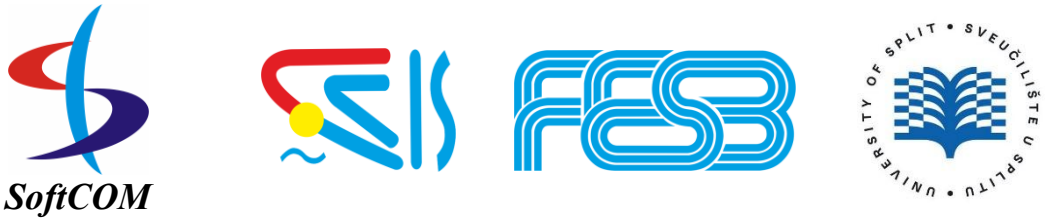
WORKSHOP ON
INFORMATION AND COMMUNICATION TECHNOLOGIES

34rd International Conference on Software,
Telecommunications and Computer Networks

SoftCOM 2026 *conjoint event*

Split, Croatia
September 17 – 19, 2026



SoftCOM 2026
conjoint event

WORKSHOP ON INFORMATION AND COMMUNICATION TECHNOLOGIES

SoftCOM 2026 *conjoint event*
Split, Croatia
September 17 – 19, 2026

Edited by:

Dinko Begušić
Joško Radić
Matko Šarić
Katarina Babić

University of Split
FESB Split

SoftCOM library

2026 34th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2026 conjoint event)
Technically cosponsored by IEEE Communications Society
Copyright © 2026 by FESB, University of Split.
All rights reserved.

ISSN 1847-3598 (Print)

Additional copies of this proceedings are available from
FESB, University of Split
R. Boskovica 32
21000 Split
Croatia
Tel. 385 21 305 795
Fax: 385 21 305 655
Email: softcom@fesb.hr
www.fesb.hr/SoftCOM

Editorial assistant: Katarina Babić
Printed by Galešnik d.o.o. Hvar

CONTENTS

WORKSHOP ON INFORMATION AND COMMUNICATION TECHNOLOGIES

WICT: WORKSHOP ON INFORMATION AND COMMUNICATIONS TECHNOLOGIES

Network-Assisted Headway Selection and Software-Induced Capacity in Homogeneous ACC Vehicle Streams **1**
Gianluca Mazzini (University of Ferrara, Italy)

From Control to Orchestration: Transformation of Data Governance in the AI Era **7**
Snjezana Rosandic and Edita Škec (Ericsson Nikola Tesla, Croatia); Katarina Babić and Dinko Begusic (University of Split, Croatia)

Approximate integration defined in Java and Python **14**
Mauro Pullin (University of Padova & Studio Ing. Mauro Pullin, Italy)

Didactic introduction to lambda-calculus **21**
Mauro Pullin (University of Padova & Studio Ing. Mauro Pullin, Italy)

High-Precision Temperature Sensor Architecture for Human-Mimicking Phantoms **25**
Stanisław Nowak, Grzegorz Żelazek, Wojciech Cichocki, Paweł Leszczyński, Krzysztof Cichoń and Cyryl Prentki (Poznan University of Technology, Poland)

Software Defined Radio for Attacks and Intrusion Detection in Wireless Networks **31**
Nikola Danilović (Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Croatia); Kresimir Grgic (Josip Juraj Strossmayer University of Osijek, Croatia)

GNSS Spoofing Detection Using Discrete Wavelet Transform and Support Vector Machines **35**
Nika Ćatipović and Katarina Babić (University of Split, Croatia)

Integrated Technologies for Wildfire Prevention and Emergency Response **40**
Slavko Tucaković (Croatian Firefighting Association Zagreb, Croatia)

REVIEWERS LIST **46**

AUTHORS INDEX **47**

Network-Assisted Headway Selection and Software-Induced Capacity in Homogeneous ACC Vehicle Streams

Gianluca Mazzini
University of Ferrara
g.mazzini@ieee.org

***Abstract*—This paper develops a minimal analytical model for a homogeneous single-lane stream of vehicles equipped with Adaptive Cruise Control (ACC) and for its network-assisted coordinated extension. A constant-time-headway equilibrium rule induces a triangular fundamental diagram with closed-form capacity, critical density, and congestion-wave speed. The original contribution is an implementation-independent cross-layer interface: safety, string-stability, latency, and comfort requirements determine the minimum admissible headway, while that value selects the complete macroscopic traffic model. Controller analysis, experiments, or communication-aware design may therefore determine the feasible headway without changing the traffic mapping. The analysis proves that capacity is strictly decreasing in headway and that, at high speed, the coordinated-to-uncoordinated capacity gain approaches the inverse headway ratio. The mapping is exact under the stated assumptions; its micro-to-macro construction is consistent with simulation-based CACC research, and its affine equilibrium premise is supported by published experiments on commercial ACC systems. Numerical results illustrate substantial but constraint-limited capacity gains and the accompanying change in congestion-wave propagation.**

***Index Terms*—Adaptive cruise control, cooperative adaptive cruise control, vehicular networks, headway control, software-induced capacity, traffic flow, fundamental diagram, kinematic waves.**

I. Introduction

Adaptive Cruise Control (ACC) regulates longitudinal motion from a desired speed and the measured distance to the preceding vehicle. Cooperative Adaptive Cruise Control (CACC) augments local sensing with communicated motion information, enabling feedforward and multi-vehicle anticipation in experimental controllers and platoons [1], [2]. Such information can extend the feasible operating region toward shorter headways, although the achievable reduction depends on vehicle dynamics, controller design, information topology, and communication delay [3]–[5].

At traffic scale, the central question is how a headway certified at vehicle level translates into aggregate road performance. Experiments show that commercial ACC systems exhibit approximately affine equilibrium spacing–speed relations and that their headway settings strongly affect capacity, storage, and congestion-wave speed [6], [7]. Conversely, string-unstable commercial implementations may reduce bottleneck throughput despite the theoretical potential of automation [8]. These findings call for a clear separation between the mechanism that establishes an admissible headway and the macroscopic traffic model induced by that headway.

This paper provides that separation for a homogeneous ACC stream governed by a constant-time-headway policy. The microscopic equilibrium rule is mapped into a triangular fundamental diagram whose capacity, critical density, and congested-wave speed are explicit functions of the headway. Network-assisted operation is represented by the smallest headway jointly compatible with safety, stability, latency, and comfort. The resulting contribution is a reusable cross-layer map: any controller, experiment, or communication architecture that certifies a feasible headway can be compared through the same traffic-level quantities. The analysis also identifies the exact coordinated capacity gain and its high-speed limit, while showing that a capacity increase is accompanied by a change in congestion-wave propagation.

The formulation is intentionally minimal and does not replace controller-level stability analysis, microsimulation, or vehicle experiments. It provides a transparent first-order interface through which their headway-related results can be compared without recalibrating a complete traffic model. The remainder of the paper positions the contribution, develops the equilibrium and conservation-law model, introduces network-assisted headway selection, derives capacity and wave effects, assesses analytical and external support, and discusses scope and implications.

II. Related Work and Positioning

Kinematic-wave traffic theory, introduced by Lighthill, Whitham, and Richards [9], [10], closes vehicle conservation with a fundamental diagram relating equilibrium flow and density. At vehicle level, constant-time-headway control yields an affine equilibrium spacing rule and has long provided a tractable basis for local and string-stability analysis of ACC streams [11], [12].

CACC research has examined how communicated predecessor or leader information can support shorter headways while retaining string stability. Ploeg et al. demonstrated a cooperative controller experimentally, whereas Milanés and Shladover identified ACC and CACC dynamic responses from field data [1], [2]. Bian et al. showed analytically that multiple-predecessor V2V information can lower the headway bound imposed by string stability [3]. Ma et al. quantified feedback

delay and parasitic lag from experiments and derived stricter stability conditions [4]; Liu et al. later adapted the communication topology to delay so as to reduce spacing while preserving stability [5]. These approaches determine feasible headways for particular dynamics, controllers, and information structures.

Empirical studies provide the complementary traffic-flow evidence. The OpenACC initiative documents heterogeneous commercial ACC behavior [6]. Shang and Stern showed through calibrated simulation that string-unstable commercial ACC may reduce bottleneck capacity [8], while Li et al. reconstructed fundamental diagrams from worldwide experiments and established the strong influence of headway on capacity, jam spacing, and equilibrium wave speed [7]. Li analytically characterized the associated trade-off among safety, mobility, and stability [13]. Donà et al. showed that radar-based multianticipation may improve stability and motorway capacity even without V2V communication [14]; consequently, reduced headway is evidence of improved sensing and control, but not by itself proof that communication is the unique enabler.

At traffic and network scale, Shi and Li constructed a triangular fundamental diagram for automated traffic from empirical and theoretical car-following descriptions and extended it to mixed traffic [15]; Zhou and Zhu modeled mixed human-driven and connected-automated streams through stochastic headways [16]. Melson et al. derived a CACC fundamental diagram from the detailed MIXIC car-following model, validated the reduction through microsimulation, and used it in dynamic traffic assignment [17]. More recently, Jiang et al. mapped analytical car-following laws to dynamic fundamental diagrams that retain hysteresis and transient control effects [18]. These approaches predict traffic effects for specified vehicle populations or car-following models. The present paper addresses the complementary question of how any externally certified headway can be translated into a complete macroscopic model. Compared with controller analysis, the proposed formulation accepts rather than derives the feasible headway; compared with empirical studies, it is not tied to one vehicle or dataset; compared with microsimulation, it omits transients and heterogeneity but gives immediate closed-form predictions; and unlike macroscopic models that prescribe a fundamental diagram, it constructs that diagram from the admissible headway. Its original contribution is this implementation-independent composition of control and communication feasibility with traffic-level performance. Table I summarizes the resulting division of roles.

Table I
RELATIONSHIP TO COMPLEMENTARY ANALYSIS APPROACHES.

Approach	Primary result	Role relative to this work
Controller and stability analysis	Feasible headway for specified dynamics and topology	Supplies safety or stability bounds entering T^*.
Vehicle experiments	Observed spacing, delay, and equilibrium behavior	Calibrates or checks assumptions and admissible headway.
Microscopic simulation	Transient and scenario-specific traffic response	Extends the equilibrium benchmark and tests implementation details.
Proposed interface	Traffic quantities from a certified headway	Compares the other approaches in common road-level units.

III. HOMOGENEOUS ACC EQUILIBRIUM MODEL

A. Scope and Assumptions

The model isolates the equilibrium traffic effect of a software-selected spacing rule. Vehicles are homogeneous, travel on one lane without overtaking, and share the same nominal speed, effective length, standstill gap, and time headway. Lane changes, cut-ins, transient actuator dynamics, and packet-level events are not represented in the equilibrium map; their role is to determine whether a candidate headway is admissible. Density and flow are expressed per lane, so the model represents a technology-saturated stream rather than mixed traffic. Within this scope it is both a benchmark and a building block: the mapping is exact for the stated assumptions, while heterogeneous, stochastic, or simulation-based models may use it as a reference case. The effective standstill space d captures speed-independent road occupancy, whereas Tv captures the part directly selected by the control policy.

B. Equilibrium Mapping

Consider a homogeneous single-lane stream with nominal speed $V > 0$, effective vehicle length $\ell > 0$, standstill gap $s_0 \geq 0$, and time headway $T > 0$. Defining $d = \ell + s_0$, the constant-time-headway equilibrium policy and the associated space occupancy are

$$\begin{aligned} s &= s_0 + Tv, \quad h = d + Tv, \\ \rho &= h^{-1}, \qquad\quad d = \ell + s_0. \end{aligned} \tag{1}$$

Solving for speed and enforcing the nominal-speed bound gives

$$v(\rho, T) = \min\left\{V, \frac{1/\rho - d}{T}\right\}, \qquad 0 < \rho \leq \rho_j = \frac{1}{d}, \tag{2}$$

with $v(0, T) = V$ by continuity. The corresponding flow is

$$q(\rho, T) = \rho v(\rho, T) = \min\left\{V\rho, \frac{1 - d\rho}{T}\right\}. \tag{3}$$

Thus, the ACC parameters induce a triangular fundamental diagram. Equating its two branches yields

$$\begin{aligned} \rho_c(T) &= \frac{1}{d + VT}, \\ Q(T) &= \frac{V}{d + VT} = \frac{1}{T + d/V}. \end{aligned} \tag{4}$$

At high nominal speed, the effective standstill-space term d/V becomes small and capacity is governed mainly by the programmed time headway.

IV. Conservation Law and Wave Speeds

The macroscopic evolution is described by

$$\rho_t + \partial_x q(\rho, T) = 0, \tag{5}$$

with q given by (3). For constant T, the characteristic speeds on the two branches are

$$c_f = V, \qquad c_j(T) = -\frac{d}{T}. \tag{6}$$

Hence the nominal speed determines downstream propagation in free flow, whereas the headway determines upstream propagation in the constrained branch.

A discontinuity connecting constant states ρ_L and ρ_R propagates at the Rankine–Hugoniot speed

$$s = \frac{q(\rho_R, T) - q(\rho_L, T)}{\rho_R - \rho_L}. \tag{7}$$

If ρ_L is constrained and ρ_R is in free flow, this becomes

$$s = \frac{V\rho_R - (1 - d\rho_L)/T}{\rho_R - \rho_L}, \tag{8}$$

with $s < 0$ corresponding to an upstream-moving queue front.

The characteristic speed c_j must not be confused with string stability. It is a macroscopic kinematic-wave speed: it describes how changes between equilibrium density states propagate under the conservation law. String stability instead concerns amplification or attenuation of vehicle-level disturbances and depends on controller dynamics, delays, and information topology. A spacing policy may therefore define the equilibrium branch in (3) even when a particular implementation cannot stably realize all points on that branch. Conversely, a large value of $|c_j|$ does not by itself indicate an unstable controller. This distinction motivates treating $T_{\text{stability}}$ as an external feasibility bound while retaining c_j as a separate traffic-level consequence of the certified headway.

