved results for different experiment configurations. Execution time required for various relevant steps is given.

Taking into account that in our previous works knowledge of domain modeling tools was required, the experimental workflow required much more effort and time, up to 10 minutes for a single experiment, speed up is significant.

On the other side, LLM-aided approach significantly reduces the time required for creation of a single experiment and overall cognitive overload, as only freeform text has to be provided by end.

VI. CONCLUSION

In our work, a wireless system in fading and CCI environment was observed. The both disturbances are described by α-μ distribution. SC diversity technique was used to mitigate the effects of fading and CCI and improve the system performance. To highlight the influence of fading and CCI parameters, all derived analytical results are plotted on some figures. The MGF-based ABEP is obtained for BFSK and BDPSK modulation types. We concluded from presented graphs that, in α-μ environments, more advantageous is using of BDPSK than BFSK modulation. Then, we derived and presented LCR, AFD and CC for defined system model.

The performance derived in this paper can be used for the systems in the presence of known fading and CCI with Rayleigh, Nakagami-*m*, Weibull, and One-sided Gaussian distributions, by setting special values of parameters α and μ in α-μ general distribution.

The proposed approach leveraging LLMs and MDE significantly reduces time and effort needed for wireless network experimentation, requiring only free-from natural language text as input from the end user.

In the future we will consider correlated α-μ channels, as well as other types of fading environments. The correlation between the faded channels affects badly on the PDF of SIR at the output of the receiver. Finally, we would also take into account the adoption of LLMs for purpose of automated model constraint extraction in form of Object Constraint Rules (OCL), so automated model instance consistency checking can be performed in order to verify if it complies to some reference requirements.

ACKNOWLEDGMENT

This work is done in the frame of the bilateral project "Development of Secure and Spectral Efficient Simultaneous Wireless Information and Power Transfer Systems for Large-Scale Wireless Networks" under Science and Technological Cooperation Projects (2022-2024) between the Republic of Serbia and the Republic of India.

REFERENCES

[1] D. Krstic, S. Suljovic, D. S. Gurjar, and S. Yadav, "Moment generating function based calculation of average bit error probability in an α-μ fading environment with selection diversity receiver", IARIA Congress 2023, The 2023 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications, , Valencia, Spain, November 13, 2023 to November 17, 2023, pp. 203 – 207.

[2] M. D. Yacoub, "The α-μ distribution: a general fading distribution", The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC. 2002. Lisboa, Portugal, September 15-18, 2002. doi:10.1109/pimrc.2002.1047298

[3] M. D. Yacoub, "The *α*-μ distribution: a physical fading model for the Stacy distribution", IEEE Transactions on Vehicular Technology, vol. 56, no. 1, pp. 27-34, Jan. 2007. DOI: 10.1109/TVT.2006.883753

[4] M. D. Yacoub, "The κ-μ distribution: a general fading distribution", IEEE 54th Vehicular Technology Conference. VTC Fall 2001. doi:10.1109/vtc.2001.956432

[5] M. D. Yacoub, "The κ-μ distribution and the η-μ distribution," IEEE Antennas and Propagation Magazine, vol. 49, no. 1, pp. 68-81, Feb. 2007, doi: 10.1109/MAP.2007.370983.

[6] G. Fraidenraich and M. D. Yacoub, "The α-η-μ and α-κ-μ Fading Distributions", 2006 IEEE Ninth International Symposium on Spread Spectrum Techniques and Applications, October 2006, DOI: 10.1109/ISSSTA.2006.311725

[7] G. Fraidenraich and M. D. Yacoub, "The λ - μ general fading distribution", The 2003 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference - IMOC 2003, Foz do Iguacu, Brazil, 20-23 September, 2003, pp. 49-54. doi:10.1109/imoc.2003.1244830

[8] A. Magableh and M. Matalgah,"Moment generating function of the generalized α-μ distribution with applications", IEEE Communications Letters, vol. 13, issue 6, pp. 411–413, June 2009. DOI:10.1109/lcomm.2009.090339

[9] A. M. Magableh and M. M. Matalgah, "Channel characteristics of the generalized alpha-mu multipath fading model", The 7th International Wireless Communications and Mobile Computing Conference, IWCMC 2011, Istanbul, Turkey, 4-8 July, 2011, pp. 1535-1538. DOI: 10.1109/IWCMC.2011.5982766

[10] S. P. Singh, M. Jadon, R. Kumar, and S. Kumar "BER analysis over alpha-mu fading channel using proposed novel MGF", International Journal of Wireless and Mobile Computing, vol. 10, no. 2, pp. 174 – 182, 2016. DOI: 10.1504/IJWMC.2016.076162

[11] W. H. M. Freitas, R. C. D. V. Bomfin, R. A. A. de Souza, and M. D. Yacoub, "The complex α-μ fading channel with OFDM application",

International Journal of Antennas and Propagation, vol. 2017, 2017. https://doi.org/10.1155/2017/2143541

[12] J. T. Ferreira, A. Bekker, F. Marques, and M. Laidlaw, "An enriched α − μ model as fading candidate", Mathematical Problems in Engineering, vol. 2020, 2020. https://doi.org/10.1155/2020/5879413

[13] G. Milovanović, M. Stefanović, S. Panić, J. Anastasov, and D. Krstić, „Statistical analysis of the square ratio of two multivariate exponentially correlated α-μ distributions and its application in telecommunications", Mathematical and Computer Modelling, vol. 54, no. 1-2, pp. 152–159, July 2011. DOI:10.1016/j.mcm.2011.01.046

[14] M. K. Simon and M. S. Alouini, Digital Communication over Fading Channels, 2nd ed. Hoboken, NJ, USA: Wiley-IEEE, 2004.

[15] D. Krstic, S. Suljovic, N. Petrovic, Z. Popovic, and S. Minic, "Derivation, analysis and simulation of outage performance of MIMO multi-branch SC diversity system in α−μ fading and co-channel interference environment", 11th International Conference of Applied Internet and Information Technologies, AIIT 2021, Zrenjanin, Serbia, 15 October, 2021.

[16] D. Krstic, S. Suljovic, N. Petrovic, D. S. Gurjar, S.Yadav, and A. Rastogi, "Quantum machine learning-assisted channel capacity analysis of L−branch SC diversity receiver in α − μ fading and CCI environment", 2022 IEEE Silchar Subsection Conference, (SILCON), Silchar, India, 4-6 November 2022. DOI: 10.1109/SILCON55242.2022.10028953

[17] D. Milić, S. Suljović, D. Rančić, N. Petrović, and N. Milošević, "Performance simulation for LCR of MIMO Multi-branch SC Diversity System in α-μ fading and α-μ interference channel", IX International Conference IcETRAN, Novi Pazar, Serbia, 6 - 9. June 2022, pp. 706-710.

[18] S. Suljović, D. Krstić, D. Bandjur, S. Veljković, and M. Stefanović, "Level crossing rate of macro-diversity system in the presence of fading and co-channel interference", Revue Roumaine des Sciences Techniques, Publisher: Romanian Academy, vol. 64, pp. 63–68, 2019.

[19] I. S. Gradshteyn and I. M. Ryzhik, Tables of Integrals, Series and Products Academic. New York: 1980.

[20] S. Suljović, D. Milić, S. Panić, Č. Stefanović, and M. Stefanović, "Level crossing rate of macro diversity reception in composite Nakagami-*m* and Gamma fading environment with interference", Digital Signal Processing, Vol. 102, July 2020, 102758.

[21] N. C. Sagias and G. K. Karagiannidis, "Gaussian class multivariate Weibull distributions: Theory and applications in fading channels", IEEE Transactions on Information Theory, vol. 51, issue 10, pp. 3608–3619, 2005. DOI: 10.1109/TIT.2005.855598

[22] M. Č. Stefanović, D. M. Milović, A. M. Mitić, and M. M. Jakovljević, "Performance analysis of system with selection combining over correlated Weibull fading channels in the presence of cochannel interference", AEU - International Journal of Electronics and Communications, 62(9), pp. 695–700, 2008. 21

[23] C. Stefanovic, S. Veljković, M. Stefanović, S. Panić, and S. Jovković, "Second order statistics of SIR based macro diversity system for V2I communications over composite fading channels", First International Conference on Secure Cyber Computer and Communication (ICSCCC), Jalandhar, Punjab, India, 15-17 December, 2018, pp. 569-574.

[24] G. L. Stuber, Principles of Mobile Communication, 2nd ed., Kluwer Academic Publisher, 2000.

[25] D. Ben Cheikh Battikh, "Outage probability formulas for cellular networks: contributions for MIMO, CoMP and time reversal features", PhD Thesis, 2012, Telecom ParisTech.

[26] M. S. Alouini and A. J. Goldsmith, "Capacity of Rayleigh fading channels under different adaptive transmission and diversity-combining techniques," IEEE Transactions on Vehicular Technology, vol. 48, no. 4, pp. 1165–1181, 1999.

[27] https://chat.openai.com/, accessed on 22 March 2024.

[28] D. Krstic, N. Petrovic, S. Suljovic, and I. Al-Azzoni, "AI-enabled framework for mobile network experimentation leveraging ChatGPT: case study of channel capacity calculation for η-μ fading and co-channel interference", Electronics 2023, 12, 4088, pp. 1-19, 2023. https://doi.org/10.3390/electronics12194088

[29] N. Petrović and I. Al-Azzoni, "Model-driven smart contract generation leveraging ChatGPT", The 30th International Conference on Systems Engineering, ICSEng 2023, Las Vegas, Nevada, USA August 22-24, 2023. https://doi.org/10.1007/978-3-031-40579-2_37

[30] N. Petrović and I. Al-Azzoni, "Autoamated approach to model-driven engineering leveraging ChatGPT and Ecore", 16th International Conference on Applied Electromagnetics – IIEC 2023, Niš, Serbia, August 28 – 30, 2023, pp. 166-168.

[31] D. Krstić, N. Petrović, and I. Al-Azzoni, "Model-driven approach to fading-aware wireless network planning leveraging multiobjective optimization and deep learning", Mathematical Problems in Engineering, 4140522, 2022. https://doi.org/10.1155/2022/4140522

[32] https://eclipse.dev/modeling/emf/, accessed on 22 March 2024.