V. Network-Assisted Headway Selection

In uncoordinated ACC, local sensing, safety margins, comfort, and controller design determine a feasible headway T_{ACC}. Network-assisted coordinated control can exploit communicated acceleration, speed, leader state, or multi-predecessor information and may certify a smaller value T_{CACC} [3], [5]. The two operating modes select different members of the same flux family:

$$\begin{aligned} q_{\text{ACC}}(\rho) &= q(\rho, T_{\text{ACC}}), \\ q_{\text{CACC}}(\rho) &= q(\rho, T_{\text{CACC}}). \end{aligned} \tag{9}$$

A coordinated headway cannot be reduced arbitrarily. We represent the relevant lower bounds by

$$T^* = \max\left\{T_{\text{safety}}, T_{\text{stability}}, T_{\text{latency}}, T_{\text{comfort}}\right\}. \tag{10}$$

The four terms represent safety margins, disturbance attenuation, a delay-induced headway bound associated with communication, processing, and actuation, and acceleration or jerk limits, respectively [4], [13]. The dependence of capacity on the active bound is

$$\begin{aligned} \frac{dQ}{dT} &= -\frac{V^2}{(d + VT)^2} < 0, \\ Q^* &= Q(T^*) = \frac{V}{d + VT^*}, \end{aligned} \tag{11}$$

the capacity-maximizing policy selects the smallest admissible headway, namely the largest active lower bound in (10).

The maximum operator gives this statement a precise cross-layer meaning. Let T_k denote one of the four lower bounds in (10). If T_k is the unique active bound, the local capacity sensitivity is

$$\frac{\partial Q^*}{\partial T_k} = \begin{cases} -\dfrac{V^2}{(d + VT^*)^2}, & T_k = T^*, \\ 0, & T_k < T^*. \end{cases} \tag{12}$$

At a tie between bounds the map is continuous but nonsmooth. Thus, improving a non-active constraint has no immediate capacity effect; improving the active constraint creates a gain only until another lower bound becomes dominant.

This decomposition also provides the cross-layer interpretation. A controller-specific latency margin may be written generically as

$$T_{\text{latency}} = \Phi(\tau_{\text{net}}, \tau_{\text{proc}}, \tau_{\text{act}}, \sigma_{\text{net}}, \ldots), \tag{13}$$

where the arguments denote communication, processing, and actuation delay and a measure of delay variability or reliability. Experiments and detailed models show that these quantities alter the feasible stability region [4], [5]. The present formulation does not prescribe the implementation-dependent function Φ. If a network metric z affects only this bound, its local road-level sensitivity is

$$\frac{\partial Q^*}{\partial z} = \begin{cases} -\dfrac{V^2}{(d + VT^*)^2}\dfrac{\partial \Phi}{\partial z}, & T_{\text{latency}} = T^*, \\ 0, & T_{\text{latency}} < T^*. \end{cases} \tag{14}$$

Thus, packet-level or processing improvements acquire mobility value only through their effect on the active headway bound.

The interface can be used in two directions. In forward assessment, controller analysis, experiments, or communication-aware simulation supply the admissible bounds; their maximum gives T^*, from which the complete fundamental diagram follows. In inverse design, a road-level objective first determines a required headway, and the vehicle and communication layers must demonstrate that every bound in (10) lies below it. This preserves a strict separation of responsibilities without disconnecting the layers: implementation-specific models retain their detail, while traffic analysis receives a single certified quantity with an unambiguous macroscopic meaning.

If a design effort reduces only the latency-related bound while the other bounds remain fixed, its benefit saturates at

$$\begin{aligned} T_{\text{sat}}^{(\text{lat})} &= \max\{T_{\text{safety}}, T_{\text{stability}}, T_{\text{comfort}}\}, \\ Q_{\text{sat}}^{(\text{lat})} &= \frac{V}{d + VT_{\text{sat}}^{(\text{lat})}}. \end{aligned} \tag{15}$$

This ceiling quantifies when further network improvement cannot increase capacity unless another control or safety constraint is relaxed as well.

VI. Capacity and Wave Effects of Coordination

The capacities in the two modes are

$$\begin{aligned} Q_{\mathrm{ACC}} &= \frac{V}{d + VT_{\mathrm{ACC}}}, \\ Q_{\mathrm{CACC}} &= \frac{V}{d + VT_{\mathrm{CACC}}}. \end{aligned} \tag{16}$$

Their exact ratio is

$$G_Q = \frac{Q_{\mathrm{CACC}}}{Q_{\mathrm{ACC}}} = \frac{d + VT_{\mathrm{ACC}}}{d + VT_{\mathrm{CACC}}}, \tag{17}$$

with percentage increase $100(G_Q - 1)\%$. If $T_{\mathrm{CACC}} = \alpha T_{\mathrm{ACC}}$, $0 < \alpha < 1$, then

$$G_Q(\alpha) = \frac{d + VT_{\mathrm{ACC}}}{d + \alpha VT_{\mathrm{ACC}}} \xrightarrow[VT_{\mathrm{ACC}} \gg d]{} \frac{1}{\alpha}. \tag{18}$$

The inverse-headway ratio is an upper bound at finite speed. Indeed,

$$0 < \frac{T_{\mathrm{ACC}}}{T_{\mathrm{CACC}}} - G_Q = \frac{(d/V)(T_{\mathrm{ACC}} - T_{\mathrm{CACC}})}{T_{\mathrm{CACC}}(T_{\mathrm{CACC}} + d/V)}. \tag{19}$$

The correction is controlled by the equivalent time d/V: it vanishes as speed increases, but can be non-negligible when physical occupancy is comparable with speed-dependent spacing. The exact expression should therefore be used for design, while the inverse ratio provides a transparent asymptotic interpretation. Conversely, achieving a target gain $\gamma > 1$ relative to uncoordinated ACC requires

$$T^* \le T_{\mathrm{req}}(\gamma) = \frac{d + VT_{\mathrm{ACC}}}{\gamma V} - \frac{d}{V}. \tag{20}$$

For $T_{\mathrm{req}}(\gamma) > 0$, the target is feasible if and only if every lower bound in (10) does not exceed $T_{\mathrm{req}}(\gamma)$. Equation (20) therefore converts a road-level capacity objective into the headway certificate that sensing, control, and communication must jointly provide. The normalized sensitivities make the transition explicit:

$$\begin{aligned} \mathcal{E}_T^Q &= \frac{T}{Q}\frac{\partial Q}{\partial T} = -\frac{VT}{d + VT}, \\ \mathcal{E}_V^Q &= \frac{V}{Q}\frac{\partial Q}{\partial V} = \frac{d}{d + VT}. \end{aligned} \tag{21}$$

Their absolute values sum to one. Hence capacity is speed-sensitive when standstill occupancy dominates, but becomes almost entirely headway-sensitive when $VT \gg d$. This transition is summarized by the dimensionless spacing ratio $\chi = VT/d$:

$$\frac{Qd}{V} = \frac{1}{1 + \chi}, \qquad \frac{\rho_c}{\rho_j} = \frac{1}{1 + \chi}, \qquad \frac{|c_j|}{V} = \frac{1}{\chi}. \tag{22}$$

For $\chi \ll 1$, vehicle footprint and standstill spacing dominate and changing nominal speed still affects capacity appreciably. For $\chi \gg 1$, speed-dependent spacing dominates, capacity approaches $1/T$, and the inverse-headway gain in (18) becomes accurate. The ratio therefore identifies the operating regime without introducing additional model parameters.

Headway certification may also carry uncertainty because equilibrium fits, delay measurements, and operating conditions vary. If a feasible value is known only within $T \in [T_-, T_+]$, monotonicity gives the conservative bounds

$$Q(T_+) \le Q \le Q(T_-), \qquad \rho_c(T_+) \le \rho_c \le \rho_c(T_-), \tag{23}$$

while $d/T_+ \le |c_j| \le d/T_-$. For a small perturbation ΔT, the first-order relative capacity change is

$$\frac{|\Delta Q|}{Q} \simeq \frac{VT}{d + VT}\frac{|\Delta T|}{T}. \tag{24}$$

The multiplier is the magnitude of the headway elasticity in (21). Consequently, high-χ operation, where the theoretical capacity benefit is largest, is also the regime in which uncertainty in the certified headway most directly affects capacity. Delay variability and estimation confidence should therefore enter conservative evaluations through the bound used in T^*.

The gain in critical density equals the capacity gain, whereas the change in the magnitude of the congested-wave speed is the headway ratio:

$$\begin{aligned} \frac{\rho_c(T_{\mathrm{CACC}})}{\rho_c(T_{\mathrm{ACC}})} &= G_Q, \\ \frac{|c_j(T_{\mathrm{CACC}})|}{|c_j(T_{\mathrm{ACC}})|} &= \frac{T_{\mathrm{ACC}}}{T_{\mathrm{CACC}}}. \end{aligned} \tag{25}$$

A smaller headway therefore increases both maximum sustainable flow and the density at which the nominal-speed branch ends, while also modifying upstream wave propagation. Capacity, wave behavior, safety, and string stability must consequently be evaluated together.

VII. Analytical Support and Illustrative Results

A. Correctness and External Consistency

The formulation is assessed at three complementary levels. First, the mapping from the affine equilibrium spacing policy to (3)–(4) and (6) is algebraically exact under the stated assumptions; no numerical calibration is introduced in the micro-to-macro step. Second, Melson et al. derived a fundamental diagram from the detailed MIXIC CACC model and showed that its flow predictions reproduced those obtained by implementing MIXIC in VISSIM [17]. Although their diagram is controller-specific, that agreement independently supports reduction of an equilibrium CACC rule to a macroscopic traffic representation. Third, worldwide experiments on commercial ACC systems found approximately affine equilibrium spacing–speed relations and strong headway dependence of capacity, jam spacing, and wave speed [7].

A quantitative consistency check is reported in Table II. The published minimum-headway fits for three representative systems are inserted into (4) at the reported free-flow speed of 105 km/h. The resulting capacities reproduce the published values with absolute relative errors below 0.5%. For H-min, the same parameters give $\rho_j = 84.9$ veh/km and $|c_j| = 70.7$ km/h, reproducing the published values of 85 veh/km

Table II
REPRODUCTION OF PUBLISHED MINIMUM-HEADWAY ACC CAPACITIES AT 105 KM/H [7].

Case	T [s]	d [m]	Model [veh/h/lane]	Published [veh/h/lane]	Abs. error [%]
H-min	0.60	11.78	3586	3593	0.2
W-min	0.75	9.79	3316	3322	0.2
X-min	0.54	15.00	3415	3430	0.4

and 70.7 km/h [7]. Because those published quantities are themselves obtained from equilibrium fits, this agreement is a consistency check of parameterization rather than an independent out-of-sample prediction. The independent empirical support lies in the observed affine relations and in the measured parameter ranges; the check confirms that the proposed closed-form map transports those experimental parameters into the reported traffic quantities without additional calibration.

Li et al. also report that 75% of the fitted equilibrium relations have $R^2 > 0.8$ and 54% have $R^2 > 0.9$, while minimum-headway capacities can exceed 3500 veh/h/lane and congestion-wave magnitudes can approach 100 km/h [7]. These checks do not constitute end-to-end validation of a particular network and controller. Controller-level studies additionally show that information topology and delay alter the smallest string-stable headway [3]–[5]. Taken together, the evidence supports the two links composed by the proposed interface: an implementation establishes a feasible headway, and that headway determines the corresponding equilibrium traffic quantities.

B. Constraint-Limited Gains

For illustration, let $V = 30$ m/s, $d = 7$ m, and $T_{\mathrm{ACC}} = 1.5$ s. Equation (16) gives a baseline capacity of approximately 2077 veh/h/lane. Table III reports the effects of several admissible coordinated headways.

Table III
ILLUSTRATIVE EFFECTS OF NETWORK-ASSISTED HEADWAY REDUCTION.

T_{CACC} [s]	Q_{CACC} [veh/h/lane]	Gain over ACC	$\lvert c_j \rvert$ [km/h]
1.20	2512	$+20.9\%$	21.0
1.00	2919	$+40.5\%$	25.2
0.75	3661	$+76.2\%$	33.6
0.60	4320	$+108.0\%$	42.0

The calculation is a design-sensitivity illustration rather than vehicle-specific validation. It shows that moderate headway reductions can produce substantial theoretical gains, but also that the induced wave dynamics change. The practically relevant result is therefore the gain associated with the smallest certified value T^*, not the unconstrained algebraic gain obtained by reducing T arbitrarily.

For example, let the four lower bounds in (10) be 0.8, 0.9, 1.0, and 0.7 s, with latency initially active. For the same V and d, $T^* = 1.0$ s gives 2919 veh/h/lane. Reducing the latency bound to 0.85 s shifts the active constraint to stability, so $T^* = 0.9$ s and capacity rises to approximately 3176 veh/h/lane, an 8.8% gain. Further latency reduction produces no additional capacity until the stability bound is also reduced. This example illustrates the piecewise marginal value expressed by (12).

VIII. DISCUSSION

The proposed interface is deliberately narrower than a controller model or a microsimulator. It describes homogeneous equilibrium traffic and omits mixed human–automated streams, vehicle heterogeneity, lane changes, nonlinear transients, stochastic communication failures, packet-level protocols, sensor fusion, and actuation saturation. Those effects are essential for implementation-specific assessment, but including them in the baseline would obscure the direct dependence of traffic performance on the admissible headway. The resulting quantities should therefore be interpreted as equilibrium benchmarks for a homogeneous lane, not as unconditional predictions for an arbitrary motorway scenario.

Within this scope, the model supports a practical cross-layer workflow. A control or experimental study first certifies the safety, stability, latency, and comfort bounds for the architecture under consideration. Their active maximum selects the traffic model, after which capacity, critical density, and wave speed can be compared without rebuilding a microscopic simulation for every candidate. The inverse relation in (20) also permits requirement allocation: a desired traffic gain can be translated into the maximum headway that all layers must collectively support. Finally, the active-bound sensitivity distinguishes useful technical improvements from improvements that are currently masked by another constraint.

The framework is not limited conceptually to a constant global parameter. Headway may depend on location, time, density, weather, or measured network quality, and different vehicle classes may be assigned different certified values. Those extensions require variable-flux or multiclass conservation laws and, for fast adaptation, explicit treatment of transient dynamics. They nevertheless preserve the central architectural separation advanced here: implementation-specific mechanisms establish feasibility, whereas the traffic layer translates feasible spacing into road-level consequences.