[33] N. Petrović, S. Vasić, D. Milić, S. Suljović, and S. Koničanin, "GPU-supported simulation for ABEP and QoS analysis of a combined macro diversity system in a Gamma-shadowed k-μ fading channel", Facta Universitatis, Series: Electronics and Energetics, vol. 34, no. 1, pp. 89-104, March 2021. https://doi.org/10.2298/FUEE2101089P

Identifying Semantic Similarity for UX Items from Established Questionnaires Using ChatGPT-4

Stefan Graser
Center for Advanced E-Business Studies
RheinMain University of Applied Sciences
Wiesbaden, Germany
ORCID: 0000-0002-5221-2959
stefan.graser@hs-rm.de

Martin Schrepp
SAP SE
Walldorf, Germany
ORCID: 0000-0001-7855-2524
martin.schrepp@sap.com

Stephan Böhm
Center for Advanced E-Business Studies
RheinMain University of Applied Sciences
Wiesbaden, Germany
ORCID: 0000-0003-3580-1038
stephan.boehm@hs-rm.de

***Abstract*—Questionnaires are a widely used tool for measuring the user experience (UX) of products. There exists a huge number of such questionnaires that contain different items (questions) and scales representing distinct aspects of UX, such as efficiency, learnability, fun of use, or aesthetics. These items and scales are not independent; they often have semantic overlap. However, due to the large number of available items and scales in the UX field, analyzing and understanding these semantic dependencies can be challenging. Large language models (LLM) are powerful tools to categorize texts, including UX items. We explore how ChatGPT-4 can be utilized to analyze the semantic structure of sets of UX items. This paper investigates three different use cases. In the first investigation, ChatGPT-4 is used to generate a semantic classification of UX items extracted from 40 UX questionnaires. The results demonstrate that ChatGPT-4 can effectively classify items into meaningful topics. The second investigation demonstrates ChatGPT-4's ability to filter items related to a predefined UX concept from a pool of UX items. In the third investigation, a second set of more abstract items is used to describe another classification task. The outcome of this investigation helps to determine semantic similarities between common UX concepts and enhances our understanding of the concept of UX. Overall, it is considered useful to apply GenAI in UX research.**

Keywords–User Experience (UX); Questionnaires; Semantic meaning of UX scales; Generative AI (GenAI); Large Language Model (LLM); ChatGPT; Semantic Textual Similarity (STS).

I. Introduction

Questionnaires designed to measure the user experience (UX) of products contain items that allow users to judge how effectively the product supports important aspects of user interaction and expectations. It is important to note that items in such questionnaires are semantically not independent. With the recent advances in large language models, such as ChatGPT-4, we now have the opportunity to explore the semantic similarities of UX items in a more efficient and structured manner. We enhance in this paper the first approaches [1] to use ChatGPT-4 for structuring UX items with new data and methods.

User Experience (UX) is a holistic concept in Human-Computer-Interaction (HCI) that refers to the subjective perception of users regarding the use and interaction with a product, service, or system [2]. Ensuring a good level of UX is important for the long-term success of products and services. Therefore, the perception of the users regarding UX must be investigated and measured to collect insights that can be used to enhance the UX [3]. Various methods, for example, usability tests or expert reviews, allow us to gain insights into the UX of a product. However, the most commonly used approach to measure UX is through standardized questionnaires gathering self-reported data from users [4]. These questionnaires can be applied in a cost-efficient, simple, and fast way [4][5].

UX is a complex concept that encompasses various aspects, including efficiency, learnability, enjoyment, aesthetic appeal, trust, and loyalty, among others. Since the number of questions that can be asked in a survey is limited, a single UX questionnaire can not cover all aspects comprehensively. This is why there are numerous UX questionnaires available, each optimized to address specific research questions through its items and scales. Each questionnaire measures only a subset of the potential UX aspects. Attempts to compare different UX questionnaires and to help practitioners select the most suitable one for their research questions are described in [6]–[8].

The existing UX questionnaires have been developed by various authors over a long period of time. As a result, it is not surprising that there is no consensus on the factors and items included in standardized UX questionnaires. Factors with different names may measure the same thing, while factors with the same name may measure different aspects [9]. Therefore, it is necessary to carefully examine the individual items of a scale in a questionnaire to gain a clear understanding of its meaning and potential overlap with other scales from the same or different questionnaires.

Thus, a semantic analysis of UX items from questionnaires can help to develop a deeper understanding of the meaning of UX scales. In this article, we investigate whether Generative AI, specifically ChatGPT-4, can be utilized for this purpose. We used ChatGPT-4 to analyze and compare items from existing UX questionnaires concerning their semantics. Based on this, similar items can be clustered, items representing a specific research question can be determined, and the semantic relation of commonly used UX concepts can be visualized. With this context in mind, we address the following research questions:

RQ1: *Is Generative AI able to generate a meaningful semantic classification of existing UX items?*

RQ2: *Is Generative AI able to filter items representing a predefined UX concept out of a pool of existing UX items?*

RQ3: *Can Generative AI help to uncover semantic similarities between common UX concepts and help to understand the concept of UX better?*

This article is structured as follows: Section II describes the theoretical foundation of our approach. Section III shows related work concerning the consolidation of UX factors and common ground in UX research. In addition, the research objectives are specified. Section IV illustrates the methodological approach of this study applying ChatGPT in UX research. Based on this, Sections V, VI, and VII show the three main investigations and the respective results answering the three research questions. A conclusion and outlook are given in Section VIII.

II. Theoretical Foundation

A. Concept of UX

As mentioned in the introduction, UX is a multi-faceted concept that encompasses various aspects of product quality. It is important to distinguish between the traditional concept of usability, which is defined as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" [2] and the modern concept of UX. Usability is focused on completing tasks and achieving goals with a product. UX, on the other hand, encompasses a broader spectrum of qualities that contribute to the subjective impression of a product. This includes, for example, aspects such as aesthetics or fun of use that are not directly connected to solving tasks with a system. In this sense, usability can be declared a subset of UX [8][10][11].

Hassenzahl established a distinction between pragmatic and hedonic UX qualities. Pragmatic qualities are task-related, while hedonic qualities are non-task-related [12]. However, this distinction poses some challenges. Firstly, whether some UX aspects are pragmatic or hedonic is not always clear. For example, content quality is obviously important for most web pages. If users search for specific information on a page, then high-quality content helps them find answers quickly, making it a pragmatic UX quality. On the other hand, if users stumble upon the page while browsing without a specific goal in mind, high-quality content becomes more of a hedonic quality. Secondly, pragmatic qualities adhere to a common concept as they are task-related, whereas hedonic qualities do not follow such a concept [13]; they simply encompass the remaining qualities that do not fit into the category of pragmatic qualities.

In [13], UX is conceptualized through a set of quality aspects. The basic concept is explained by defining that a "UX quality aspect describes the subjective impression of users towards a semantically clearly described aspect of product usage or product design". These UX quality aspects relate either to the external goals of the user (for example, to finish work-related tasks quickly and efficiently), to psychological needs (for example, fun of use or stimulation), sensory qualities (for example, the tactile experience when operating a device) or simply by the needs of the manufacturer to promote the product (for example, that the design looks novel and creative to attract the attention of potential customers) [13].

B. Semantic and Empirical Similarity

Our goal is to uncover the semantic similarity between items in UX questionnaires. We understand semantic similarity as the degree of likeness or resemblance between the item texts based on their meaning. In this sense, semantic similarity goes beyond surface-level syntactic or structural similarity and takes into account the context, relationships, and associations between words or phrases to determine their level of similarity [14]–[16]. Different statistics-based methods in Natural Language Processing (NLP) to measure Semantic Textual Similarity are described in the research literature [14][17]–[24]. These methods can be divided into Matrix-Based Methods, Word Distance-Based Methods, and Sentence Embedding Based Methods [25].

Large Language Models, like GPT, use word embeddings (dense vector representations of words derived with the help of deep learning mechanisms applied to vast volumes of existing texts) to calculate semantic similarity. Therefore, they are a natural choice for analyzing the semantic similarity of UX items.

However, to fully understand also the limitations of such an approach, we need to take a closer look at the relation between semantic similarity and empirical similarity of items [26][27]. In survey research, the empirical correlation of items or scales is typically used to describe how closely related they are and if they measure similar constructs. However, we may observe in studies that items with a small semantic similarity, as estimated by an LLM, show quite substantial correlations.

A well-known example is the observation that beautiful products are perceived as usable [28][29]. Thus, a substantial correlation often exists between items that measure beauty and classical usability items. Thus, visual aesthetics influence the perception of classical UX aspects like efficiency, learnability, or controllability. A similar effect exists also in the opposite direction, i.e., the perception of usability influences the perception of beauty [30][31].

Several psychological mechanisms (which, in fact, may all contribute to the effect) have been proposed to explain these unexpected empirical dependencies, for example, the general impression model [32], evaluative consistency [33], or mediator effects [34]. Another explanation is that aesthetics and usability share common aspects. It is well-known that balance, symmetry, and order [35] or alignment [36] influence the aesthetic impression. However, a user interface that looks clean, ordered, and properly aligned is also easy to scan and navigate. Users can find information faster and orient themselves more easily on such an interface. Hence, balance, symmetry, and order will also benefit efficiency or learnability [27].

Items with a high semantic similarity address similar UX aspects, and participants in a survey should give highly similar answers to such items. Therefore, items with a high semantic similarity will also show empirically high correlations. But, the converse is not always true. There may be items with low semantic similarity but substantial empirical correlations due to the aforementioned effects. Thus, we should not expect to reconstruct scales of established questionnaires by a purely semantical analysis of the items. Typically such scales are developed by an empirical process of item reduction, mostly by main component analysis, and grouping items into scales

based on their empirical correlations observed in larger studies.

C. UX Questionnaires

User experience refers to the subjective perceptions of users towards a product or system. Therefore, it is essential to gather feedback directly from users. Theoretical evaluations of UX based solely on system properties are not feasible. Since they are easy to set up and allow for the asking of many users with low effort, survey-based methods are currently the most frequently used method for quantitative UX evaluation. To ensure meaningful and comparable results, it is crucial to incorporate standardized UX questionnaires into these surveys. Additionally, collecting demographic information about participants, providing comment fields, or including specific questions can further enhance the evaluation process.

In recent decades, several standardized UX questionnaires have been developed. For instance, [9] provides a description of 40 common UX questionnaires, and an even longer list is presented in [8]. It is important to note that UX is a multifaceted concept, and no single questionnaire can cover all aspects of it. Thus, every questionnaire is based on specific UX quality aspects, which are represented as scales in the questionnaire. Each scale is represented by a number of items (questions) that correspond to the UX aspect being measured by the scale. The choice of the most suitable questionnaire for a given research question heavily depends on the specific UX quality aspects that are most relevant in that particular case. For example, when evaluating a product primarily used for professional work, classical usability aspects such as efficiency, learnability, and dependability are of high importance. Consequently, the questionnaire used for evaluation should include corresponding scales that measure these aspects. On the other hand, if the goal of the evaluation is to compare two versions of a product in terms of their visual design, a specialized questionnaire that focuses on this aspect, such as the VISAWI [37], is a better choice. Now, let's examine some examples of UX questionnaires and their item formats.