IX. CONCLUSION

This work establishes the admissible time headway as a compact interface between longitudinal-control feasibility and macroscopic traffic performance. From a constant-time-headway equilibrium rule, the model obtains a complete software-induced fundamental diagram and closed-form expressions for capacity, critical density, and congestion-wave speed. Network-assisted coordination enters through the smallest headway compatible with safety, stability, latency, and comfort, so its road-level value is measured by the active constraint that it is able to relax.

The strength of the formulation is the separation of two tasks that are often embedded in one controller-specific model. Detailed control, communication, simulation, or experimental studies remain responsible for certifying a feasible headway; the proposed analytical layer translates that certificate into common traffic units. This makes alternative controllers and

communication topologies directly comparable while retaining a transparent account of the accompanying capacity and wave-propagation effects. Published simulation-based and experimental evidence independently supports the micro-to-macro construction and the equilibrium assumptions used by the model.

The framework is therefore useful as a first-order design, screening, and interpretation tool. It identifies when additional communication or control performance can produce a meaningful road-level gain, and it supplies a consistent baseline for subsequent studies of heterogeneous traffic, spatially varying conditions, and adaptive headway policies.

REFERENCES

[1] J. Ploeg, B. T. M. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *Proc. IEEE International Conference on Intelligent Transportation Systems*, 2011, pp. 260–265.

[2] V. Milanés and S. E. Shladover, "Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 285–300, 2014.

[3] Y. Bian, Y. Zheng, W. Ren, S. E. Li, J. Wang, and K. Li, "Reducing time headway for platooning of connected vehicles via V2V communication," *Transportation Research Part C: Emerging Technologies*, vol. 102, pp. 87–105, 2019.

[4] K. Ma, H. Wang, Z. Zuo, Y. Hou, X. Li, and R. Jiang, "String stability of automated vehicles based on experimental analysis of feedback delay and parasitic lag," *Transportation Research Part C: Emerging Technologies*, vol. 145, Art. no. 103927, 2022.

[5] J. Liu, Y. Zhou, and L. Liu, "Communication delay-aware cooperative adaptive cruise control with dynamic network topologies—A convergence of communication and control," *Digital Communications and Networks*, vol. 11, no. 1, pp. 191–199, 2025.

[6] M. Makridis, K. Mattas, A. Anesiadou, and B. Ciuffo, "OpenACC. An open database of car-following experiments to study the properties of commercial ACC systems," *Transportation Research Part C: Emerging Technologies*, vol. 125, Art. no. 103047, 2021.

[7] T. Li, D. Chen, H. Zhou, Y. Xie, and J. Laval, "Fundamental diagrams of commercial adaptive cruise control: Worldwide experimental evidence," *Transportation Research Part C: Emerging Technologies*, vol. 134, Art. no. 103458, 2022.

[8] M. Shang and R. E. Stern, "Impacts of commercially available adaptive cruise control vehicles on highway stability and throughput," *Transportation Research Part C: Emerging Technologies*, vol. 122, Art. no. 102897, 2021.

[9] M. J. Lighthill and G. B. Whitham, "On kinematic waves. II. A theory of traffic flow on long crowded roads," *Proceedings of the Royal Society of London. Series A*, vol. 229, no. 1178, pp. 317–345, 1955.

[10] P. I. Richards, "Shock waves on the highway," *Operations Research*, vol. 4, no. 1, pp. 42–51, 1956.

[11] C.-Y. Liang and H. Peng, "String stability analysis of adaptive cruise controlled vehicles," *JSME International Journal Series C*, vol. 43, no. 3, pp. 671–677, 2000.

[12] P. Y. Li and A. Shrivastava, "Traffic flow stability induced by constant time headway policy for adaptive cruise control vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 4, pp. 275–301, 2002.

[13] X. Li, "Trade-off between safety, mobility and stability in automated vehicle following control: An analytical method," *Transportation Research Part B: Methodological*, vol. 166, pp. 1–18, 2022.

[14] R. Donà, K. Mattas, Y. He, G. Albano, and B. Ciuffo, "Multianticipation for string stable adaptive cruise control and increased motorway capacity without vehicle-to-vehicle communication," *Transportation Research Part C: Emerging Technologies*, vol. 140, Art. no. 103687, 2022.

[15] X. Shi and X. Li, "Constructing a fundamental diagram for traffic flow with automated vehicles: Methodology and demonstration," *Transportation Research Part B: Methodological*, vol. 150, pp. 279–292, 2021.

[16] J. Zhou and F. Zhu, "Modeling the fundamental diagram of mixed human-driven and connected automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 115, Art. no. 102614, 2020.

[17] C. L. Melson, M. W. Levin, B. E. Hammit, and S. D. Boyles, "Dynamic traffic assignment of cooperative adaptive cruise control," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 114–133, 2018.

[18] J. Jiang, Y. Zhou, X. Wang, and S. Ahn, "On dynamic fundamental diagrams: Implications for automated vehicles," *Transportation Research Part B: Methodological*, vol. 189, Art. no. 102979, 2024.

From Control to Orchestration: Transformation of Data Governance in the AI Era

Snježana Rosandić, Edita Škec
Ericsson Nikola Tesla d.d.
Croatia
{snjezana.rosandic, edita.skec}@ericssonnikolatesla.com

Katarina Babić, Dinko Begušić
University of Split
Croatia
{krados, begusic}@fesb.hr

***Abstract*—Since 2020, artificial intelligence (AI) has transformed data governance from a largely control-based discipline into a more adaptive, metadata-driven, and automated capability. The widespread adoption of operational machine learning, generative AI, retrieval-augmented generation (RAG), and early agentic systems has changed how organizations govern data quality, metadata, lineage, access, compliance, and stewardship. This paper reviews the post-2020 evolution of data governance and discusses major changes in governance models, tools, and organizational roles. The research shows that AI both increases the need for better governance processes and enables governance automation through active metadata, policy-as-code, intelligent classification, and workflow orchestration. The paper concludes by identifying new challenges and future research directions for trustworthy, AI-driven enterprise systems.**

Keywords—artificial intelligence, data governance transformation, active metadata, policy-as-code, agentic AI, intent-based governance.

I. INTRODUCTION

Data governance has historically evolved in response to changes in enterprise data architecture, regulatory requirements, and business expectations. In earlier phases of enterprise computing, governance was primarily associated with database administration, access control, data ownership, and compliance. Over time, the emergence of data warehousing, big data platforms, cloud environments, and data catalogs expanded the governance scope from technical control to organizational coordination and broader policy arrangements for responsible data access, sharing, protection, and use [1]-[5].

In line with DAMA-DMBOK, data governance defines policies, standards, decision rights, accountabilities, and controls needed to manage data assets responsibly across their lifecycle [3]. In contrast, data management concerns the operational execution of data-related practices, including storage, integration, quality management, metadata management, and lifecycle administration. In practice, these governance principles become operational through mechanisms such as data catalogs, which provide a metadata-driven single source of truth for discovering data assets, documenting ownership, tracing lineage, assessing quality, and capturing usage constraints. Prior work on data catalogs and data monetization illustrates that the catalog also connects data assets with governance roles and responsibilities, making accountability visible and actionable across the enterprise [6]. AI governance extends this perspective to the responsible design, deployment, monitoring, and oversight of AI systems, including transparency, explainability, safety, bias, and accountability concerns [7], [8], [9], [10], [11].

This role becomes even more important in AI-enabled environments, where trustworthy model development, retrieval, automation, and compliance depend on well-described, governed, and traceable data assets. The growing use of artificial intelligence (AI) has introduced a qualitatively different phase in this evolution [2], [7]. AI has affected data governance in two directions simultaneously. First, AI systems depend on high-quality, well-described, secure, and compliant data [8], [12], thereby increasing pressure on governance maturity. Second, AI itself has begun to automate governance tasks such as classification, lineage discovery, anomaly detection, policy enforcement, and stewardship support [13], [14]. As a result, data governance is increasingly moving from static documentation and manual workflows toward dynamic, machine-assisted, and eventually autonomous operating models [12], [15], [16]. Although numerous studies examine specific governance challenges associated with machine learning, generative AI, active metadata, or agentic systems, a comprehensive review of how AI has transformed data governance as a discipline since 2020 remains limited.

The growing importance of data governance is reflected in the increasing volume of scientific publications on the topic, which reached 826 publications in 2023 [17]. This trend can be attributed to several key factors, including growing investment in research, accelerated technological progress, the effects of global events such as the COVID-19 pandemic, and the adoption of AI technologies in various domains.

A recent MIT study also shows that, despite $30–40 billion in enterprise investment in generative AI, nearly 95% of organizations are still seeing no measurable return [18]. This gap, referred to as the GenAI Divide, has been observed by both users and providers: only about 5% of AI pilots generate significant value, while the majority remain without profit-and-loss impact. The large gap between AI investment and realized value suggests that data governance maturity may be a critical prerequisite for successful AI adoption, in addition to advances in AI technologies themselves.

To address this gap, this narrative review synthesizes selected academic, regulatory, standards-based, and industry sources on the evolution of data governance in response to operational machine learning, generative AI, RAG, and emerging agentic systems. The analysis is conceptually grounded in the co-evolution of data architectures, governance practices, and AI paradigms, which are treated throughout this

review as mutually reinforcing drivers of organizational transformation.

The main argument is that post-2020 AI adoption has transformed data governance from a supporting control function into a central orchestration layer for trustworthy, scalable, and intelligent data use. In this context, orchestration refers to the coordinated execution of governance through metadata, policies, roles, automated controls, human oversight, and technical workflows across data and AI systems. AI-driven data governance cannot be treated as a standalone capability because it depends fundamentally on the maturity of underlying governance practices. Without structured, high-quality, and well-governed data assets, the application of AI, including its use in governance processes, remains limited or ineffective. Data governance must therefore first evolve to meet the requirements of AI systems before AI can be effectively applied to advance governance itself.

The remainder of this review is organized as follows. Section II discusses the transition from traditional to AI-driven data governance. Section III reviews the post-2020 shift across operational machine learning, generative AI, active metadata, and agentic systems. Sections IV and V synthesize the core areas of transformation and examine emerging directions, including distributed sovereignty and intent-based governance. Section VI discusses the main findings and future research directions, while Section VII concludes the paper.

II. From Traditional to AI-Driven Governance

Before 2020, many organizations still relied on data governance models centered on centralized ownership, manual stewardship, and static control mechanisms. In relational and warehouse-based environments, governance was often tied to database administrators, enterprise architects, and compliance teams, while metadata was incomplete, manually maintained, and disconnected from operational pipelines. Governance artifacts such as glossaries, ownership matrices, and policy documents were frequently maintained outside execution environments [3], [5].

The expansion of big data platforms, cloud ecosystems, distributed storage, schema-on-read practices, and self-service analytics gradually challenged this model. Traditional gatekeeping became insufficient, requiring governance to evolve from centralized restriction toward federated enablement. Although data catalogs, lineage tools, and stewardship workflows gained traction, they often remained only partially implemented or poorly integrated into daily data operations [19].

AI adoption after 2020 further intensified these tensions. AI systems require trustworthy training and inference data, while also introducing governance challenges related to explainability, bias, privacy, model risk, and accountability. Because AI systems improve only when they continuously ingest relevant and high-quality data from their environment [20], weak governance impedes feedback loops and prevents data from becoming truly AI-ready. Gartner research [21] shows that 63% of organizations lack confidence in their data readiness for AI and predicts that 60% of AI projects will fail to realize value by 2027 because of incohesive data governance. Similarly, a secondary summary of Gartner's 2025 AI Hype Cycle positions data readiness, ModelOps, AI trust, risk, and security management (AI TRiSM), and responsible AI governance as enabling disciplines required to move enterprise AI from experimentation toward scalable value realization [22]. Consequently, governance can no longer remain a retrospective control function; it must become continuous, embedded, and operational.

III. The Post-2020 Shift: AI as Both Data Governance Driver and Enabler

A. 2020–2022: Operational Machine Learning and Governance Acceleration

During the 2020–2022 period, operational machine learning became one of the first AI paradigms to substantially reshape data governance. Its adoption increased the volume, velocity, and complexity of enterprise data use, while also raising expectations for data quality, traceability, explainability, and regulatory compliance. As ML models were increasingly used for prediction, optimization, and decision support, organizations faced stronger pressure to ensure that training and operational data were accurate, representative, well-documented, and continuously monitored. In this sense, operational ML acted first as a driver of new governance requirements, exposing the limitations of manual data profiling, static rules, fragmented metadata, and retrospective compliance checks [13].

At the same time, the same ML capabilities that created new governance demands also became important instruments for improving governance execution. Machine learning techniques enabled more scalable data profiling, automated anomaly detection, metadata enrichment, lineage support, and compliance monitoring. Unlike traditional rule-based validation, unsupervised learning methods can identify previously unknown outliers, inconsistencies, and quality deviations without requiring all conditions to be predefined. This made governance controls more adaptive and better suited to dynamic enterprise environments in which data structures, sources, and usage contexts change frequently [13].

This dual role is particularly visible in data quality management and compliance-oriented classification. On the one hand, ML-based applications depend on reliable and well-governed data because poor data quality can undermine model accuracy, increase bias, and weaken regulatory accountability. On the other hand, ML-based anomaly detection and automated classification can help organizations identify data errors, categorize information according to sensitivity or regulatory relevance, and detect complex patterns that conventional rule-based controls may miss. For example, Random Forest and Support Vector Machine algorithms have been applied to classify personal data for GDPR-oriented compliance, while neural models have been used to discover more complex data patterns and relationships [14].

Operational ML also contributed to the emergence of predictive data governance. As governance requirements became more continuous and risk-based, ML models provided mechanisms for identifying potential data quality, compliance, or operational risks before they materialize. Predictive models based on time-series analysis, ensemble learning, or historical governance data can support proactive

risk management by forecasting likely compliance breaches, data quality incidents, or workflow deviations [14]. This shifted governance from retrospective control toward earlier detection, continuous monitoring, and anticipatory oversight.

The same pattern can be observed beyond narrowly defined governance tasks. Machine learning supported broader enterprise information management by improving information extraction, routing, and contextual decision support. AI-based frameworks combining natural language processing, machine learning, and knowledge graphs have been proposed to route information based on context, urgency, topic similarity, and historical patterns [23]. These developments show that ML did not only automate isolated governance functions but also helped embed governance logic into the flow of enterprise information and decision-making.