Díaz-Oreiro et al. [38] reported that the User Experience Questionnaire (UEQ) [39] is currently the most widely used questionnaire for UX evaluation. The UEQ developed by [39] is based on the distinction of UX aspects into pragmatic and hedonic scales [12][39]. The questionnaire consists of six scales:

- **Attractiveness**: Overall impression of the product. Do users like or dislike it?
- **Perspicuity**: Is it easy to get familiar with the product and to learn how to use it?
- **Efficiency**: Can users solve their tasks without unnecessary effort? Does it react fast?
- **Dependability**: Does the user feel in control of the interaction? Is it secure and predictable?
- **Stimulation**: Is it exciting and motivating to use the product? Is it fun to use?
- **Novelty**: Is the design of the product creative? Does it catch the interest of users?

The scales Perspicuity, Efficiency, and Dependability are pragmatic scales, Stimulation, and Novelty are hedonic scales, and Attractiveness is a pure valence scale (overall impression, which does not relate to concrete properties of the interaction between user and product) [39].

Each scale consists of four items. The items are semantic differentials with a 7-point Likert scale, i.e., each item consists of a pair of opposite terms that represent a UX dimension, for example, *inefficient - efficient*, *confusing - clear*, *not interesting - interesting* or *conventional - inventive*. Further details can be found online [40].

Many other questionnaires (especially the questionnaires that focus on usability, i.e., task-related UX quality) employ a different measurement concept. These questionnaires contain items that pertain to specific interface elements. For example, the Purdue Usability Testing Questionnaire [41] contains items like "Is the cursor placement consistent?" or "Does it provide visually distinctive data fields?". Other questionnaires use more abstract statements about the product to which the participants can express how much they agree or disagree on an answer scale, for example, "I found the system unnecessarily complex" (from the System Usability Scale [42]) or "The software provides me with enough information about which entries are permitted in a particular situation" or "Messages always appear in the same place" (from ISOMETRICS [43]). This type of item is more concrete but can only be applied to a certain type of product. In addition, there are several questionnaires that can be applied only for special application domains, for example, web pages, e-commerce, or games (for an overview of common questionnaires and item formulations, see [8]). The diverse formulation of items in UX questionnaires makes it challenging to categorize them based on their semantic meaning.

As previously mentioned, each UX questionnaire focuses on a specific subset of all possible UX quality aspects. Therefore, it is common practice to combine or utilize multiple questionnaires simultaneously in order to cover all relevant aspects required to answer a specific research question. However, due to the presence of different items and scales, participants may find it more challenging to complete the evaluation. Therefore, [44] developed the UEQ+, a modular framework. The framework is based on described factors with their respective items covering the construct UX as broadly as possible. Researchers can choose from a set of 27 UX quality aspects according to the respective product to evaluate and create an individualized UX questionnaire. Further information can be found online [44][45].

III. Research Objective and Related Work

Due to the various UX questionnaires, many different factors and items exist. Hence, a lack of common ground in breaking down the concept of UX can be shown in the field of quantitative UX evaluation. Against this background, only a little research was done to consolidate general UX factors and, thus, find a common understanding. This results in a respective research gap. Only three records concerning a consolidation based on empirical similarity and two records in relation to semantic similarity can be found in the literature. In the following, we present the respective approaches.

Regarding a consolidation based on empirical similarity, [46] can be first listed. [46] analyzed existing questionnaires from the literature and consolidated the collected factors based on their definition. This resulted in a consolidated list of general UX factors [46]. The second approach by [6] was based

on this. In this article, [6] also conducted the consolidation based on the definitions as well as experts. The latest approach was done by [13], resulting in a list of consolidated UX factors. In this context, the term UX quality aspect (See Section II-A) was introduced and can be considered equivalent to the term UX factor. The UX quality aspects based on [13] are shown in the following table (see Table I).

TABLE I: CONSOLIDATED UX FACTORS BASED ON [13].

(#)	Factor	Descriptive Question
(1)	Perspicuity	Is it easy to get familiar with the product and to learn how to use it?
(2)	Efficiency	Can users solve their tasks without unnecessary effort? Does the product react fast?
(3)	Dependability	Does the user feel in control of the interaction? Does the product react predictably and consistently to user commands?
(4)	Usefulness	Does using the product bring advantages to the user? Does using the product save time and effort?
(5)	Intuitive Use	Can the product be used immediately without any training or help?
(6)	Adaptability	Can the product be adapted to personal preferences or personal working styles?
(7)	Novelty	Is the design of the product creative? Does it catch the interest of users?
(8)	Stimulation	Is it exciting and motivating to use the product? Is it fun to use?
(9)	Clarity	Does the user interface of the product look ordered, tidy, and clear?
(10)	Quality of Content	Is the information provided by the product always actual and of good quality?
(11)	Immersion	Does the user forget time and sink completely into the interaction with the product?
(12)	Aesthetics	Does the product look beautiful and appealing?
(13)	Identity	Does the product help the user to socialize and to present themselves positively to other people?
(14)	Loyalty	Do people stick with the product even if there are alternative products for the same task?
(15)	Trust	Do users think that their data is in safe hands and not misused to harm them?
(16)	Value	Does the product design look professional and of high quality?

In relation to the described approaches, UX factors are typically constructed by using empirical methods of item reduction, such as principal component analysis (PCA). For this, items are grouped into factors based on their empirical correlations. As a result, scales may consist of items that represent, at least at first sight, semantically different concepts. Thus, in some cases, it is not directly clear to describe what the semantic meaning behind a scale is. This provides a completely new perspective on the concept of UX. To get a deeper understanding of the concept of UX, it makes sense to analyze the purely semantic similarities of items and to investigate a structuring based on this concept.

Up to now, only two approaches have conducted the semantic textual similarity in the field of UX research [47][48]. One of the studies is accepted for publication in 2024 [48]. Both studies applied NLP techniques at the level of the measurement items, analyzing the semantic textual similarity between the items. The main goal of both approaches was to conduct a common ground based on semantically similar measurement items. For this, a Sentence Transformer Model and a Sentence Transformer-based Topic Modeling technique were applied to analyze the semantic structure of the textual items [47][48].

The first study by [47] measured the sentence similarity by applying the Sentence Transformer Model Augmented SBERT (AugSBERT), which is a cross- and bi-encoder Transformer architecture [24]. The AugSBERT encodes the textual UX measurement items into embedding in a vector space. Based on the spatial distance, the cosine similarity between the items was calculated, and items were clustered based on a determined similarity threshold. This results in different clusters with semantically similar items [47]. The second study (which is to be published) extends the first procedure by applying the Sentence Transformer-based Topic Modeling BERTopic developed by [49]. The procedure is similar to the first approach, encoding the textual items into embeddings using the SBERT [23]. Based on this, BERTopic clustered the different embeddings [48]. The results of both studies indicate that innovative NLP techniques can be useful in determining semantic textual similarity. However, several weaknesses in both approaches can be recorded. For further insights, we refer to the respective articles [47][48].

Since the release of ChatGPT in November 2022, the development and popularity of GenAI have increased rapidly in various fields, e.g., NLP is revolutionized [50][51]. GenAI can be applied to different tasks ranging from process support to automation to enhance productivity. This article presents an extended approach based on [1] applying GenAI in UX research. For this, we aim to find out whether GenAI can be usefully applied in this field. We used ChatGPT-4 as LLM [52] to (1) (re-)construct UX factors, (2) detect and assign similar items to existing UX concepts, and (3) to analyze the semantic textual similarity of measurement items as well as assign them to the respective similar UX quality aspect. The detailed approach is explained in the following Section IV.

IV. Methodological Approach

In this study, we applied a large language model (LLM), which is becoming increasingly popular in both academia and industry. LLMs are statistical language models referring to the following characteristics [53]:

- large-scale
- pre-trained
- transformer-based neural networks

Due to their structure, LLMs indicate strong language understanding and generation abilities. Therefore, complex language tasks can be solved. Moreover, LLMs can be augmented by external knowledge and tools. Thus, LLMs are useful for a broad range of deep learning and natural language processing tasks. This also represents the largest area of application of LLMs in research, as the initial objective in the development of LLMs was to increase the performance of NLP tasks [53]–[57].

Within this domain, LLMs can effectively be used for tasks related to natural language understanding, such as text classification or semantic understanding, referring to the comprehension and interpretation of language based on the underlying semantic meaning and intent (See Section II-B). Previous research indicates the good performance of LLMs regarding these tasks [53][57]. Concerning text classification, Yang and Menczer showed that ChatGPT produced acceptable results in text classification [58]. Even though the capabilities of semantic understanding by LLMs are constrained, they also indicate reasonable results [57].

In this study, we applied an LLM for text classification and semantic understanding, with the main focus on the latter. In particular, ChatGPT-4 was used to analyze UX measurement items to determine similarity topics based on semantically similar items. ChatGPT-4 is a large multimodal model developed by OpenAI. The LLM is based on the Generative Pretrained Transformer architecture GPT-4. For detailed insights, we refer to OpenAI (https://openai.com/gpt-4) [52].

The methodological approach is a four-step procedure consisting of data collection and three investigations using ChatGPT-4. The three investigations consist of text-processing tasks referring to text classification and semantic understanding based on input data and prompting. The detailed approach is described below.

As a first step, data was collected. A set of 40 established UX questionnaires [9] was analyzed. We excluded all questionnaires with (1) a semantic differential scale and (2) a divergent measurement concept, i.e., specifically formulated items focusing on a concrete evaluation objective. This resulted in a list of 19 questionnaires with 408 measurement items. Figure 1 illustrates the data collection process.