The relationship between ML and data governance therefore remained reciprocal. Operational ML intensified the need for mature governance foundations, including data quality, metadata completeness, lineage, interpretability, and compliance controls. At the same time, ML provided practical mechanisms for automating governance activities that had previously depended on manual inspection and static rules. Empirical work in the banking sector confirms this reciprocity by showing that stronger governance practices are associated with improved model accuracy, better interpretability, and fewer false positives in fraud-detection models [24]. The 2020–2022 period can therefore be understood as the first post-2020 stage in which AI became both a source of new governance challenges and a technological enabler of more automated, predictive, and continuously monitored data governance.

B. 2022–2024: Generative AI and the Rise of Active Metadata

The release of ChatGPT in November 2022 marked the transition of generative AI from research experimentation into mainstream enterprise use, reaching an estimated 100 million monthly active users within two months [25]. Enterprise adoption followed rapidly: by the middle of 2023, nearly one-third of surveyed organizations had integrated generative AI into at least one business function on a regular basis, while only about one in five had established a formal usage policy [26]. This gap between adoption speed and governance readiness became a defining feature of the period [27]. Concrete incidents, such as confidential data leaks at Samsung after employees copy-pasted sensitive material into ChatGPT, demonstrated that governance was no longer limited to controlling stored data, but also had to address what models could reveal, reproduce, or fabricate in response to prompts [28]. NIST's Generative AI Profile further formalized risks such as confabulation, privacy leakage, prompt injection, and intellectual property infringement [29].

A parallel architectural shift further intensified governance demands. Retrieval-Augmented Generation (RAG), first proposed as a method for grounding language model outputs in external knowledge sources [30], became a dominant pattern for deploying generative AI on proprietary enterprise data. By combining language models with vector indexes of organizational content, RAG can reduce hallucination and support more traceable responses, but it also brings governance directly into the technical pipeline. Embeddings, vector stores, document chunking, contextual metadata, access control, freshness, and provenance all influence the reliability and compliance of generated outputs.

Metadata management was therefore reconceptualized during this period. Traditional passive metadata, which was manually maintained and often disconnected from live pipelines, was increasingly replaced by active metadata: continuously analyzed metadata capable of triggering alerts, updating lineage, enforcing policies, and supporting automated governance actions [31]. This shift responded to a significant readiness gap, as a 2024 industry survey found that only 29% of technology leaders strongly agreed that their enterprise data met the quality, accessibility, and security standards required to scale generative AI [32]. Regulatory developments reinforced the same direction: China's Interim Measures for the Management of Generative AI Services required lawful training-data provenance and respect for intellectual property [33], while the European Union's AI Act negotiations emphasized transparency, provenance, and controls for general-purpose and foundation models [8].

Taken together, the 2022–2024 period reframed metadata from passive documentation into an operational substrate for data governance. Generative AI created governance obligations that static, manually maintained metadata could not satisfy, while RAG architectures and regulatory convergence increased the need for continuous, traceable, and machine-actionable records of how data is sourced, transformed, retrieved, and used. This shift provides the technical and organizational foundation for the later emergence of autonomous, agent-driven governance models.

C. 2024–Present: Agentic Systems and Early Autonomous Governance

Since 2024, agentic AI has increasingly become a major focus of AI governance because such systems can autonomously plan, select tools, retrieve contextual data, maintain memory, interact with external systems, and execute multi-step tasks with limited human input [16], [34]. Compared with generative AI systems that primarily produce outputs, agentic systems expand governance concerns to the full decision-execution chain, including agent identity, permissions, tool access, contextual inputs, memory, traceability, coordination, security, and downstream effects.

This shift requires governance to verify not only whether an AI-generated output is accurate or compliant, but also whether the actions that produced it were authorized, reproducible, traceable, and aligned with organizational intent [16], [34], [35]. In agentic AI, data pipelines provide context, enable tool use, support memory, enforce controls, and create traceability for autonomous actions [34], [35]. If these pipelines are not cataloged, versioned, monitored, and linked with lineage, organizations lose visibility into what data the agent consumed, which transformations occurred, which tools were invoked, and what output triggered an action, directly affecting auditability, reproducibility, and compliance [35].

Early autonomous governance therefore refers to mechanisms that can monitor, enforce, document, and adapt controls within data and AI workflows with limited manual intervention. Policy-as-code (PaC) operationalizes this shift

by translating governance requirements into executable, versioned, and testable rules that can be enforced at runtime across data access, lineage validation, privacy constraints, and agent actions [36]. Together, policy-as-code, active metadata, and continuous monitoring provide the technical foundation for governing agentic AI systems.

This marks an early movement toward autonomous governance, where governance becomes machine-actionable, continuously enforced, and embedded in AI-enabled enterprise systems. Agentic systems therefore represent a new phase in the evolution of data governance, shifting the focus from governing data and models toward governing autonomous, data-driven actions across complex enterprise ecosystems.

IV. Core Areas of Transformation in Data Governance

The transformation of data governance in the AI era can be understood across four macro dimensions. First, at the strategic level, the focus of governance is shifting from compliance-driven control toward trust, risk management, and value creation, positioning data governance as a key enabler of AI adoption and business outcomes rather than merely a regulatory necessity [2], [21], [22].

Second, at the operational level, governance has evolved from static, manual, and document-driven practices toward real-time, automated, and executable approaches, where policies are increasingly implemented as code and enforced continuously within data and AI pipelines, for example through policy-as-code and continuous monitoring [36].

Third, at the architectural level, traditional centralized governance models and siloed data environments are being replaced by federated governance structures and graph-based data ecosystems, supported by active metadata and end-to-end observability across data and ML pipelines [19], [31].

Finally, in terms of scope expansion, the boundaries of data governance have expanded beyond structured enterprise data. Governance now also includes models, prompts, embeddings, autonomous agents, and multimodal data such as text and images, requiring lifecycle oversight across the entire AI decision pipeline [29], [34], [35].

Table I summarizes this transformation by comparing traditional data governance before 2020 with AI-driven data governance after 2020 across key dimensions of purpose, execution, metadata use, accountability, scope, human oversight, and risk. Together, these dimensions show that AI-driven data governance is not merely an extension of existing governance practices, but a structural transformation of governance scope, execution mechanisms, and accountability models.

TABLE I. Evolution from Traditional to AI-Driven Data Governance

Dimension	Traditional data governance	AI-driven data governance
Primary purpose	Compliance, control, ownership, documentation	Trust, AI readiness, value creation, and risk-aware orchestration
Governance model	Periodic, manual, document-based	Continuous, automated, embedded, machine-actionable
Metadata role	Passive documentation, manually maintained	Active metadata for monitoring, lineage, enforcement, workflow automation
Policy execution	Manual interpretation and retrospective audits	Policy-as-code, runtime enforcement, automated compliance checks
Accountability model	Data owners, stewards, centralized governance bodies	Federated accountability across data, AI, risk, compliance, and platform roles
Governed scope	Structured enterprise data, reports, databases, and data warehouses	Data assets, models, prompts, embeddings, pipelines, agents, and AI-generated outputs
Human oversight	Human-led approval and manual exception handling	Human-in-the-loop validation of automated governance decisions and high-risk uses
Main risk	Poor quality, unclear ownership, compliance gaps	Untrusted AI outputs, data leakage, hallucination, bias, autonomous action risk

The comparison shows that the core shift is not only technological, but also organizational and operational. Governance moves from periodic control toward continuous orchestration embedded in data and AI workflows. However, despite increasing automation, AI-driven data governance still requires human-in-the-loop oversight for validating automated classifications, resolving policy exceptions, approving high-risk data uses, and maintaining organizational accountability.

V. Emerging Directions: Distributed Sovereignty and Intent-Based Governance

The "AI on the edge" paradigm shifts data processing from centralized cloud environments to distributed edge nodes, reducing latency but also limiting central visibility and control. This decentralization complicates traditional governance mechanisms such as lineage tracking, auditability, and compliance enforcement [37].

Distributed AI systems introduce increased risks related to privacy, fairness, and robustness because of fragmented data and execution environments [38]. Edge environments further amplify context dependency, as local data conditions vary significantly across nodes. Federated and edge learning approaches highlight challenges such as non-independent and identically distributed (non-IID) data and system heterogeneity, directly affecting governance [39]. As a result, governance must evolve from static, centralized models to distributed and embedded approaches.

Policy-as-code enables this shift by translating governance rules into executable controls that can be enforced across distributed pipelines and nodes. This ensures consistent enforcement of access control, privacy, and lineage requirements throughout the data lifecycle [36]. In combination with agentic AI, ungoverned data and pipelines pose significant decision risks because of reduced visibility and control. Therefore, policy-as-code becomes a critical enabler of trustworthy, auditable, and scalable governance in distributed edge AI systems [35], [36]. These distributed

conditions create the need for governance models that can express high-level intent while enforcing controls locally and dynamically across heterogeneous environments.

Intent-based governance (IBG) represents a potential governance paradigm for increasingly autonomous and distributed AI ecosystems, including highly autonomous future AI scenarios, where high-level organizational intents are automatically translated into executable policies that continuously guide and constrain autonomous agents, ensuring trustworthy, auditable, and aligned behavior at scale.

Recent literature on agentic generative AI and highly autonomous AI systems reinforces the need for distributed and intent-based governance models. Highly autonomous systems introduce risks related to recursive self-improvement, multi-agent coordination, and post-deployment behavior that cannot be governed through static policy documents alone. In distributed AI ecosystems, governance must therefore shift toward modular agent design, adaptive safety mechanisms, and coordinated oversight across organizational, technical, and regulatory boundaries. From a data governance perspective, this implies that lineage, provenance, access control, and policy enforcement must become continuous, machine-actionable controls embedded directly into data and AI pipelines. These requirements align closely with intent-based governance, where high-level organizational and regulatory intents are translated into executable policies that dynamically constrain autonomous agents across decentralized environments [16].

VI. Discussion and Future Research

The findings reviewed in this paper indicate that AI and data governance are evolving through a reinforcing co-evolutionary cycle rather than independent developments. Each new AI paradigm increases the demand for more mature governance capabilities, which enables organizations to deploy increasingly complex AI systems in a more trustworthy and scalable manner.

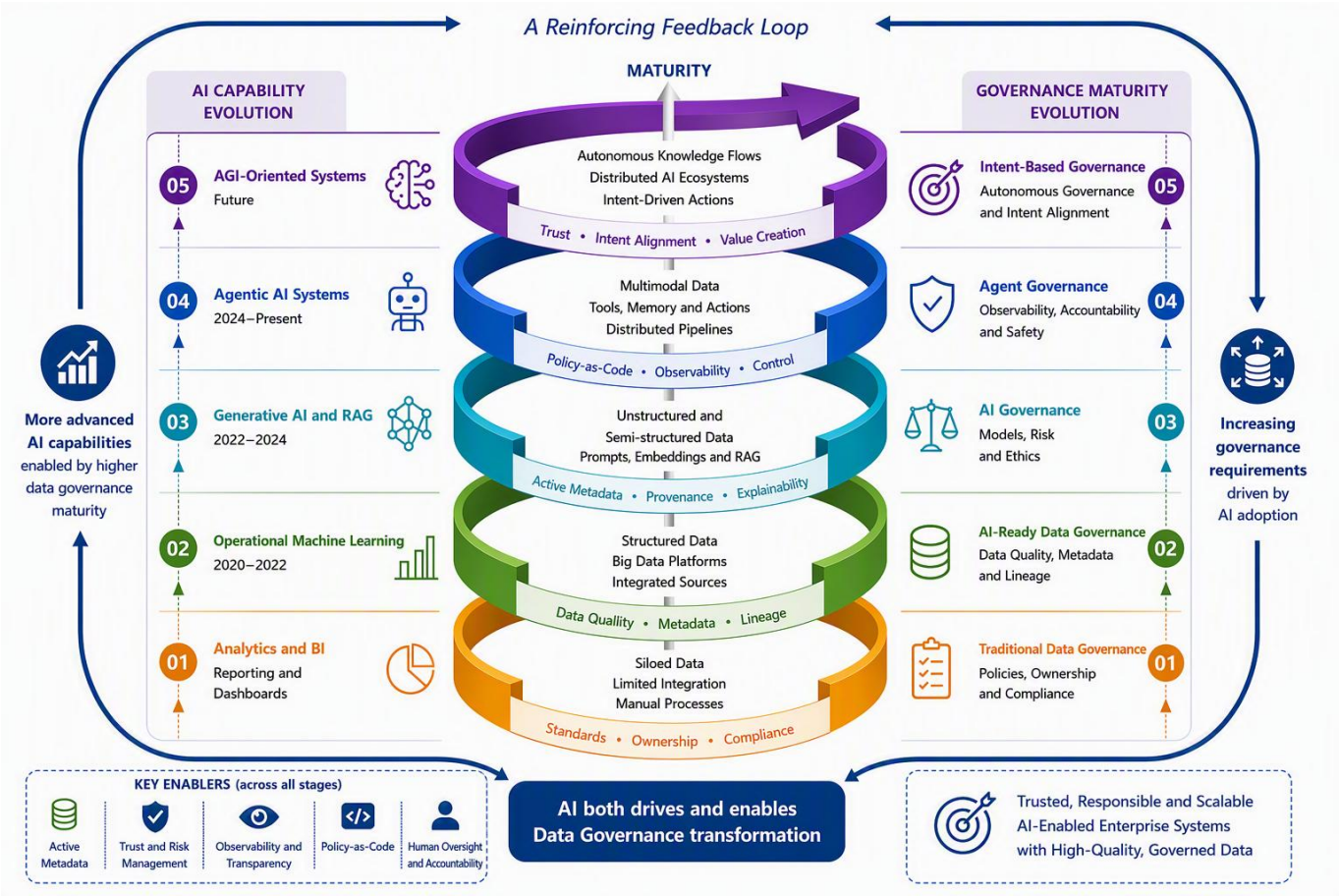


Fig. 1. AI-driven co-evolution of data governance maturity and AI capabilities.

Fig. 1 presents the authors' conceptual synthesis of the reviewed insights, illustrating the progression from traditional data governance toward AI-ready, autonomous, and intent-based governance models. The figure emphasizes a maturity trajectory in which governance capabilities evolve from manual ownership, documentation, and compliance controls toward active metadata, policy-as-code, automated monitoring, and ultimately intent-based coordination of autonomous agents. In this cycle, AI adoption increases the volume, diversity, and criticality of governed data use, while governance maturity determines whether advanced AI systems can be deployed safely, transparently, and with measurable organizational value.

The reviewed literature consistently suggests that AI does not reduce the need for data governance; rather, it exposes and

amplifies existing governance weaknesses. Poor data quality, incomplete metadata, weak lineage, unclear ownership, and fragmented policy enforcement become more consequential when data is used by generative or agentic AI systems. Risks such as data poisoning can transform data quality weaknesses into AI system vulnerabilities, making robust data governance controls essential for prevention and mitigation [40]. A major implication of the post-2020 transformation is therefore the shift from governance as documentation to governance as execution. Policies, classifications, access rules, compliance requirements, and accountability mechanisms increasingly need to be represented in machine-actionable form and enforced continuously across data, model, prompt, embedding, pipeline, and agent lifecycles. Active metadata emerges as a central governance substrate because it connects data assets, policies, pipelines, models, prompts, embeddings, outputs, and actions into an auditable and dynamically updated control layer. When these foundations are in place, data governance can also support data productization and AI monetization, creating new value streams for organizations [6], [41]. Agentic AI further extends governance beyond data and model management toward the governance of autonomous actions, requiring future frameworks to account not only for what data is used, but also which agents used it, under which intent, through which tools, and with what downstream effects.

Future research should move from conceptual discussion toward empirical validation of AI-driven governance models. In particular, further studies are needed on how organizations can measure AI-ready data governance maturity, implement policy-as-code across heterogeneous data platforms, and evaluate the effectiveness of active metadata in improving auditability, compliance, and model trustworthiness. Additional research is required on governance mechanisms for RAG pipelines, distributed AI environments, and agentic systems, where data, models, tools, prompts, embeddings, and decisions are increasingly interconnected.

Overall, the discussion indicates that the next stage of data governance research will need to integrate technical, organizational, and regulatory perspectives. The key research challenge is no longer only how to govern data, but how to govern dynamic data-driven ecosystems in which AI systems increasingly retrieve, interpret, generate, and act on data in semi-autonomous or autonomous ways. This requires governance models that are measurable, machine-actionable, auditable, and adaptable across both centralized and distributed enterprise environments.

Recent initiatives such as the European Commission's assessment of an EU-level registry for Text and Data Mining (TDM) opt-outs [42] further illustrate the growing importance of standardized metadata-based governance mechanisms for managing data provenance, intellectual property rights, and permissible data usage in AI ecosystems, reinforcing the convergence of data governance and AI governance as a central challenge for future research and practice.

VII. Conclusion

This review examined how the rapid adoption of artificial intelligence since 2020 has transformed data governance from a primarily control-oriented discipline into a dynamic, metadata-driven, and increasingly automated enterprise capability. The analysis shows that AI and data governance are evolving in a mutually reinforcing relationship: AI systems require mature governance foundations to operate reliably, while AI technologies increasingly enable governance automation through classification, lineage discovery, anomaly detection, active metadata, policy-as-code, and continuous monitoring.

The review also shows that the governance challenge has expanded beyond traditional data quality, ownership, and compliance concerns. With the rise of generative AI, RAG architecture, active metadata, and agentic systems, organizations must govern not only data assets but also prompts, embeddings, models, pipelines, tools, autonomous agents, and downstream actions. This expansion makes auditability, provenance, access control, policy automation, and governance observability central requirements for trustworthy AI-enabled enterprises.

Overall, the post-2020 transformation indicates that AI does not reduce the need for data governance; rather, it increases its strategic importance. Mature data governance becomes a prerequisite for scalable, trustworthy, and value-generating AI, while emerging governance mechanisms such as active metadata, policy-as-code, and intent-based governance point toward more autonomous and adaptive governance models. These findings suggest that data governance should be treated not as a supporting compliance layer, but as a core orchestration capability for AI-enabled enterprise transformation.

The main contribution of this review is the synthesis of post-2020 data governance transformation as a staged evolution driven by successive AI paradigms: operational machine learning, generative AI and RAG, active metadata, and emerging agentic systems. By organizing the literature along this trajectory, the paper shows that data governance is becoming an execution-oriented capability embedded directly into data and AI pipelines, rather than a static organizational control layer. For practitioners, this implies that investments in AI should be accompanied by measurable improvements in data quality, metadata completeness, lineage, ownership, policy automation, and governance observability. For researchers, the review highlights the need for empirically validated maturity models, governance metrics, and architectural patterns capable of supporting trustworthy AI in distributed and increasingly autonomous enterprise ecosystems. In addition to synthesizing the literature, the review offers an evaluative comparison framework for distinguishing traditional, AI-ready, and AI-driven data governance capabilities across dimensions such as purpose, metadata role, policy execution, accountability, automation, scope, and human oversight.

References

[1] W. Alhammad et al., “The impact of AI on data governance practices,” Journal of Decision Systems, vol. 35, no. 1, 2026, doi: 10.1080/12460125.2026.2675043.

[2] KPMG, “Data Governance in the Age of AI,” KPMG, New York, NY, USA, 2025. [Online]. Available: https://kpmg.com/kpmg-us/content/dam/kpmg/pdf/2025/data-governance-age-ai.pdf.

[3] DAMA International, DAMA-DMBOK: Data Management Body of Knowledge, 2nd ed. Basking Ridge, NJ, USA: Technics Publications, 2017.

[4] OECD, Going Digital to Advance Data Governance for Growth and Well-being, OECD Publishing, Paris, 2022, doi: 10.1787/e3d783b0-en.

[5] IEEE, Big Data Governance and Metadata Management: Standards Roadmap, 2020. [Online]. Available:

https://www.nist.gov/system/files/documents/2025/03/18/BDGMM_Standards_Roadmap-2020.pdf.

[6] S. Rosandić, "The Role of Data Catalog in Advancing Data Monetization Strategy," in Proc. 48th ICT and Electronics Convention (MIPRO), Opatija, Croatia, 2025, pp. 1164–1169, doi: 10.1109/MIPRO65660.2025.11131961.

[7] OECD, "AI, Data Governance and Privacy," 2024. [Online]. Available: https://securesustain.org/wp-content/uploads/2024/07/AI-data-governance-and-privacy-Synergies-and-areas-of-international-co-operation.pdf.

[8] Artificial Intelligence Act (Regulation (EU) 2024/1689), Official Journal version of 13 June 2024. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=OJ:L_202401689.

[9] E. Tabassi, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," NIST AI 100-1, National Institute of Standards and Technology, 2023, doi: 10.6028/NIST.AI.100-1.

[10] ISO/IEC 42001:2023, Information technology — Artificial intelligence — Management system, ISO/IEC, Geneva, Switzerland, 2023.

[11] ISO/IEC 23894:2023, Information technology — Artificial intelligence — Guidance on risk management, ISO/IEC, Geneva, Switzerland, 2023.

[12] W. Yang et al., "Impact of modern AI in metadata management," 2025. [Online]. Available: https://link.springer.com/article/10.1007/s44230-025-00106-5.

[13] P. Mahendra, P. Doshi, A. Verma, and S. Shrivastava, "A Comprehensive Review of AI and ML in Data Governance and Data Quality," in Proc. 2025 3rd International Conference on Inventive Computing and Informatics (ICICI), Bangalore, India, 2025, pp. 1–6, doi: 10.1109/ICICI65870.2025.11069464.

[14] J. Osakwe, M. Ziezo, and I. Waiganjo, "AI-Driven Approaches for Optimising Data Governance in Large Enterprises: A Systematic Literature Review," in Proc. 2025 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), Windhoek, Namibia, 2025, pp. 205–209, doi: 10.1109/ETNCC66224.2025.11299748.

[15] Gartner, "Take a Modern Approach to Data Governance," Gartner Research, Stamford, CT, USA, 2025. [Online]. Available: https://www.gartner.com.

[16] S. Joshi, "AI Governance in the Era of Agentic Generative AI and AGI: Frameworks, Risks, and Policy Directions," International Journal of Innovative Research in Computer Science and Technology, vol. 13, no. 5, pp. 14–24, 2025, doi: 10.55524/ijircst.2025.13.5.3.

[17] H. Hassani, X. Huang, and S. MacFeely, "Mapping the evolution of data governance scientific research," Data & Policy, vol. 7, Art. no. e51, 2025, doi: 10.1017/dap.2025.10014.

[18] MIT Center for Information Systems Research, "State of AI in Business 2025: Preliminary Findings from Project NANDA," MIT CISR, Cambridge, MA, USA, 2025. [Online]. Available: https://mlq.ai/media/quarterly_decks/v0.1_State_of_AI_in_Business_2025_Report.pdf.

[19] K. Burchardi, D. Nath, and J. Lan, "Federated Data Governance Model," Boston Consulting Group (BCG), Boston, MA, USA, 2024. [Online]. Available: https://media-publications.bcg.com/Federated-Data-Governance-Model.pdf.

[20] O. V. Egwuatu, "Ethical and Governance Challenges of AI in Information Systems: Toward Responsible Adoption in Enterprise Systems," World Journal of Advanced Research and Reviews, vol. 27, no. 2, pp. 1593–1603, Aug. 2025, doi: 10.30574/wjarr.2025.27.2.3064.

[21] R. Edjlali, "Lack of AI-Ready Data Puts AI Projects at Risk," Gartner Research, Stamford, CT, USA, 2025. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2025-02-26-lack-of-ai-ready-data-puts-ai-projects-at-risk.

[22] LemonVolt, "Gartner's AI Hype Cycle 2025: Cutting Through the Noise to Real Use Cases," 2025. [Online]. Available: https://cdn.prod.website-files.com/68e2953718576ae8097b7cfd/68efaff129a48a7e8d0fdde3_Gartner%27s%20AI%20Cycle%202025.pdf.

[23] K. Saripudi, "Intelligent Information Flow: Leveraging Artificial Intelligence for Real-Time Decision-Making in Business Management Systems," in Proc. 2026 1st International Conference on Artificial Intelligence & Machine Learning in Communication and Power Systems (AIMLCPS), Berhampur, India, 2026, pp. 1–6, doi: 10.1109/AIMLCPS68702.2026.11542365.

[24] S. Singamsetty, "Efficacy of Data Governance: A Cutting-Edge Approach to Ensuring Data Quality in Machine Learning for the Banking Industry," in Proc. 2024 2nd International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi Campus, Centurion University of Technology and Management, Odisha, India, 2024, pp. 1–7, doi: 10.1109/SCOPES64467.2024.10991944.

[25] K. Hu, "ChatGPT sets record for fastest-growing user base – analyst note," Reuters, Feb. 2, 2023. [Online]. Available: https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/.

[26] McKinsey & Company, "The state of AI in 2023: Generative AI's breakout year," 2023. [Online]. Available: https://www.mckinsey.com/~/media/mckinsey/business%20functions/quantumblack/our%20insights/the%20state%20of%20ai%20in%202023%20generative%20ais%20breakout%20year/the-state-of-ai-in-2023-generative-ais-breakout-year_vf.pdf.

[27] A. Taeihagh, "Governance of generative AI," Policy and Society, vol. 44, no. 1, pp. 1–22, Jan. 2025, doi: 10.1093/polsoc/puaf001.

[28] TechCrunch, "Samsung bans use of generative AI tools like ChatGPT after April internal data leak," May 2, 2023. [Online]. Available: https://techcrunch.com/2023/05/02/samsung-bans-use-of-generative-ai-tools-like-chatgpt-after-april-internal-data-leak/.

[29] NIST, "Artificial Intelligence Risk Management Framework: Generative AI Profile," NIST AI 600-1, Jul. 2024, doi: 10.6028/NIST.AI.600-1.

[30] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Advances in Neural Information Processing Systems, vol. 33, 2020, arXiv:2005.11401.

[31] Gartner, "Market Guide for Active Metadata Management," Gartner Research, Stamford, CT, USA, 2022. [Online]. Available: https://www.gartner.com/en/documents/4004082.

[32] IBM Institute for Business Value, "What Is AI-Ready Data?" IBM Corporation, Armonk, NY, USA, 2024. [Online]. Available: https://www.ibm.com/think/topics/ai-ready-data.

[33] Cyberspace Administration of China, Interim Measures for the Management of Generative AI Services, 15th Decree, 2023. [Online]. Available: https://www.cac.gov.cn/2023-07/13/c_1690898327029107.htm.

[34] M. Abou Ali, F. Dornaika, and J. Charafeddine, "Agentic AI: A comprehensive survey of architectures, applications, and future directions," Artificial Intelligence Review, vol. 59, no. 11, 2026, doi: 10.1007/s10462-025-11422-4.

[35] T. Miller, "Controlled Agentic AI Systems: A Governance-Driven Architecture for Auditable and Reproducible Decision Pipelines," Machine Learning and Knowledge Extraction, vol. 8, no. 5, Art. no. 125, 2026, doi: 10.3390/make8050125.

[36] G. Vamshi Krishna, "Policy-as-Code for AI Data Platforms: Enforcing Privacy, Lineage, and Access Control End-to-End," 2024, doi: 10.54660/GMPJ.2024.1.6.228-232.

[37] S. Duan et al., "Distributed Artificial Intelligence Empowered by End-Edge-Cloud Computing: A Survey," IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 591–624, First Quarter 2023, doi: 10.1109/COMST.2022.3218527.

[38] W. Wei and L. Liu, "Trustworthy Distributed AI Systems: Robustness, Privacy, and Governance," arXiv:2402.01096, 2024. [Online]. Available: https://arxiv.org/pdf/2402.01096.

[39] Z. Wang et al., "Federated Continual Learning for Edge-AI: A Comprehensive Survey," arXiv:2411.13740, 2024. [Online]. Available: https://arxiv.org/pdf/2411.13740.

[40] A. Souly et al., "Poisoning Attacks on LLMs Require a Near-constant Number of Poison Samples," arXiv:2510.07192, 2025. [Online]. Available: https://arxiv.org/abs/2510.07192.