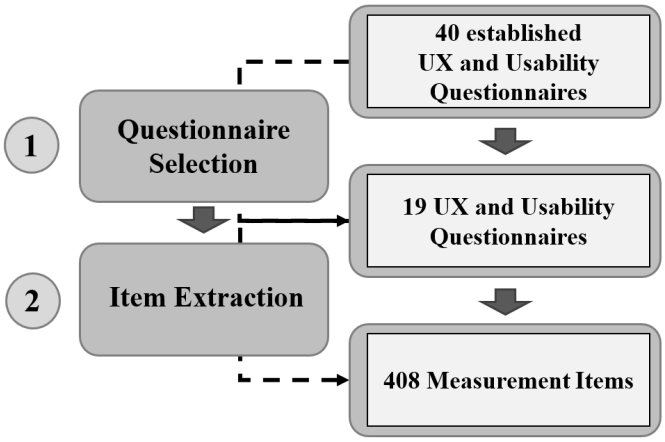


Figure 1: Data Collection.

In the second step, we aimed to gather insights into whether GenAI can perform a (Re-)Construction of UX Factors (see Section V). We introduced all items to ChatGPT-4, and six prompts were formulated. The prompts described the task for the LLM generating topics based on semantically similar items.

In the third step, we aimed to detect suitable items fitting a pre-defined UX concept very well based on the analyzed data set of items from the first step. Therefore, we formulated a generic prompt and adjusted it to each quality aspect to detect appropriate items for existing UX quality aspects (see Section VI). Such detecting and assignment is particularly useful for "ad-hoc surveys" that do not use a standardized questionnaire to measure UX, but just a bunch of self-made questions to find out something specific. This often requires spontaneous additional questions. Thus, before formulating new items, the search and detection of measurement items within an existing item pool using GenAI is quite practical.

In the fourth step, we want to go beyond such detection by analyzing the semantic textual similarity of the UX measurement items. We applied ChatGPT to standardize all items artificially. Afterward, all adjectives of positively formulated items were extracted. Based on this, we again used ChatGPT to analyze the semantic textual similarity of all adjectives. Moreover, semantically similar items were assigned to the respective semantically suitable UX quality aspect. The four-step procedure is visualized in the following Figure 2.

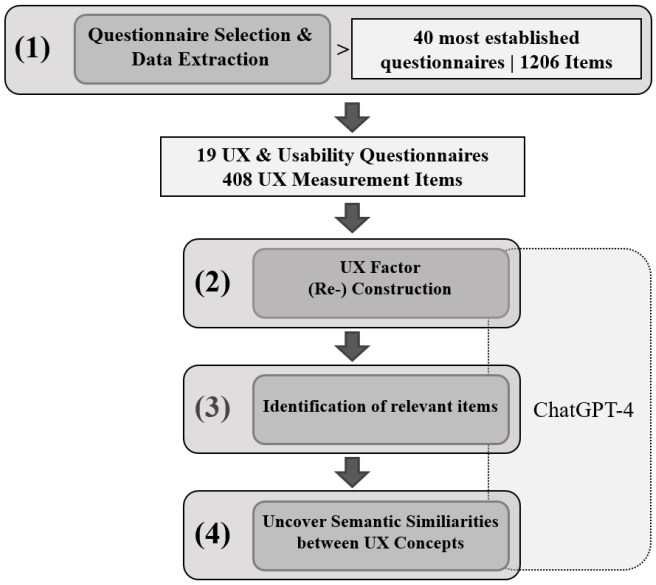


Figure 2: Methodological Approach.

The item detection for all quality aspects concerning the third step as well as step four represent the extension of this approach in relation to [1]. Further details on the procedure and the results of the respective steps are shown in the following Sections V, VI, and VII.

V. UX Factor (Re-) Construction

A. Definition of Prompts

After data collection, the second step of the procedure was performed. This first experimental part aims to answer RQ1 whether GenAI can be used to (re-)construct common UX factors. Therefore, ChatGPT was applied to (re-) construct UX factors based on the UX measurement items in relation to their semantic textual similarity. We formulated six prompts. The different tasks given to ChatGPT are described in detail below. The prompts are shown in the following:

- **prompt1**: "Can you extract the questions with a high similarity, i.e., answering about similar topics?"
- **prompt2**: "Can you break this down more detailed?"
- **prompt3**: "Can you try to break down each section into more subsections with its own category?"
- **prompt4**: "Can you improve your categorization?"
- **prompt5**: "In literature, I can find such a list with 16 UX factors.—*inserted the defined quality aspects (see Table I)*—. Can you compare this list with your categorization and contrast these lists?"
- **prompt6**: "I would like you to take your categorization you have done earlier and improve this into more generalized, holistic topics"

At first, a simple classification was performed (***prompt1***). We further tried to break this classification down to determine more specific topics (***prompt2***). In the third step, the topics were divided into subcategories by inserting ***prompt3***. ***Prompt4*** specifies the task of a topic improvement. In particular, the LLM shall optimize the respective topics and subtopics classified so far and, thus, create a further advanced classification. With ***prompt5***, we introduced existing UX quality aspects (see Section III) to ChatGPT, comparing them with the AI-generated topics in relation to their similarities and differences. By taking these into account, we lastly aimed to generate and improve the categorizations into more general topics, providing a holistic perspective with ***prompt6***. Thus, the formulated prompts mainly refer to exploratory structuring and improvement of the data.

In the following, the different prompts given to ChatGPT and the respective results are presented.

B. Results

1) Prompt1: Primary Classification: Regarding the first prompt, ChatGPT provided a first classification by themes resulting in six topics. In addition, the respective classified items were assigned to each generated topic. We have only provided the first three most representative items for each category (see Appendix A1). The classification is shown in the following:

- (1) **Usability and Ease of Use**
- (2) **Design and Aesthetics**
- (3) **User Engagement and Experience**
- (4) **Trust and Reliability**
- (5) **Information Access and Clarity**
- (6) **Issues and Errors**

Results show that common topics emerge. Topics with both functional and emotional properties were generated. In relation to the classified items, the generated topics based on the item classification are considered plausible. However, the item formulations are very specific compared to the rather broad generated categorizations. As an example, we can show Topic (1) named **Usability and Ease of Use**. The first three representative items of this topic, however, refer specifically to Ease of Use. Thus, the AI-generated topics from the first step are very broad.

As a result, we can show that ChatGPT can identify logical topics based on the semantic textual structure. However, a classification of six topics based on a total of 408 items is very superficial.

2) Prompt2: More Detailed Classification: We proceeded by asking the LLM for a more specific classification, deriving a more detailed classification. Therefore, ***prompt2*** was applied. As a result, ChatGPT classified ten topics. The respective items of the ten topics can be seen in the Appendix (see A2).

- (1) **Ease of Use**
- (2) **Complexity and Usability Issues**
- (3) **Design and Appearance**
- (4) **Engagement and Immersion**
- (5) **Performance and Responsiveness**
- (6) **Reliability and Trust**
- (7) **Information Quality and Access**
- (8) **Errors and Bugs**
- (9) **Learning and Memorability**
- (10) **Effectiveness and Efficiency**

Referring to the results of the second classification, four more topics are contained and, thus, it is more precious. In more detail, Topic (1) was further divided into two topics compared to *prompt1*. In addition, classifications of Performance and Responsiveness, Learning and Memorability, and Effectiveness and Efficiency were added. Compared to the first classification, the functional, task-related topics were further broken down. Thus, the majority of AI-generated topics relate to a rather pragmatic quality. Topic (1), (2), (5), (7), (8), (9), and (10) are of pragmatic property whereas (3) and (4) are of hedonic property addressing the emotional perception of the user. Moreover, Topic (6) – Reliability and Trust – contains both pragmatic and hedonic items. This indicates that, in general, the measurement items seem to be more pragmatic-oriented among the topics. To conclude, it was possible to distinguish the topics more precisely and categorize them in a more detailed way using the LLM.

However, the classified items within the different topics are still broad concerning the formulation. Some items can be applied to many different scenarios, e.g., *"it meets my needs"*, whereas other items show a high specification, e.g., *"I feel comfortable purchasing from the website"*. Thus, it seems logical to provide an even more detailed categorization into subcategories.

3) Prompt3: Extended Classification: By inserting ***prompt3*** we aimed to generate a more detailed classification within the different topics. We asked ChatGPT for a specific breakdown into subsections resulting in 22 further subtopics:

- **Ease of Use**
  System Usability—Website Usability—Application Usability
- **Complexity and Usability Issues**
  System Complexity—Frustration and Difficulty—System Limitations
- **Design and Appearance**
  Visual Attraction—Layout and Structure—Design Consistency
- **Engagement and Immersion**
  Time Perception and Involvement—Depth of Experience
- **Performance and Responsiveness**
  Speed of Response
- **Reliability and Trust**
  Website Trustworthiness—System Reliability
- **Information Quality and Access**
  Quality of Information—Accessibility of Information
- **Errors and Bugs**
  Technical Issues—Error Messages
- **Learning and Memorability**
  Learning Curve—Recall and Retention
- **Effectiveness and Efficiency**
  Functional Efficiency—Expected Functionality

The generated division into main- and sub-topics confirms the specification and characteristics of the measurement items on different levels. This can be traced back to the different characteristics and focus of the UX questionnaires and their items. All three prompts determined the level of categorization. A further step was taken prompting ChatGPT to make improvements.

4) Prompt4: Classification Improvement: Regarding *prompt4*, ChatGPT was given the task of improving the classification without any further specifications. This results in six main topics with 16 subtopics. The number of main topics was reduced. This returns to a rather broad generation of topics. Moreover, a broad spectrum of sub-topics was generated. Concerning the sub-topics, ChatGPT changed the categorizations and classified both pragmatic and hedonic topics together. For instance, **Aesthetics and Design** is grouped with **Navigation and Usability**.

Besides this, the LLM mainly generates suitable topics and respective sub-topics. For instance, the main topic **System Usability and Performance** contains the three sub-topics **Ease of Use, Efficiency and Speed, and Functionality and Flexibility** being purely pragmatic. By comparing this topic generation to the definition by the DIN ISO [2], it mainly captures the whole concept of usability. However, more topics are of pragmatic property than hedonic property.

- **System Usability and Performance**
  Ease of Use—Efficiency and Speed—Functionality and Flexibility
- **User Engagement and Experience**
  Engagement Level—Aesthetics and Design—Confusion and Difficulty
- **Information and Content**
  Clarity and Understandability—Relevance and Utility—Consistency and Integration
- **Website-specific Feedback**
  Navigation and Usability—Trust and Security—Aesthetics and Design
- **Learning and Adaptability**
  Learning Curve—Adaptability
- **Overall Satisfaction and Recommendation**
  Satisfaction—Recommendation

Considering the results, a two-level structure by main and sub-topics is presented. It must be mentioned that some main topics, being rather broad, contain sub-topics with pragmatic as well as hedonic properties. To sum up, ChatGPT generates a useful improvement of topics in general.