[41] McKinsey, "Global Tech Agenda," 2026. [Online]. Available: https://www.mckinsey.com/capabilities/mckinsey-technology/our-insights/mckinsey-global-tech-agenda-2026.

[42] European Commission: Directorate-General for Communications Networks, Content and Technology, Visionary Analytics, and Capgemini Invent, "Study to assess the feasibility of a registry of text and data mining opt-outs expressed by rightholders," Final report, Publications Office of the European Union, 2026, doi: 10.2759/4959861.

Approximate integration defined in Java and Python

Mauro Pullin
Department of Neuroscience
University of Padova
Via VIII Febbraio, 2 – 35122 Padova, Italy
mauro.pullin@unipd.it

***Abstract* – This work describes how to calculate the definite integral of a function, which cannot be integrated analytically, using the rectangle method and the trapezoidal method. The computer implementation of the analyzed algorithms using the Java and Python programming languages is then discussed.**

Keywords – definite integral, indefinite integral, numerical integration, rectangular integration, rectangle method, trapezoidal integration, trapezoid method, method of parabolas.

I. INTRODUCTION

A. Premise

In Italy, first-year university students attended a variety of high schools, where they were not always taught the fundamentals of computer science, and programming in particular. Students who take a computer science course on programming languages in their first year of university learn about some important concepts, among which are conditional structures and iterative loops (Savitch, 1997, *[4]*) (Downey, 2002-2012 *[11]*). An iterative loop is composed of a set of instructions that must be executed multiple times to complete an operation. It's very instructive to analyze how iterative loops work by solving real-world problems, such as those originating in mathematics. In general, an iterative loop can be of three types (Savitch, 1997, *[4]*): *"for"* loop, or a loop with a counter; *"while"* loop, or a loop with a conditional at the beginning; *"do...while"* loop, or a loop with a conditional at the end.

The approximate calculation of definite integrals is one of the problems that can be solved with the help of computer science and, in particular, with the use of iterative loops.

We recall that the definite integral between a and b of the function y=f(x) represents the area (with sign) of a trapezoid (figure 1) (Mbobi, 2012, *[10]*).

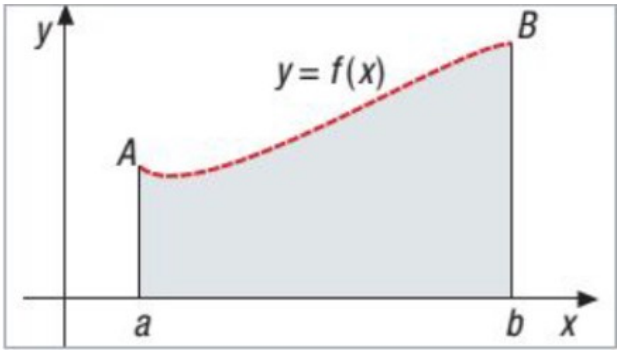


Figure 1: defined integration of a function

The fundamental theorem of integral calculus allows, under appropriate assumptions, to calculate the value of the definite integral from the antiderivative F(x) of the function f(x), which is determined through indefinite integration.

We can write (Bergamini & Barozzi, 2024 *[5]*):

$$\int_a^b f(x)\,dx = F(b) - F(a)$$

where $F(x) = \int f(x)\,dx$

It may happen, however, that the primitive F(x) cannot be determined using analytical methods.

For example, the following indefinite integrals cannot be calculated:

$$\int e^{-x^2}\,dx = ?$$

$$\int (\sin x) / x\,dx = ?$$

In these cases, it is necessary to resort to numerical integration (Camagni & Nikolassy, 2023 *[6]*).

B. Article organization

The article analyzes three methods for the approximate calculation of definite integrals: the rectangle method, the trapezoid method, and the parabola method. The paper concludes by analyzing the teaching motivations underlying the work, including the development of student self-esteem.

II. THE METHOD OF RECTANGLES

A. Introduction

Teaching experience has shown that studying and implementing numerical integration methods promotes both the learning of computer science and a better understanding of mathematical concepts (Bergamini & Barozzi, 2024 *[5]*).

In the rectangle method, the region to be integrated is divided into n rectangular parts all having the same width h=(b–a)/n, where h is called the integration step.

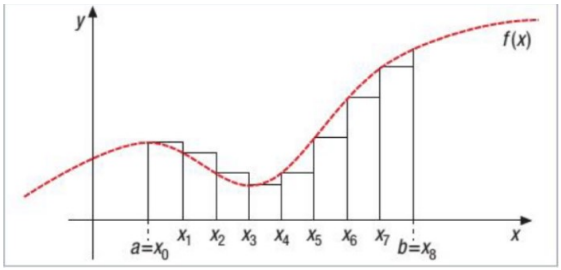


Figure 2: numerical integration using the rectangle method

Unless there is an error, the sum of the areas of the rectangles equals the area under the curve with the equation y=f(x). It is intuitively understood that the error decreases as the number n of intervals increases.

B. Computer implementation in Java language

We know we live in an increasingly digital world, but teaching programming isn't just about preparing future computer science experts. It's, above all, about teaching students how to reason. Programming means breaking down a complex problem into simpler parts, identifying relationships, and building solutions step by step. In other words, it means developing logical thinking, analytical skills, and a critical mind.

So we ask ourselves another question: why do we teach and learn Java as the first programming language? Java is one of the most widespread and established languages in the world. It is used in industry, in large applications, and in enterprise systems. But its value lies not only in its widespread use: it lies in its structure (Savitch, 1997, *[4]*). Java is an object-oriented language. This means it teaches students how to model reality through concepts like classes and objects. In practice, it helps students represent the world in an abstract and organized way. It's not just about writing code, but about building mental models. One of the most important aspects of Java, from an educational perspective, is its rigor. Java requires precision. It doesn't allow easy shortcuts. Every instruction must be clear, complete, and correct. This may seem like a limitation, especially at the beginning, but it's actually a great educational resource. It forces you to think before writing, to organize your ideas, and to follow a logic.

Furthermore, Java introduces fundamental concepts of modern computing, such as encapsulation, inheritance, and polymorphism. These words may seem complex, but they represent powerful tools for organizing your thoughts and managing complexity (Savitch, 1997, *[4]*).

Another important element is modularity. In Java, code is organized into parts, modules, that can be reused. This teaches a fundamental principle: you don't start from scratch every time, but build on what already exists. This approach is valid not only in programming, but in every area of knowledge. Of course, let's be honest: Java isn't the easiest language to get started with. The syntax can be complex, and compared to other, more straightforward languages, it can seem less intuitive. But this initial complexity is precisely what provides a training ground. Learning Java means training the mind for precision, discipline, and clarity. This doesn't mean that teaching should be rigid or theoretical. On the contrary: to be effective, it must be concrete, engaging, and progressive. It's important to start with simple examples, related to students' real-world experiences, and gradually build more advanced skills. Practical projects, small games, and simulations are essential tools for making learning active and meaningful. Java also offers a great advantage: it's a language that stays with students over time. What you learn in university holds true in research and the workforce. It's not just a teaching exercise, but a marketable skill. Java isn't just a programming language. It's an educational tool. It's a training ground for thinking. It teaches you to be precise, to be rigorous, but also to be creative in finding solutions (Savitch, 1997, *[4]*). Perhaps this is precisely the point: teaching Java isn't about teaching how to write code. It's about teaching how to build solutions, how to organize your thoughts, and how to better understand the complex world we live in.

For the computer implementation of the analyzed algorithm, the programming language Java and the programming language Python was chosen. Let's see below the source code related to the integration of Archimedes' parabola (which would be integrable exactly).

```
import java.util.Scanner;
import java.math.*;
public class IntegrazioneRettangolare {
   public static void main(String args[]) {
      Scanner tastiera = new Scanner(System.in);
      double a = -1.0, b = 1.0, N, h, k, x, y, AreaTot = 0.0, AreaRett;
      do {
         System.out.print("Insert N: ");
         N = tastiera.nextDouble();
         if (N <= 0) {
            System.out.println("N is wrong, it must be positive");
         }
      } while(N <= 0);
      System.out.print("Insert the lower bound a: ");
      a = tastiera.nextDouble();
      do {
         System.out.print("Insert the upper bound b: ");
         b = tastiera.nextDouble();
         if (b < a) {
            System.out.println("Incorrect upper bound");
         }
      } while(b < a);
      h = (b - a) / N;
      for(k = 0; k < N; k++) {
         x = a + k * h;
         y = AltezzaParabola(x);
         AreaRett = h * y;
         AreaTot = AreaTot + AreaRett;
      }
      System.out.println("Result = " + AreaTot);
   }
   public static double AltezzaParabola(double x) {
      double y;
      y = 1 - Math.pow(x,2);
      return y;
   }
}
```

C. Computer implementation in Python language

Programming is often thought of as learning to use a computer in an advanced way. In reality, it's much deeper. Programming means learning to think. It means taking a complex problem, breaking it down into simpler parts, finding a logical sequence of steps, and checking whether the solution works. In other words, teaching programming means developing fundamental skills: logical thinking, analytical skills, problem-solving, but also patience and precision. At this point, a second question arises: what second language can be used to begin this journey?

In recent years, Python has established itself as one of the most effective teaching tools (Downey, 2002-2012 *[11]*). And that's no coincidence. Python is a language designed to be simple, readable, and close to human language. Its code is clean, essential, and free of superfluous elements. This allows students to focus on what really matters: reasoning. When a student writes their first lines of code in Python, they don't immediately encounter a syntax barrier. They don't have to remember too many complex rules. Instead, they can quickly see the result of what they've written. This is a fundamental aspect: immediate feedback. The ability to achieve results quickly increases motivation, reduces frustration, and makes learning more effective. The student doesn't feel stuck, but supported. Another great value of Python is its versatility. It's not just an educational language: it's a real tool, used in a wide range of fields, from data science to artificial intelligence, from web development to automation. This means that what is learned in school isn't confined to a theoretical exercise, but can be applied in concrete, current contexts.

From an educational perspective, Python has a decisive advantage: it reduces cognitive load. Students don't have to expend energy understanding complex syntax, but can focus on the problem at hand. This way, the focus of learning returns to thinking. Python is therefore a tool that makes programming accessible. It lowers the entry threshold without lowering the level. It allows a greater number of students to approach these skills, even without advanced technical training (Downey, 2002-2012 *[11]*). Naturally, it's important to be balanced. Python is not without limitations. Precisely because it is simple, it can sometimes obscure the complexity of its underlying mechanisms. If used without adequate guidance, it can lead to superficial learning. For this reason, the role of the teacher is fundamental. Choosing the right language isn't enough: it's necessary to build an effective teaching path. A path that starts with concrete problems, that engages students, and that encourages experimentation. It's helpful to propose practical activities, small projects, and real-world situations. For example, analyzing simple data, creating small games, or automating everyday tasks. This way, students immediately see the meaning of what they're doing.

Another important aspect is collaborative work. Programming isn't just an individual activity: it's also about sharing, comparing, and collectively building solutions. This develops fundamental transversal skills. I'd also like to emphasize an often overlooked point: Python shouldn't be seen as a point of arrival, but as a starting point. It's an ideal tool for getting started, building confidence, and developing the foundations. These skills can then be transferred to other, more structured languages.

In this sense, Python and other languages, like Java, are not in competition, but complementary. Python opens the door, Java helps structure and consolidate.

I'd like to conclude this paraghaphwith a reflection. Teaching Python isn't simply about teaching a programming language. It means offering students a tool to better understand reality, to tackle complex problems, and to develop critical and structured thinking (Downey, 2002-2012 *[11]*). In an increasingly complex world, these skills are essential. And Python, with its simplicity and power, represents one of the most effective ways to begin this journey.

For the computer implementation of the analyzed algorithm, also the programming language Python was chosen. Here is a function written in Python that is used to implement the rectangle method:

```
def rettangoli(f, a, b, n):
    dx = (b - a) / n
    area = 0
    for i in range(n):
        x = a + i * dx
        area += f(x) * dx
    return area
```

III. THE TRAPEZOIDAL METHOD

A. Introduction

In the trapezoidal method, the interval of integration [a,b] is divided into n equal parts, and the area of the trapezoids with height h=(b–a)/n and bases given by the value of f(x) at the ends of each subinterval is calculated.

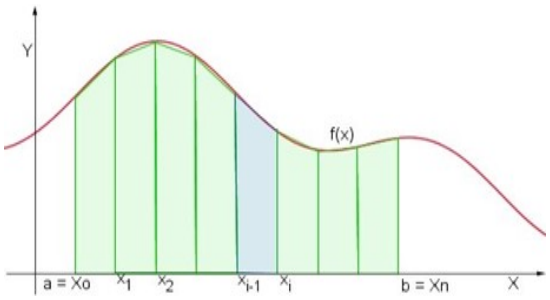


Figure 3: numerical integration using the trapezoidal method

B. Implementation in Java

Let's see below the source code related to the trapezoidal integration of Archimedes' parabola.

```
import java.util.Scanner;
import java.math.*;
public class IntegrazioneTrapezoidale {
   public static void main(String args[]) {
      Scanner tastiera = new Scanner(System.in);
      double a = -1.0, b = 1.0, N, h, k, x, y1, y2, AreaTot = 0.0, AreaTrap;
      do {
         System.out.print("Insert N: ");
         N = tastiera.nextDouble();
         if (N <= 0) {
            System.out.println("N is wrong, it must be positive");
         }
      } while(N <= 0);
      System.out.print("Insert the lower bound a: ");
      a = tastiera.nextDouble();
      do {
         System.out.print("Insert the upper bound b: ");
         b = tastiera.nextDouble();
         if (b < a) {
            System.out.println("Incorrect upper bound");
         }
      } while(b < a);
      h = (b - a) / N;
      for(k = 0; k < N; k++) {
         x = a + k * h;
         y1 = AltezzaParabola(x);
         x = a + (k + 1) * h;
         y2 = AltezzaParabola(x);
         AreaTrap = (y1 + y2) * h / 2;
         AreaTot = AreaTot + AreaTrap;
      }
      System.out.println("Result = " + AreaTot);
```

```
    }
    public static double AltezzaParabola(double x) {
        double y;
        y = 1 - Math.pow(x,2);
        return y;
    }
}
```

C. *Computer implementation in Python language*

For the computer implementation of the analyzed algorithm, also the programming language Python was chosen (Downey, 2002-2012 *[11]*). Here is a function written in Python that is used to implement the trapezoidal method:

```
def trapezi(f, a, b, n):
    dx = (b - a) / n
    area = (f(a) + f(b)) / 2
    for i in range(1, n):
        x = a + i * dx
        area += f(x)
    return area * dx
```

IV. THE METHOD OF PARABLES

A. *Introduction*

As for approximate integration using the parabola method, the underlying idea is surprisingly simple, and also very elegant. If a curve is complicated, we can try replacing it, at least locally, with something simpler. In this case, we don't use a straight line, but a parabola.

Why a parabola? Because a parabola is the simplest polynomial that can "follow" the curve of a curved function well. With three points, we can determine a parabola, and this parabola can approximate the function much more accurately than a simple straight line.

The method works like this: divide the interval into subintervals, take three points at a time, and construct a parabola that passes through these points. Then calculate the area under the parabola, which is easy to determine, and add all the contributions. The result is an estimate of the definite integral.

The method of parabolas is a perfect example of how mathematics addresses complexity: not by always seeking exact solutions, but by building increasingly precise approximations. This approach is fundamental not only in mathematics, but in all sciences. In physics, engineering, and economics, we often can't have an exact solution. But we can have a good enough solution. And often, that's exactly what's needed.

From an educational perspective, the method of parabolas is incredibly valuable. First, it bridges geometry and analysis. Students see concretely how a curve can be approximated by simpler shapes. It's not just a calculation, it's a visual representation. Second, it introduces the concept of error. Every approximation involves an error, but this error can be controlled, reduced, and estimated. It's a fundamental idea: there's not just right or wrong, but also "how close to right." Furthermore, this method develops algorithmic thinking. It's a procedure: divide, calculate, add. It's a repeatable, structured process, very similar to the way computers work. And it's no coincidence that methods like this are the basis of numerical computation, that part of mathematics that allows computers to solve real-world problems.

I'd also like to emphasize a cultural aspect. The parabola method demonstrates a living, dynamic, adaptable mathematics. It's not just a set of formulas to memorize, but a language for describing and addressing reality.

In conclusion, we can say that the parabola method isn't just a technique for calculating integrals. It's an example of how mathematics teaches us to think: to simplify without trivializing, to approximate without losing rigor, to find solutions even when perfection isn't possible. And perhaps this is its most important lesson.

Together with the Rectangle and Trapezoid methods, it constitutes the set of Numerical Integration methods known as Newton-Cotes methods, whose characteristic is the subdivision of the integration interval [a,b] into n equal parts. While in the other two methods, the number n of subdivisions can be any integer, for Simpson's method, the number n of subdivisions must necessarily be an even number.

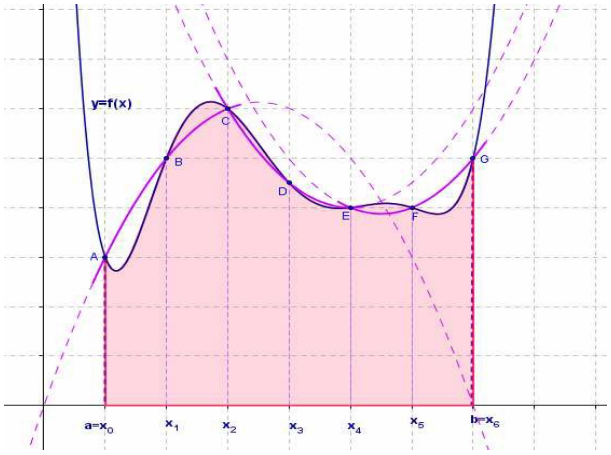


Figure 4: numerical integration using the method of parables

Let us therefore assume that we have divided the integration interval [a,b] into n equal parts (n being an even integer) and that we have calculated the ordinates corresponding to all the "nodes" x_i.

That is, let the coordinates of the points A(a;f(a)), B(x_1;f(x_1)), C(x_2;f(x_2)), D(x_3;f(x_3)), ...

In the rectangle and trapezium methods, we discussed "construction of rectangles and trapeziums" for each of the subdivision intervals. We could also have said that, in each interval, the integrand function f(x) is replaced by a more elementary function: by the function y=k (a constant function) for the rectangle method, and by the function y=mx+p (a linear function) for the trapezium method.

Well, in Simpson's method, the graph of the integrand function y=f(x), present in two consecutive intervals, is replaced with the graph of the parabola that has three points in common with the graph of f(x); otherwise, the procedure is similar. The final Cavalieri-Simpson formula for the entire integration interval [a,b], which allows us to calculate the definite integral of f(x)dx between the endpoints a and b, is as follows:

$$Area = \frac{\Delta x}{3}\left[f(a) + f(b) + 2 \cdot \sum_{i=1}^{k-1} f(x_{i\ pari}) + 4 \cdot \sum_{i=1}^{k} f(x_{i\ dispari}) \right]$$

N.B.: For ease of writing, let k=n/2 (n even; pari = even, dispari = odd)

B. IT implementation

Together with the Rectangle and Trapezoid methods, it constitutes the set of Numerical Integration methods known as Newton-Cotes methods, whose characteristic is the subdivision of the integration interval [a,b] into n equal parts. While in the other two methods, the number n of subdivisions can be any integer, for Simpson's method, the number n of subdivisions must necessarily be an even number.

For educational purposes, it's best to let students do the implementation work in Java or Python. They can be provided with the initial portion of the program, which they will then have to complete. As an example, the following is what can be given to the class:

```
do {
   System.out.print("Insert the first extreme of integration: ");
   a = tastiera.nextDouble();
   System.out.print("Insert the second extreme of integration: ");
   b = tastiera.nextDouble();
   if (b <= a) {
      System.out.println("Incorrect integration extremes ");
   }
} while(b <= a);
do {
   System.out.print("Enter n as positive and even: ");
   n = tastiera.nextInt();
   if (n <= 0 || n % 2 != 0) {
      System.out.println("Incorrect value of n ")
   }
} while(n <= 0 || n % 2 != 0);

// calculating the amplitude h of an interval
h = (double) (b - a) / n;

// to be completed by students...
…
```

C. Computer implementation in Python language

For the computer implementation of the analyzed algorithm, also the programming language Python was chosen (Downey, 2002-2012 *[11]*).

Here is a function written in Python that is used to implement the parables method:

```
def simpson(f, a, b, n):
   if n % 2 != 0:
      raise ValueError("n deve essere pari")

   dx = (b - a) / n
   area = f(a) + f(b)

   for i in range(1, n):
      x = a + i * dx
      if i % 2 == 0:
         area += 2 * f(x)
      else:
         area += 4 * f(x)

   return area * dx / 3
```

V. Educational insights

A. Introduction

Below, we would like to explore the educational aspects of this work in greater detail. First, it is important to help students understand the geometric meaning of the integral defined as the area under a curve. Furthermore, it is important to understand why, in many cases, the integral cannot be solved analytically and numerical methods are used.

Three methods of approximating the integral were examined and compared: rectangles, trapezoids and parabolas. It is important to encourage students to consider how the quality of the approximation depends on the choice of subdivision and the characteristics of the function. The approach described in this work fosters the connection between visual and numerical reasoning, developing the relationship between abstract concepts and practical calculation.

The teaching activities described in this work have proven to be very useful for deepening both the mathematical and computer science aspects. An interesting study to propose to students at this point could consist in studying the behavior of the approximation error as a function of the number of evaluations of the integrand function required.

For a long time, mathematics has been associated with the idea of a precise, elegant solution, enclosed in a formula. However, in real applications, things are often different. Many problems do not have an exact solution expressible in a simple form. Others are too complex to be solved with analytical methods. And this is where numerical computation comes in. Numerical computation is the branch of mathematics concerned with finding approximate solutions to complex problems, using algorithmic procedures and computational tools, often implemented on computers. In other words, we don't always seek the perfect solution: we seek a sufficiently precise solution, obtained efficiently. We could say that numerical computation is the bridge between theoretical mathematics and the real world. To better understand its importance, we can consider a simple example: the calculation of an integral. In many cases, it is not possible to find a primitive with classical techniques. Or consider the solution of nonlinear equations, the simulation of physical phenomena, or the prediction of economic models. In all these cases, numerical computation becomes indispensable.

A fundamental aspect is that numerical computation is not limited to "doing calculations." It introduces a new way of thinking: algorithmic thinking. It means breaking down a problem into elementary steps, defining a procedure, and repeating it until an increasingly accurate result is obtained. This approach is the basis of how modern computers work. It's no coincidence that numerical computation is closely linked to computer science. Without numerical algorithms, there would be no scientific simulations, 3D graphics, artificial intelligence, weather forecasting, or advanced engineering models.

VI. Conclusions

We could say that much of the technology we use today is based on numerical methods. Another key concept of numerical computation is approximation. Unlike pure mathematics, where absolute accuracy is sought, in numerical computation we accept the idea that every result has a margin of error. But this error is not a problem: it is a quantity that can be studied, controlled, and reduced. This is one of the most important ideas to understand: there is not only right and wrong, but also "how precise" is. From an educational perspective, numerical computation has enormous value. First, it helps connect mathematics to reality. Students no longer work solely with abstract symbols, but see concrete applications: data, simulations, models. Second, it develops computational thinking. Students learn to build algorithms,

think step-by-step, and verify results. Furthermore, it introduces a very important idea: mathematics is not always exact, but it is always useful. This profoundly changes the perception of the discipline. Another fundamental aspect is the role of digital tools. Today, numerical computation is closely linked to computers and programming languages. Without tools like Python, MATLAB, or other computing environments, many numerical methods would be too long or complex to apply manually. But it is important to emphasize that computer science and computers do not replace mathematical thinking: they amplify it. The role of the student or researcher remains central in developing the method and interpreting the results. A very significant example is the simulation of physical phenomena, such as the motion of planets, the diffusion of heat, or the behavior of fluids. In these cases, there are no simple formulas that describe everything exactly. Numerical calculation allows us to build approximate but extremely effective models.

I would also like to emphasize a cultural aspect. Numerical calculation represents a paradigm shift: it is no longer just about solving problems, but about simulating them, exploring them, and understanding them through successive approximations. It is a dynamic, experimental approach, very close to the scientific method.

In conclusion, we can say that numerical calculation is one of the most powerful tools of modern mathematics. Not because it renounces precision, but because it accepts the complexity of the real world and addresses it with tools that are simultaneously flexible, efficient, and rigorous. In a certain sense, numerical calculation teaches us an important lesson: it is not always possible to have a perfect answer, but it is always possible to get closer to the truth with ever greater precision. And this, in science as in life, makes a big difference. The implementation of the methods for approximate integral calculation, described in this work, also achieves an important educational result: it stimulates conscious, experiential, and intrapersonal development, open-mindedness, and reflection, contributing to the improvement of students' self-esteem (Quapp & Holschemacher, 2022 [2]). Reflecting on how solving numerical calculation problems can contribute to developing self-esteem, the following considerations can be made. When we think of mathematics, we often imagine numbers, formulas, and procedures. We rarely connect it with something emotional, like self-confidence. Yet, daily classroom experience tells us otherwise. Many students approach mathematics with uncertainty, sometimes fear. They have internalized the idea that they "don't have the talent," that they are not capable. This belief often becomes an even greater obstacle than the subject matter. So the question is: can we use mathematics not only to teach content, but also to build confidence? The answer is "yes". And numerical calculation is a particularly effective tool in this regard. Why? Because numerical calculation presents problems that, while complex, can be approached step by step. It doesn't immediately require a perfect solution, but rather allows students to gradually get closer to the result. This is a fundamental point. In numerical computation, an error isn't a failure: it's part of the process. Each approximation is an improvement over the previous one. Each attempt brings us closer to the solution. For a student, this is a profoundly different experience than a traditional exercise, where the result is often perceived as right or wrong, without nuance. Here, however, a richer idea comes into play: "I'm improving." And improving is something you can perceive, measure, and recognize. This fuels motivation and, above all, self-confidence.

Another important aspect is the structure of numerical computation problems. They are often algorithmic processes: you start with an initial idea, apply a rule, verify the result, and repeat the process. It's a clear, structured, and accessible path. This allows students not to feel lost. They know what to do, they know the next step. And each step completed is a small success. These small successes are crucial.

Self-esteem doesn't come from sudden, great results, but from a series of positive experiences repeated over time. Numerical calculation, with its iterative nature, offers exactly this type of experience.

There's another element I'd like to emphasize: the visibility of progress. When a student applies, for example, an iterative method, they can see how the result gets closer and closer to the correct value. This progress is concrete, observable. It's not abstract. And when progress is visible, it becomes credible. The student can say: "I'm understanding. I'm improving. I'm capable." This step is crucial. From an educational perspective, all of this requires a precise choice on the part of the teacher. It's not enough to simply propose numerical calculation exercises. It's necessary to create an environment in which mistakes are accepted, analyzed, and valued. An environment in which the process matters as much as the result. It's important to give space for reflection: "What happened? Why does the method work? How can I improve on my next attempt?" In this way, the student is no longer a performer, but becomes the protagonist of their own learning.

I would also like to emphasize the role of digital tools. The use of programming environments, such as Python and Java, allows students to experiment, quickly see results, modify parameters, and observe what changes. This makes learning more active and engaging. But, once again, the tool is only a means. What matters is the meaning of the experience. And meaning, in this case, is very powerful: I can tackle a complex problem, I can make mistakes, I can correct myself, and I can succeed. This is the heart of self-esteem. Not the belief that you're always good, but the confidence that you can improve. I would like to conclude with a reflection. Numerical calculation teaches us that precision is built through successive approximations. It's not achieved immediately, but rather achieved step by step. Perhaps the same is true for student growth. Self-esteem doesn't arise overnight either. It builds slowly, through experiences in which the student feels capable, competent, and progressing. So, perhaps teaching numerical calculation isn't just about teaching math. It's about offering students a concrete growth experience. An experience in which they learn not only to solve problems, but to believe they can do so (Quapp & Holschemacher, 2022 [2]).