5) Prompt5: Comparison Towards Existing Consolidation: As we have already described in Section III, some approaches were conducted to consolidate UX factors and find common ground by analyzing semantic and empirical similarity. However, only the former records by [6][13][46] focusing on empirical similarity provided a systematic list of UX factors/UX quality aspects. Thus, a comparison between approaches based on empirical similarity and consolidation based on semantic similarity is useful. For this, we consulted the latest existing UX concepts (see Table I) developed by [13] and compared them to the AI-generated categories. We aimed to compare existing consolidations based on empirical similarities and the topics based on semantic similarities generated by LLM. In particular, we inserted the existing UX quality aspects and formulated the prompt as follows: *"In literature, I can find such a list with 16 UX factors.—inserted the defined quality aspects (see Table I) [13]—. Can you compare this list with your categorization and contrast these lists?"*. The comparison is illustrated in Table II.

TABLE II: COMPARISON OF EXISTING UX QUALITY ASPECTS [13] AND AI-GENERATED TOPICS.

(#)	UX Quality Aspects	AI-generated Sub-Topics
(1)	Perspicuity	Ease of Use—Learning Curve
(2)	Efficiency	Efficiency and Speed
(3)	Dependability	Consistency and Integration
(4)	Usefulness	Functionality and Flexibility—Relevance and Utility
(5)	Intuitive use	Ease of Use
(6)	Adaptability	Adaptability
(7)	Novelty	-
(8)	Stimulation	Engagement Level
(9)	Clarity	Clarity and Understandability
(10)	Quality of Content	Relevance and Utility
(11)	Immersion	Engagement Level
(12)	Aesthetics	Aesthetics and Design—Aesthetics and Design
(13)	Identity	-
(14)	Loyalty	Loyalty
(15)	Trust	Trust and Security
(16)	Value	Perceived Value

Before considering the results, it must be noted that the quality aspects by [13] do not consist of sub-topics. Results show some fundamental differences. Firstly, it must be stated that the LLM did not allocate all AI-generated topics to the existing quality aspects. In particular, the categorization does not include the UX quality aspects of *Novelty* and *Identity*. Furthermore, specific items and factors overlap as some AI-generated factors were allocated to more than one quality aspect. In general, the consolidation by [13] (see Table I) is more generalized without a focus on a specific interactive product. For instance, the LLM categorized the sub-topic *Trust and Security* in the main topic *Website-specific Feedback*. This indicates that *Trust and Security* specifically refers to the context of Websites. In contrast, *Trust* as a stand-alone quality aspect by [13] is defined more generally. To conclude, UX quality aspects based on former approaches concerning empirical similarity indicate a more holistic view covering both pragmatic and hedonic aspects of UX, whereas the AI-generated topics and sub-topics show a stronger focus on the pragmatic property as well as a deeper focus on specific products. Due to a high degree of specification, problems with general applicability may arise. Nevertheless, there are many similarities between the two consolidations, and thus, the AI-generated topics by ChatGPT can be considered logical.

6) Prompt6: Construction of Generalized Categories: Based on the former results, there is still a lack of certain generality and focus on hedonic properties within the AI-generated categories. For this, *prompt6* was formulated to create more generalized topics and, thus, to provide a more holistic view of UX. We prompted as follows: *"I would like you to take your categorization you have done earlier and improve this into more generalized, holistic topics"*. In this context, it is important to see which items represent the AI-generated topics, as the consolidation is originally based on the semantic similarity of the measurement items. We prompted

ChatGPT to issue the top five items representing the respective topic best. Concerning the results, the LLM generates a comprehensive overview with generalized UX factors as well as their definitions and items. A two-dimensional separation into the main topic and sub-topics can be shown. Additionally, both pragmatic and hedonic properties are contained. Thus, ChatGPT provides a comprehensive and generalized view of the construct of UX.

In particular, ChatGPT generated six main topics and 15 sub-topics (see Appendix A3). Concerning the results, the consolidated and AI-generated topics concerning a holistic view of UX fit well compared to previous research. Thus, the LLM is useful in deriving general UX concepts based on AI-generated topics. Pragmatic and hedonic properties are captured. The items are almost entirely coherent with each other and fit the construct. In particular, pragmatic topics show high similarities to existing literature and can be considered as well generated. However, applying ChatGPT still faces some weaknesses. For instance, different classifications of items differ quite strongly and are accordingly not representative of the respective topic. In this context, the topic ***Identity*** can be listed. In addition, items (4) and (5) (see Appendix A3) categorized in **Consistency and Integration** must be mentioned. The item's property is hedonic, whereas the topic and classified items (1)-(3) are considered pragmatic. Thus, a semantic relation between obviously pragmatic and hedonic items can be indicated. This coincides with previous research (see Section III). To illustrate the fit between item property and topic characteristics, we added a **(+)** for a suitable item fit and a **(-)** for an unsuitable item fit. It also may be that some items are contained in multiple topics due to a rather general formulation. In this case, the researchers added **(+-)** (see Appendix A3).

VI. Identification of Relevant Items

Up to this point, we have demonstrated how GenAI can be used to define a semantic structure on a large set of items from UX questionnaires. Another quite natural use case is to detect those items that best represent a clearly defined UX concept. In this section, we provide several examples to illustrate this.

A. Definition of a Generic Prompt

We use a special prompt (in the following referred to as *prompt7*) for this purpose. On top of the prompt, there was a short instruction and explanation of a typical UX concept.

For example, for Learnability (how easy or difficult it is to get familiar with a product) the corresponding instruction was:

"Below there is a list of statements and questions related to the UX of a software system. Select all statements or questions from this list that describe how easy or difficult it is to learn and understand how to use the software system. List these statements or questions. Start with those statements and questions that describe this best.

The list of 408 items from UX questionnaires was placed directly below this instructional part of the prompt.

This prompt can easily be adapted to represent other UX concepts if the part "Select all statements or questions from this list that describe how easy or difficult it is to learn and understand how to use the software system." is replaced by another formulation.

B. Results

For this example, the resulting list contained items that refer to ease of learning *(It was easy to learn to use this system)*, intuitive understanding *(The system was easy to use from the start)*, or aspects that support the user to handle the product *(Whenever I made a mistake using the system, I could recover easily and quickly)*.

The top 10 items filtered out for Learnability are:

1) It was easy to learn to use this system
2) I could effectively complete the tasks and scenarios using this system
3) I was able to complete the tasks and scenarios quickly using this system
4) I felt comfortable using this system
5) The system gave error messages that clearly told me how to fix problems
6) Whenever I made a mistake using the system, I could recover easily and quickly
7) The information provided with this system (online help, documentation) was clear
8) It was easy to find the information I needed
9) The information provided for the system was easy to understand
10) The information was effective in helping me complete the tasks and scenarios

Thus, the detected items fit well with the request in the prompt.

To assess the quality of other UX concepts, we modified the prompt by using various replacements for the variable part mentioned earlier. We explored the following additional UX concepts:

- **Efficiency:** Select all statements or questions from this list that describe how efficient or inefficient it is to work with the software system.
- **Usefulness:** Select all statements or questions from this list that describe whether the software system is useful or not.
- **Dependability:** Select all statements or questions from this list that describe if the user feels in control when he or she works with the software system or if this is not the case.
- **Stimulation:** Select all statements or questions from this list that describe how stimulating or boring it is to work with the software system.

Appendix A4 shows the top 10 items per concept. Again, the detected items fit well with the UX concepts described in the prompt.

However, there are some differences that must be highlighted. For the classical UX concepts of Efficiency, Usefulness, and Dependability, the top 10 items showed a strong alignment with these concepts. There are a few exceptions that would be classified differently by a UX expert. For example, *The processing times of the software are easy for me to estimate* was classified under Efficiency, but it is a classical item that reflects Dependability (does the user feel in control and can predict the behavior of the system). This misclassification may be due to the presence of the words *processing times*. Similarly, items 9 and 10, which were assigned to Usefulness, are more closely related to Dependability.

In terms of Stimulation, some of the items were a good fit for the concept, particularly the first four. However, the remaining items did not adequately capture the essence of Stimulation. This can be attributed to the fact that our initial item set was derived from older questionnaires that primarily focused on usability, neglecting hedonic aspects like Stimulation. Therefore, it is not surprising that the language model selected these rare examples, while the rest of the chosen items only loosely corresponded to Stimulation. This example clearly demonstrates that language models can assist UX researchers in identifying suitable items, but it is crucial to evaluate the results and make necessary corrections critically.

VII. Uncover Semantic Similarities between common UX Concepts

In our first two investigations, we utilized a collection of items derived from traditional usability questionnaires. We had to omit semantic differentials due to their distinct format compared to the statement-based items, which poses challenges for automatic analysis by a language model. For our third study, we created a new item set.

The items have been artificially created in order to achieve a highly standardized format, which would not have been possible if we had directly selected them from UX questionnaires. Each item follows the structure ”I perceive the product as <adjective>”. For example, ”I perceive the product as efficient” or ”I perceive the product as exciting”. Only positive adjectives are used. The adjectives were extracted from existing items in UX questionnaires using two methods. For semantic differentials, simply the positive term was taken (for example, from inefficient/efficient, we take the positive term efficient). For items represented as statements, we removed all other parts of the item and kept only the positive adjective. If the item has a negative formulation, i.e., there is no positive adjective, the item is ignored. For example, the item ”Is the cursor placement consistent?” is transformed into ”I perceive the product as consistent”.

In total, 135 artificial items could be constructed. See [8][59] for a similar technique to display typical UX items from standardized questionnaires as a word cloud.

A. Definition of a Generic Prompt

We use a standard prompt (referred to in the following as *prompt8*) to filter those items that correspond to a typical UX concept. On top of the prompt, there was a short instruction and explanation of a typical UX concept. For example, for Learnability the corresponding instruction was:

Below there is a list of statements related to user experience of a product. Select all statements from this list that describe that it is easy to learn and to understand how to use the product. List these statements or questions. Start with those statements and questions that describes this best.

The list of 135 artificial items was placed directly below this instructional part of the prompt.

For other UX concepts, the part *Select all statements from this list that describe that it is easy to learn and to understand how to use the product* was replaced. The rest of the prompt stays stable.

The following replacements were used:

- **Learnability:** Select all statements from this list that describe that it is easy to understand and to learn how to use the product.
- **Efficiency:** Select all statements from this list that describe that users can solve their tasks using the product efficiently without unnecessary effort and that the product reacts fast on user commands or data entries.
- **Dependability:** Select all statements from this list that describe that the user feels in control of the interaction and think it is secure and predictable.
- **Stimulation:** Select all statements from this list that describe that it is exciting, motivating and fun to use the product?
- **Novelty:** Select all statements from this list that describe that users perceive the product as original and creative.
- **Aesthetics:** Select all statements from this list that describe that the product looks beautiful, aesthetic and appealing.
- **Adaptability:** Select all statements from this list that describe that the user perceives that the product can be easily adapted to his or her personal preferences or working styles.
- **Usefulness:** Select all statements from this list that describe that users perceive the product as useful.
- **Value:** Select all statements from this list that describe that the product design looks professional and of high quality.
- **Trust:** Select all statements from this list that describe that the users think that their data are in safe hands and are not misused.
- **Clarity:** Select all statements from this list that describe that users think that the user interface of the product looks ordered, structured, and is of low visual complexity.

Each prompt was utilized in three separate runs of ChatGPT-4. For the final analysis, we only considered items that were consistently assigned to the concept in all three runs.

B. Results

The following graphic depicts the results (see Figure 3). The words in upper case font represent the UX concepts. Lowercase font the adjectives of the items (rest removed to avoid clutter). A line shows if an adjective was related to a UX concept by ChatGPT.

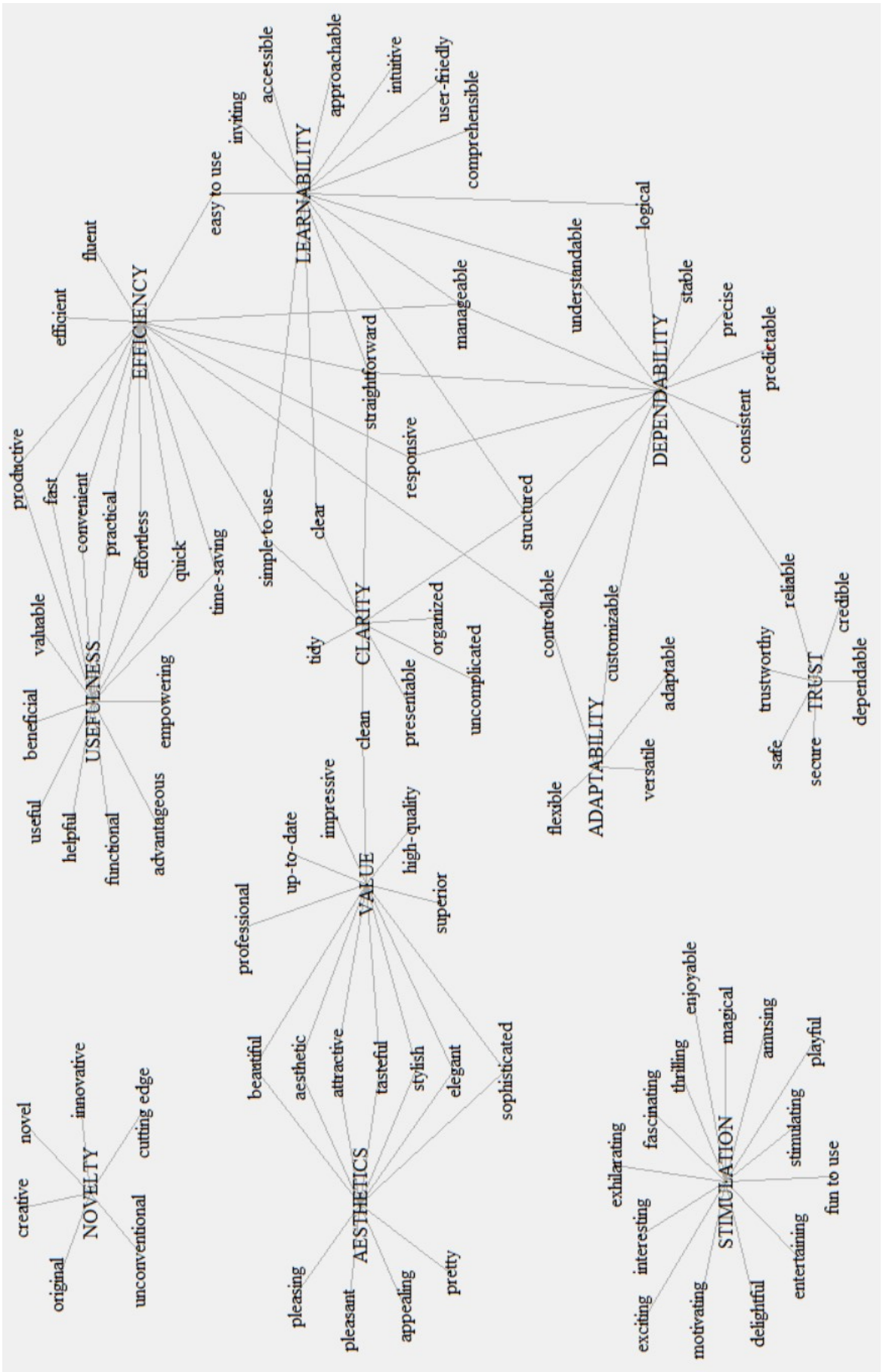


Figure 3: Uncover Semantic Similarities.

On the left side of the chart, we see the hedonic UX aspects Novelty, Stimulation, Aesthetics, and Value. Stimulation and Novelty do not share any item with other UX concepts, i.e., they represent semantically clearly distinct properties. Aesthetics and Value share a lot of items, they have a huge semantic overlap. This is a quite natural result. Value represents the feeling that a product looks professional and of high quality. But of course, a product that does look aesthetically unappealing will not be regarded as professional or of high quality. Trust is more or less isolated, but shares one item with Dependability. Adaptability is connected to Dependability and Efficiency. Usefulness is heavily connected to Efficiency. Efficiency and Learnability are connected by just one item, while both are heavily connected to Dependability. A very interesting observation is the indirect connection between Aesthetics and the classical usability criteria of Efficiency, Dependability, and Learnability. This connection is established over Value and Clarity. This fits well with empirical studies [27] that showed that Clarity is a mediator variable that explains the dependency between Aesthetics and classical usability dimensions.

Of course, we should be careful not to over-interpret these results. The outcome might be different if we modify the formulations in the prompts and of course, also depend on the version of the used LLM. However, such analyses are quite useful for understanding what typical UX concepts mean semantically and how much they overlap.

Another interesting question is how well the selected items fit empirically constructed scales. Most of the UX aspects used in this investigation correspond to scales in the UEQ or UEQ+. For the scale construction in those questionnaires, pools of items were created, data were collected from participants that evaluated different products with all items from the item pool, a principal component analysis was performed, and the four best-fitting items per component were then selected to represent the scale [39][44]. Not all adjectives used in our semantic analysis were contained in these item pools and the same is true vice versa. Thus, we can not expect a perfect match, but it is worth checking how close the empirically constructed scales are to the semantic analysis.

We list these scales and the positive term from the corresponding items (semantic differentials) in the following. The term is bold if it is also assigned to the corresponding category in our semantic analysis.

- Efficiency: **fast**, **efficient**, **practical**, organized
- Learnability (Perspicuity): **understandable**, **easy to learn**, **clear**, **easy**
- Dependability: **predictable**, supportive, secure, meets expectations
- Stimulation: valuable, **exciting**, **interesting**, **motivating**
- Novelty: **creative**, inventive, leading edge, **innovative**
- Aesthetics: **beautiful**, **stylish**, **appealing**, **pleasant**
- Adaptability: adjustable, changeable, **flexible**, extendable
- Usefulness: **useful**, **helpful**, **beneficial**, rewarding
- Value: valuable, presentable, **tasteful**, **elegant**
- Clarity: well-grouped, **structured**, ordered, **organized**
- Trust: **secure**, **trustworthy**, **reliable**, transparent

The correspondence between the semantically constructed item assignment and the empirical assignment is remarkably close for most categories. Even in cases where the items do not fully match, a comparison reveals a high degree of similarity. However, there are a few rare exceptions. For instance, the term *valuable* is represented in the UEQ+ as part of the UX scale Value (which is not surprising). In the conducted semantic analysis, it is assigned to Usefulness, which is somewhat less natural. While the overall fit to empirically constructed scales is good, there are a few exceptions that would benefit from careful review by a human expert to improve the results.

VIII. Conclusion and Future Work

In this research, we present a GenAI-based approach concerning UX research. The article aims to investigate the usefulness of GenAI in this research field. We applied the LLM ChatGPT-4 to analyze two pools of UX items from established UX questionnaires concerning three different approaches. In particular, we conducted whether GenAI can (1) (re-) construct common UX factors, (2) detect similar items, and (3) cover the semantic similarity as well as assign adjectives to semantic similar UX concepts.

A. Implications

We showed that LLMs can be usefully applied in UX research. ChatGPT was able to (1) (re-) construct and classify UX factors, (2) detect and assign similar items to the respective quality aspects, and (3) identify the semantic textual structure of the measurement items as well as assign semantic similar items to the suitable quality aspects. To conclude, applying ChatGPT was useful for conducting all three tasks. The three research questions (see Section I) can be confirmed. Thus, applying GenAI in the field of UX enhances research.

However, LLMs are inherently non-deterministic models. Hence, applying the same sequence of prompts once again, the resulting classifications will differ. Nevertheless, this is no problem as there is no objectively ”correct” classification of UX factors. Compared to the practice, conducting the same task independently by several UX experts will also result in different classifications. By applying GenAI for this task, however, the effort required for such an automatic classification is extremely low. Thus, the possibility to create such classifications quickly and efficiently allows an explorative search for semantic structures in large sets of items, uncovering interesting hidden dependencies that would be hard to detect with a manual analysis by UX experts.

Considering the results regarding the UX factor (re-) construction, ChatGPT generated a consolidated list of topics, subtopics, and items representing the concept of UX comprehensively. Within the AI-constructed topics, both pragmatic and hedonic aspects were contained. By comparing AI-generated topics with existing UX concepts, a good alignment can be illustrated. In relation to the second task, semantically similar items were detected and assigned to the existing quality aspects based on their respective definition. Regarding the third task, the LLM was useful in uncovering the semantic textual similarity of the items and assigning them to the respective UX concept.

B. Limitations and Future Research

Concerning this approach of this paper, several limitations must be drawn. Within the first step of data collection (see Section IV), semantic differentials that are a quite common item format in UX questionnaires must be excluded from the analysis to ensure at least a low level of item comparability. This mainly concerns the steps of UX factor (re-) construction and item identification. By including all formats of items, the LLM may achieve even better results.

Future research in prompt engineering shall investigate the possibility of allowing a combination of all common item formats in one analysis. Moreover, analyzing the semantic textual similarity and comparing common UX concepts (see Section VII) provides the possibility of breaking down the construct of UX in a new way.

From a practical perspective, GenAI can be usefully applied for different tasks in UX evaluation scenarios in general. More specifically, the different UX evaluation methods and their respective procedure steps must be analyzed. Based on this, the context and tasks in which GenAI is practicable and applicable must be identified. Afterward, the application within the various scenarios must be tested.

The results of this approach can be taken as a measurement framework for quantitative UX evaluation. Moreover, a UX questionnaire can be derived from the AI-generated topics and the respective items in relation to semantic textual similarity. This results in the first AI-generated UX questionnaire, which is also the first constructed UX questionnaire based on semantic similarity instead of empirical similarity. Furthermore, a comprehensive item list could be detected so that researchers do not have to develop new items but can instead use the existing pool. Thus, providing suitable measurement items quickly and easily would enhance UX evaluation and help researchers. At least, the AI-generated items could be further validated to compromise valid, reliable, and useful results.

This approach is a further step towards a common ground in UX research on the level of the measurement items. The fundamental difference between empirical and semantic similarity is to be emphasized. Moreover, this work can be seen as a first step towards a new research agenda in the field of UX.

APPENDIX

A1: Respective first three allocated items of AI-generated topics prompt1:

Usability and Ease of Use
The system is easy to use.
I found the system unnecessarily complex.
I thought the system was easy to use.

Design and Aesthetics
The design is uninteresting.
The design appears uninspired.
The color composition is attractive.

User Engagement and Experience
I felt calm using the system.
I was so involved in this experience that I lost track of time.
I lost myself in this experience.

Trust and Reliability
I feel comfortable purchasing from the website.
I feel confident conducting business on the website.
It is a site that feels secure.

Information Access and Clarity
I am able to get the information I need easily.
Provides quick and easy access to finding information.
Provides relevant information.

Issues and Errors
The system is too inflexible.
The interaction with the system is irritating.
The interaction with the system is frustrating.

A2: Respective first three allocated items of AI-generated topics of prompt2:

Ease of Use
The system is easy to use.
I thought the system was easy to use.
This system is easy to use.

Complexity and Usability Issues
I found the system unnecessarily complex.
I think that I would need the support of a technical person to be able to use this system.
I found this application confusing to use.

Design and Appearance
The design is uninteresting.
The design appears uninspired.
The color composition is attractive.

Engagement and Immersion
I felt calm using the system.
I was so involved in this experience that I lost track of time.
I lost myself in this experience.

Performance and Responsiveness
The software responds too slowly.
This software responds too slowly to inputs.
The interaction with the system is fast.

Reliability and Trust
I feel comfortable purchasing from the website.
I feel confident conducting business on the website.
It is a site that feels secure.

Information Quality and Access
I am able to get the information I need easily.
Provides quick and easy access to finding information.
Provides relevant information.

Errors and Bugs
The system is too inflexible.

The interaction with the system is irritating.
The software has at some time stopped unexpectedly.

Learning and Memorability
I think that I would need the support of a technical person to be able to use this system.
I easily remember how to use it.
It is easy to learn to use it.

Effectiveness and Efficiency
It helps me be more effective.
It meets my needs.
It does everything I would expect it to do.

A3: Generalized AI-generated UX topics with their respective top-5 items:

Usability and Intuitiveness

Ease of Use:
1) The system is easy to use. (+)
2) I think that I would need the support of a technical person to be able to use this system.(+)
3) I would imagine that most people would learn to use this system very quickly.(+)
4) I learned to use it quickly.(+)
5) I can use it without written instructions.(+)

Efficiency and Speed:
1) The interaction with the system is fast.(+)
2) The system responds too slowly.(+)
3) This software responds too slowly to inputs.(+)
4) The speed of this software is fast enough.(+)
5) Has fast navigation to pages.(+)

Adaptability:
1) The system is too inflexible.(+)
2) This software seems to disrupt the way I normally like to arrange my work.(+)
3) It is flexible.(+)
4) It requires the fewest steps possible to accomplish what I want to do with it.(+- Efficiency)
5) It is relatively easy to move from one part of a task to another.(+- Efficiency)

Content Quality and Clarity

Relevance and Utility:
1) Provides relevant information.(+)
2) It meets my needs.(+)
3) It is useful.(+)
4) Provides information content that is easy to read.(+)
5) It does everything I would expect it to do.(+)

Consistency and Integration:
1) I thought there was too much inconsistency in this system.(+)
2) I found the various functions in this system were well integrated.(+)
3) I don't notice any inconsistencies as I use it.(+)
4) Everything goes together on this site.(+-)
5) The site appears patchy.(+-)

Clarity and Understandability:
1) The way that system information is presented is clear and understandable.(+)
2) Provides information content that is easy to understand.(+)
3) I think the image is difficult to understand.(+)
4) The layout is easy to grasp.(+)
5) I do not find this image useful.(-)

Engagement and Experience

Engagement Level:
1) I was so involved in this experience that I lost track of time.(+)
2) I lost myself in this experience.(+)
3) I was really drawn into this experience.(+)
4) I felt involved in this experience.(+)
5) I was absorbed in this experience.(+)

Stimulation:
1) This experience was fun.(+)
2) I continued to use thr application out of curiosity.(+)
3) Working with this software is mentally stimulating.(+)
4) I felt involved in this experience.(+)
5) During this experience I let myself go.(+- Engagement Level)

Aesthetics and Design:
1) This application was aesthetically appealing.(+)
2) The screen layout of the application was visually pleasing.(+)
3) The design is uninteresting.(+)
4) The layout appears professionally designed.(+)
5) The design appears uninspired.(+)

Trust and Reliability

Trust and Security:
1) I feel comfortable purchasing from the website.(+)
2) I feel confident conducting business on the website.(+)
3) Is a site that feels secure.(+)
4) Makes it easy to contact the organization.(+)
5) The website is easy to use.(-)

Dependability:
1) This software hasn't always done what I was expecting.(+)
2) The software has helped me overcome any problems I have had in using it.(+)
3) I can recover from mistakes quickly and easily.(+)
4) I can use it successfully every time.(+)
5) Error messages are not adequate.(+)

Novelty and Identity

Novelty:
1) The layout is inventive.(+)

2) The layout appears dynamic.(-)
3) The layout appears too dense.(-)
4) The layout is pleasantly varied.(-)
5) The design of the site lacks a concept.(-)

Identity:
1) Conveys a sense of community.(+)
2) The offer has a clearly recognizable structure.(-)
3) Keeps the user's attention.(-)
4) The layout is not up-to-date.(-)
5) The design of the site lacks a concept.(-)

Value and Loyalty

Perceived Value:
1) I consider my experience a success.(+)
2) My experience was rewarding.(+)
3) The layout appears professionally designed.(+)
4) The color composition is attractive.(+)
5) It is wonderful.(+)

Loyalty:
1) I would recommend the application to my family and friends.(+)
2) I would recommend this software to my colleagues.(+)
3) I will likely return to the website in the future.(+)
4) I think that I would like to use this system frequently.(+)
5) I would not want to use this image.(+)

A4: Top 10 items filtered for additional UX concepts

Efficiency
1) When I work on tasks with the software, I often need more time than planned.
2) I sometimes have to search for a long time for functions that I need for my work.
3) Working with this software is sometimes cumbersome.
4) The software forces me to perform superfluous steps.
5) There are too many input steps to complete some tasks.
6) The system can only be operated in a rigidly predefined manner.
7) The processing times of the software are easy for me to estimate.
8) The software makes my task processing difficult due to inconsistent design.
9) System errors (e.g., "crash") occur during my work with the software.
10) In an error situation, the software provides concrete information on how to correct the error.

Usefulness
1) The software helps me to complete my work task better than expected without extra effort.
2) With the software, I can sometimes even exceed my desired goals without any extra effort.
3) The software allows me to increase the quality of my work without any extra work.
4) The software offers me all the possibilities I need to work on my tasks.
5) The software is tailored to the tasks I need to work on.
6) The software allows me to enter data as required by the task.
7) The software offers me a repeat function for recurring work steps.
8) Even non-routine work tasks can be easily processed with the software.
9) The software provides me with information about the current operation and usage options on request.
10) The software provides sufficient information for me about which inputs are currently permitted.

Dependability
1) I felt in control of the interaction with the system.
2) The system didn't always do what I wanted.
3) The system didn't always do what I expected.
4) The interaction with the system is unpredictable.
5) The system can only be operated in a rigidly predefined manner.
6) The software forces me to perform superfluous steps.
7) The software allows me to interrupt the editing step, although it expects an input.
8) It is possible to abort at any time when entering a command.
9) The software offers me the possibility to jump from any menu level directly back to the main menu.
10) The software offers me the possibility of customization (e.g., in menus, screen displays) to my individual needs and requirements.

Stimulation
1) I sometimes forget the time when I work with the software.
2) The software also allows me to approach my tasks creatively.
3) When I have some free time, I just play around with the software.
4) Even if my actual task is already done satisfactorily, I sometimes try to make it even better with the help of the software.
5) The software forces me to perform superfluous steps.
6) Working with this software is sometimes cumbersome.
7) The system can only be operated in a rigidly predefined manner.
8) The software makes my task processing difficult due to inconsistent design.
9) The product exhilarates me.
10) The product relaxes me.

REFERENCES

[1] S. Graser, S. Böhm, and M. Schrepp, "Using ChatGPT-4 for the identification of common ux factors within a pool of measurement items from established ux questionnaires," in *CENTRIC 2023: The Sixteenth International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services*, 2023, pp. 19–28.

[2] I. O. for Standardization 9241-210:2019, *Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems*. ISO - International Organization for Standardization, 2019.

[3] M. Rauschenberger, M. Schrepp, M. P. Cota, S. Olschner, and J. Thomaschewski, "Efficient measurement of the user experience of interactive products. how to use the user experience questionnaire (ueq).example: Spanish language version," *Int. J. Interact. Multim. Artif. Intell.*, vol. 2, pp. 39–45, 2013.

[4] W. B. Albert and T. T. Tullis, *Measuring the User Experience. Collecting, Analyzing, and Presenting UX Metrics*. Morgan Kaufmann, 2022.

[5] A. Assila, K. M. de Oliveira, and H. Ezzedine, "Standardized usability questionnaires: Features and quality focus," *Computer Science and Information Technology*, vol. 6, pp. 15–31, 2016.

[6] A. Hinderks, D. Winter, M. Schrepp, and J. Thomaschewski, "Applicability of user experience and usability questionnaires," *J. Univers. Comput. Sci.*, vol. 25, pp. 1717–1735, 2019.

[7] A. Hodrien and T. Fernando, "A review of post-study and post-task subjective questionnaires to guide assessment of system usability," *Journal of Usability Studies*, vol. 16(3), no. 3, pp. 203–232, 2021.

[8] M. Schrepp, *User Experience Questionnaires: How to use questionnaires to measure the user experience of your products?* KDP, ISBN-13: 979-8736459766, 2021.

[9] M. Schrepp, "A comparison of UX questionnaires - what is their underlying concept of user experience?" In *Mensch und Computer 2020 - Workshopband*, C. Hansen, A. Nürnberger, and B. Preim, Eds., Bonn: Gesellschaft für Informatik e.V., 2020. DOI: 10.18420/muc2020-ws105-236.

[10] H. M. Hassan and G. H. Galal-Edeen, "From usability to user experience," in *2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, 2017, pp. 216–222. DOI: 10.1109/ICIIBMS.2017.8279761.

[11] J. Preece, Y. Rogers, and H. Sharp, *Interaction Design: Beyond Human-Computer Interaction*. Wiley John + Sons, ISBN-13 978-1119020752, 2015.

[12] M. Hassenzahl, "The thing and I: Understanding the relationship between user and product," in *Funology: From Usability to Enjoyment*, M. A. Blythe, K. Overbeeke, A. F. Monk, and P. C. Wright, Eds. Dordrecht: Springer Netherlands, 2004, pp. 31–42, ISBN: 978-1-4020-2967-7. DOI: 10.1007/1-4020-2967-5_4.

[13] M. Schrepp *et al.*, "On the importance of UX quality aspects for different product categories," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. In Press, pp. 232–246, Jun. 2023. DOI: 10.9781/ijimai.2023.03.001.

[14] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality*, retrieved: 10/2023, 2013. eprint: 1310.4546. [Online]. Available: https://arxiv.org/abs/1310.4546.

[15] T. Kenter, A. Borisov, and M. de Rijke, *Siamese cbow: Optimizing word embeddings for sentence representations*, 2016. eprint: 1606.04640.

[16] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, *Supervised learning of universal sentence representations from natural language inference data*, 2018. eprint: 1705.02364.

[17] Y. Li, D. McLean, Z. A. Bandar, J. D. O'shea, and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 8, pp. 1138–1150, 2006.

[18] H. P. Luhn, "A statistical approach to mechanized encoding and searching of literary information," *IBM Journal of Research and Development*, vol. 1, no. 4, pp. 309–317, 1957.

[19] K. Spärck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 60, no. 5, pp. 493–502, 2004.

[20] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, "Okapi at trec-3," *Nist Special Publication Sp*, vol. 109, pp. 109–126, 1995.

[21] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[22] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning*, PMLR, 2014, pp. 1188–1196.

[23] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Conference on Empirical Methods in Natural Language Processing*, 2019.

[24] N. Thakur, N. Reimers, J. Daxenberger, and I. Gurevych, "Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks," *arXiv preprint arXiv:2010.08240*, Oct. 2020.

[25] X. Sun *et al.*, "Sentence similarity based on contexts," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 573–588, 2022, ISSN: 2307-387X. DOI: 10.1162/tacl_a_00477.

[26] I. Gilboa, O. Lieberman, and D. Schmeidler, "Empirical similarity," *The Review of Economics and Statistics*, vol. 88, no. 3, pp. 433–444, 2006.

[27] M. Schrepp, R. Otten, K. Blum, and J. Thomaschewski, "What causes the dependency between perceived aesthetics and perceived usability?," pp. 78–85, 2021.

[28] M. Kuroso and K. Kashimura, "Apparent usability vs. inherent usability, chi'95 conference companion," in *Conference on human factors in computing systems, Denver, Colorado*, 1995, pp. 292–293.

[29] N. Tractinsky, "Aesthetics and apparent usability: Empirically assessing cultural and methodological issues," in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, 1997, pp. 115–122.

[30] W. Ilmberger, M. Schrepp, and T. Held, "Cognitive processes causing the relationship between aesthetics and usability," in *HCI and Usability for Education and Work: 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2008, Graz, Austria, November 20-21, 2008. Proceedings 4*, Springer, 2008, pp. 43–54.

[31] A. N. Tuch, E. E. Presslaber, M. Stöcklin, K. Opwis, and J. A. Bargas-Avila, "The role of visual complexity and prototypicality regarding first impression of websites: Working towards understanding aesthetic judgments," *International Journal of Human-Computer Studies*, vol. 70, no. 11, pp. 794–811, 2012.

[32] C. E. Lance, J. A. LaPointe, and A. M. Stewart, "A test of the context dependency of three causal models of halo rater error.," *Journal of Applied Psychology*, vol. 79, no. 3, pp. 332–340, 1994.

[33] G. T. Ford and R. A. Smith, "Inferential beliefs in consumer evaluations: An assessment of alternative processing strategies," *Journal of Consumer Research*, vol. 14, no. 3, pp. 363–371, 1987.

[34] D. A. Norman, *Emotional design: Why we love (or hate) everyday things*. Civitas Books, 2004.

[35] D. C. L. Ngo, L. S. Teo, and J. G. Byrne, "Formalising guidelines for the design of screen layouts," *Displays*, vol. 21, no. 1, pp. 3–15, 2000.

[36] G. Bonsiepe, "A method of quantifying order in typographic design," *Visible Language*, vol. 2, no. 3, pp. 203–220, 1968.

[37] M. Moshagen and M. Thielsch, "Facets of visual aesthetics.," *International Journal of Human-Computer Studies, 25 (13), 1717-1735.*, no. 68(10), pp. 689–709, 2010.

[38] I. Díaz-Oreiro, G. López, L. Quesada, and Guerrero, "Standardized questionnaires for user experience evaluation: A systematic literature review," *Proceedings*, vol. 31, pp. 14–26, Nov. 2019. DOI: 10.3390/proceedings2019031014.

[39] B. Laugwitz, T. Held, and M. Schrepp, "Construction and evaluation of a user experience questionnaire," in *HCI and Usability for Education and Work*, A. Holzinger, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 63–76, ISBN: 978-3-540-89350-9.

[40] M. Schrepp, J. Thomaschewski, and A. Hinderks, "UEQ User Experience Questionnaire," 2018, retrieved: 10/2023. [Online]. Available: https://www.ueq-online.org/.

[41] H. X. Lin, Y.-Y. Choong, and G. Salvendy, "A proposed index of usability: A method for comparing the relative usability of different software systems," *Behaviour & Information Technology*, vol. 16, no. 4-5, pp. 267–277, 1997.

[42] J. Brooke, "Sus: A "quick and dirty'usability," *Usability Evaluation in Industry*, vol. 189, no. 3, pp. 189–194, 1996.

[43] G. Gediga, K.-C. Hamborg, and I. Düntsch, "The isometrics usability inventory: An operationalization of iso 9241-10 supporting summative and formative evaluation of software systems," *Behaviour & Information Technology*, vol. 18, no. 3, pp. 151–164, 1999.

[44] M. Schrepp and J. Thomaschewski, "Design and validation of a framework for the creation of user experience questionnaires," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. InPress, pp. 88–95, Dec. 2019. DOI: 10.9781/ijimai.2019.06.006.

[45] M. Schrepp and J. Thomaschewski, "UEQ+ a modular extension of the user experience questionnaire," 2019, retrieved: 10/2023. [Online]. Available: http : / / www . ueqplus . ueq - research.org/.

[46] D. Winter, M. Schrepp, and J. Thomaschewski, "Faktoren der User Experience: Systematische Übersicht über produktrelevante UX-Qualitätsaspekte," in *Workshop*, A. Endmann, H. Fischer, and M. Krökel, Eds. Berlin, München, Boston: De Gruyter, 2015, pp. 33–41, ISBN: 9783110443882. DOI: doi : 10.1515/9783110443882-005.

[47] S. Graser and S. Böhm, "Quantifying user experience through self-reporting questionnaires: A systematic analysis of the sentence similarity between the items of the measurement approaches," in *HCI International 2023 – Late Breaking Posters*, C. Stephanidis, M. Antona, S. Ntoa, and G. Salvendy, Eds., Cham: Springer Nature Switzerland, 2024, pp. 138–145, ISBN: 978-3-031-49212-9.

[48] S. Graser and S. Böhm, "Applying augmented sbert and bertopic in ux research: A sentence similarity and topic modeling approach to analyzing items from multiple questionnaires," in *Proceedings of the IWEMB 2023, Seventh International Workshop on Entrepreneurship, Electronic, and Mobile Business*, accepted for publication, to be published in 2024, 2023.

[49] M. Grootendorst, "Bertopic: Neural topic modeling with a class-based tf-idf procedure," *arXiv preprint arXiv:2203.05794*, 2022.

[50] Y. Cao *et al.*, "A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt," *arXiv:2303.04226*, pp. 1–44, 2023, retrieved: 10/2023. [Online]. Available: https://arxiv.org/abs/2303.04226.

[51] A. Bandi, P. V. S. R. Adapa, and Y. E. V. P. K. Kuchi, "The power of generative ai: A review of requirements, models, inputndash;output formats, evaluation metrics, and challenges," *Future Internet*, vol. 15, no. 8, pp. 260–320, 2023, ISSN: 1999-5903. DOI: 10.3390/fi15080260.

[52] OpenAI, "Gpt-4 technical report," *ArXiv*, vol. abs/2303.08774, 2023, retrieved: 10/2023. [Online]. Available: https://arxiv.org/abs/2303.08774.

[53] S. Minaee *et al.*, "Large language models: A survey," *arXiv preprint arXiv:2402.06196*, 2024.

[54] T. B. Brown *et al.*, "Language models are few-shot learners," *ArXiv*, vol. abs/2005.14165, 2020.

[55] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," *ArXiv*, vol. abs/2203.02155, 2022.

[56] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, "Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[57] Y. Chang *et al.*, "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 3, pp. 1–45, 2024.

[58] K.-C. Yang and F. Menczer, "Large language models can rate news outlet credibility," *ArXiv*, vol. abs/2304.00228, 2023.

[59] B. Rummel and M. S. Martin, "UX Fragebögen und Wortwolken," *Mensch und Computer 2019-Workshopband*, 2019.