VII. REFERENCES

[1] Riadh Habash, "An experiential engineering learning model for knowledge and state of flow creation", Athens Journal of Technology and Engineering, Volume 10, Issue 2, June 2023, Pages 89-106

[2] Ulrike Quapp, Klaus Holschemacher, "Management strategies for engineering faculties under consideration of current developments in the higher education sector", Athens Journal of Technology and Engineering, Volume 9, Issue 2, June 2022, Pages 147-158

[3] Rosalie Van Baest, “Stimulating conscious development mechanism for movement in engineering education”, Athens Journal of Technology and Engineering, Volume 6, Issue 3, Pages 195-210

[4] Walter Savitch, “Java: an introduction to problem solving and programming”, Pearson Education, 1997

[5] Massimo Bergamini, Graziella Barozzi, “Matematica.blu 2.0”, 2^ ed., volume 5, Zanichelli Editore, 2024

[6] Paolo Camagni, Riccardo Nikolassy, “InfoM@T per il liceo scientifico opzione scienze applicate”, Hoepli Editore, 2023

[7] https://www.treccani.it/enciclopedia/integrazione-numerica_%28Enciclopedia-della-Matematica%29/

[8] https://it.wikipedia.org/wiki/Integrazione_numerica

[9] Alfio Quarteroni, “Matematica numerica: esercizi, laboratori e progetti”, Springer-Verlag Italia, 2008

[10] Aimé Mbobi, “Principles and applications of numerical integration methods”, Lap Lambert Academic Publishing, 2012

[11] Allen B. Downey, “Think Python: how to think like a computer scientist”, O’Reilly Media, 2002-2012

[12] R. W. Hamming, “Numerical methods for scientists and engineers”, McGraw-Hill, 1973

[13] A. Quarteroni, R. Sacco, F. Saleri, “Matematica numerica”, Springer, 2010

Didactic introduction to Lambda-calculus

Mauro Pullin
Department of Neuroscience
University of Padova
Via VIII Febbraio, 2 – 35122 Padova, Italy
mauro.pullin@unipd.it

***Abstract* – This paper presents a didactic introduction to the λ-calculus aimed at first-year university students, with the goal of clarifying the concept of function and the relationship between mathematics and computer science. After highlighting some common misconceptions, the work proposes a three-lesson instructional path designed to develop basic knowledge of computability, the relationship between programs and functions, and the foundations of λ-calculus. The syntactic and semantic aspects of the formalism are examined, along with reduction strategies and parameter-passing mechanisms. The paper also relates λ-calculus to the main programming paradigms, emphasizing its role as a fundamental theoretical model. The adopted teaching approach aims to foster critical thinking, argumentation skills, and awareness of learning processes, showing how a formal theoretical framework can support the understanding of core concepts in computer science.**

***Keywords* – function, computation, computability, formal language, programming paradigms.**

I. Premise

This article begins by introducing the educational context in which the work is developed. It then examines the curriculum outline. The teaching methodology and key concepts are then examined. A comparison is made between different programming paradigms and lambda calculus. Finally, the conclusions are drawn.

II. Introduction

Lambda-calculus, or λ-calculus, is a very important formal system in computer science, defined in 1936 by mathematician Alonzo Church. It was developed to formally analyze functions and their computation (and thus their computer implementation). Functions are expressed using a formal language, which establishes the rules for forming a term; the latter uses a rewriting system, which defines how terms can be reduced and simplified.

The combination of simplicity and expressiveness has made lambda calculus a common tool in various scientific fields. In mathematics and computer science, particularly in computability theory, it was developed to study recursive functions as effectively computable. In computer science and programming language theory, it is the main prototype of functional programming languages and recursion.

This work — which is not a treatise on λ-calculus, but rather a didactic introduction to the topic — is intended for first-year university students.

We have therefore become aware of some gross misconceptions regarding computer science and mathematics.

Many first-year university students have unfortunately developed the belief that mathematics and computer science are separate disciplines, with completely different methodologies. In particular, they are convinced that in mathematics one seeks only exact solutions to problems, while in computer science they believe that one can and should also "approach solutions by trial and error", that is by successive approximations. The idea that it is necessary to preliminarily demonstrate the convergence of a method by trial and error, to avoid the risk of never getting close to the solution, but instead moving further and further away from it, rarely occurs to students who have graduated from high school.

First-year university students know that both computer science and mathematics refer to the concept of a function, but sometimes they think it's so different in the two disciplines that they wonder why the same word is used for such different concepts. Specifically, students believe that in computer science, a function refers to a subprogram to which values are passed and which returns results, while in mathematics, they think it refers to an "object" that must necessarily be associated with a graph representing its behavior.

Given the context in which we worked, it is certainly not possible to fully develop λ-calculus, knowledge of which is not part of the standard first-year university curriculum.

Therefore, we decided to limit ourselves to providing students with only some important preliminary notions.

III. Educational Context

A. Educational and training objectives

This topic was developed in the context of a first-year university program in a computer programming course within a science-related degree program (degrees in computer engineering, electronic engineering, or biomedical engineering). The knowledge we sought to develop includes correct terminology and the use of specific symbols, the concept of the effective computability of a function, the correlations between programs and functions, and the basic concepts of λ-calculus. We aim to develop various skills, including analyzing the correctness of reasoning and the ability to argue, compare, and formulate conjectures. Our educational objectives are to foster interest and critical thinking, the development of analytical, logical, reflective, and connecting skills, the ability to reflect and argue in support of one's choices, and the ability to reflect on the elements of any problem situation.

B. Route diagram

To develop the topic, we have proposed a sequence of three lessons: a review of the notion of function and basic concepts of λ-calculus; the syntax, semantics, and structure of λ-calculus; programming paradigms and λ-calculus.

The teaching methodology follows begins with an awareness of individual and collective knowledge in the given context, which is what happens in the first lesson of the described curriculum (first lesson).

In the second lesson, students gain an understanding of the nature of what they need to learn and the relationship between themselves and the topic being studied, outlining the subsequent steps of study and analysis. They also acquire an awareness of the skills to be implemented and the concepts to be revisited and mastered.

In the third and final lesson, students develop an awareness of the contextualization of skills and concepts in new and diverse situations.

At the end of the three lessons, students were asked to write a short summary essay and a PowerPoint presentation to be projected in class (since there were many students, we had to draw lots for the presentations to be projected; the others were available on the course's IT platform).

If conducted well, the entire series of lessons — though brief — can develop a good level of proficiency in each student, both in λ–calculus and — in general — in programming.

We have observed that, thanks to these lessons, the percentage of students who passed the programming exam on the first try has increased.

If well conducted, the entire series of lessons – despite being short – manages to develop in each student a reflective awareness of their own knowledge products and processes regarding the topic studied.

IV. Lesson Development

A. Teaching methodology and key concepts

The teaching methodology consisted of lectures, using PowerPoint presentations, and students were given ample opportunity to ask for clarification.

The following teaching methodology begins with a few premises: the notion of a function, the notion of symbolic calculation, and the notion of a computer program; these premises are covered in the first lesson. In the second lesson, students gain an understanding of the nature of what they are expected to learn and the relationship between themselves and the topic of study, outlining the subsequent phases of study and analysis. They also gain an awareness of the skills to be put into practice and the concepts to be reviewed and mastered. In the third and final lesson, students develop an awareness of the contextualization of skills and concepts in new and diverse situations.

In the first lesson, the focus was on the key concepts of λ-calculus. It is important to note that λ-calculus, a rewriting system formally defined by mathematician Alonzo Church in the 1930s, was designed to formally analyze function definitions.

It was developed to formally analyze function definitions and their applications, and is also an interesting tool for studying recursion. As a rewriting system, it provides a description of well-formed terms, which are sequences of symbols recognized by the system and capable of being rewritten by it.

λ-calculus, in fact, defines a set of rewriting rules that precisely determine how the terms themselves can be rewritten. In this way, the rewriting process becomes a true calculus. For this reason, λ-calculus will be referred to below as "the" calculus, unless there is a risk of ambiguity.

λ-calculus has its roots in the concept of functions. As is well known, in mathematics, functions are defined as associations between the elements of one set (domain) and the elements of another set (codomain).

For example, if we call a function that returns its square for a real number a square, we write:

$$square: R \rightarrow R$$

to indicate that a square maps elements of the set of real numbers to other elements of the same set.

If a function can associate a result with a given argument passed to it, we say that the function is defined for that value passed as an argument. If a function is defined on every element of the domain, it is called a total function; otherwise, we say it is a partial function.

Now we are invited to reflect on an important fact: the set-theoretic notation of a function, such as

$$f: R \rightarrow R$$

does not provide information on how to actually get from the received arguments to the associated result. In other words, it does not describe how to calculate the result, also called the output value of the function itself, from its arguments, also called input values. λ-calculus, like the theory of recursive functions and the Turing machine, is a formalism that allows us to define the "mechanical" side of functions, that is, the procedures that produce output values from given input values.

Just as physicists build mathematical models to understand, explain, and predict natural phenomena, so theoretical computer scientists build models based on mathematics and logic to study phenomena related to automatic information processing. The analogy with physics, however, ends there: the object of computer science, in fact, is not the universe of natural phenomena, but rather a plurality of "worlds" created by humans to represent and process information; these are therefore essentially artificial worlds. This characteristic of computer science influences the role played by theory, which is ambivalent: while the technological development of computer science sometimes precedes theory, which is therefore called upon to interpret and organize new findings a posteriori, the opposite is also historically true: theory preceded technological application, or rather, provided a framework of principles, a paradigm, inspiring a large number of applications both directly and from a methodological and technical standpoint; consider, for example, the Turing Machine.

As is well known, part of programming theory studies the relationships between programs and the processes they generate. Indeed, when we write a program, we do so because we are interested in the process it will generate during execution. It is therefore important to have both "prediction tools" which, given a program, tell us which process it will generate, and "synthesis tools" which guide us in writing a program that generates exactly the desired process.

λ-calculus can be considered an abstract programming language, extremely simplified syntactically, that can be used as a reference paradigm for all sequential programming languages. λ-Calculus is an example of how theory can precede technological application; in fact, it was born well before programming languages and well before the computer. It was proposed by Alonzo Church in the 1930s as a formalism to provide a logical foundation for mathematics, with the aim of precisely defining the notion of a computable function and, more generally, of an effective operation. Its encounter with computer science dates back to the late 1960s, when high-level programming languages, which are independent of computer architecture, became widely used.

The fundamental concept of λ-calculus is that of a function described by a symbolic manipulation rule.

For example, a polynomial with integer coefficients, such as

$$x^2 - x + 5$$

is an object of the ring Z[x] of polynomials over Z (the ring of integers) with indeterminate x; but it is also a way to express a function f from Z to Z, such that, for example,

$$f(2) = 2^2 - 2 + 5 = 7$$

Wanting to distinguish the function f from the polynomial that generates it (which in fact are mathematically different objects) we can write

$$f = \lambda x.\ x^2 - x + 5$$

and axiomatize the behavior of the new operator λ with the following formula:

$$(\lambda x.\ x^2 - x + 5)\ a \rightarrow a^2 - a + 5$$

B. *Programming paradigms and λ-calculus*

It is important to study the connections between programming paradigms and λ-calculus.

Programming languages have undergone a significant evolution, and new paradigms have been introduced, the most important and current of which is the object-oriented paradigm.

Each paradigm consists of an abstraction of the computer and its operating mode. The computer is seen as an automaton in the imperative paradigm, as a function evaluator in the functional paradigm, as an automatic prover in the logic programming paradigm.

The object-oriented paradigm views the computer as a simulator, in which the "objects" represent entities in a virtual world, each with specific "knowledge" and "skills," which exchange information and requests.

This paradigm clearly aims to reproduce a cooperative work situation. This paradigm is intrinsically concurrent and as such not completely modelable with λ-calculus, which instead is essentially sequential.

One of the most current topics of research in the field of programming theory is the definition of an abstract language on which to study object-oriented programming, and the proposals that have been made are mainly based on λ-calculus, to describe the sequential fragment of these languages, enriched with specific constructs to model concurrency.

At this point, it's appropriate to have students reflect on an important aspect of the current "world" of computer science: the fact that the radical shift in the intended use of computers, increasingly less as a tool for calculation and increasingly for communication, is evident to all: the success of the Internet is the most obvious example.

The theory here must address the study of distributed computations whose behavior cannot be traced back to an input-output function, but rather to a communication protocol that is often non-deterministic.

However, λ-calculus continues to be a point of reference and inspiration for the development of formalisms that should serve as prototypes and reference mathematical models in the world of concurrency and distributed computations.

V. Related Work

Between 2001 and 2010, λ-calculus was taught in the specialization courses of the SSIS of Veneto (a post-graduate school for teacher training). This experience highlighted that the study of λ-calculus facilitates the understanding of different programming paradigms. Lambda calculus is also successfully taught at the University of Florence (Italy), in the "Ulisse Dini" Mathematics Department.

VI. Conclusions

In conclusion, it's worth reflecting on an important aspect of current computer science: the radical shift in the intended use of computers, increasingly less as a tool for calculation and more as a means of communication (the success of the Internet is the most striking example). Theory here must address the study of distributed computations, whose behavior cannot be traced back to an input-output function, but rather to a communication protocol that is often nondeterministic. However, λ-calculus continues to be a point of reference and a source of inspiration for the development of formalisms that should serve as prototypes and reference mathematical models in the world of concurrency and distributed computing.

VII. REFERENCES

[1] https://it.wikipedia.org/wiki/Lambda_calcolo

[2] Alonzo Church, "A set of postulates for the foundation of logic", Annals of Mathematics, Second Series, Department of Mathematics, Princeton University1932.

[3] Henk Barendregt, "The lambda calculus: its syntax and semantics", North-Holland, 1984.

[4] Henk Barendregt, Wil Dekkers, Richard Statman, "Lambda calculus with types", Cambridge University Press, 2013.

[5] J. Roger Hindley & Jonathan P. Seldin, "Lambda-calculus and combinators: an introduction", Cambridge University Press, 2008

[6] Raúl Rojas, "A tutorial introduction to the lambda calculus", 2015

[7] Peter Selinger, "Lecture notes on the lambda calculus", Department of Mathematics and Statistics, Dalhousie University, 2008

[8] Achim Jung, "A short introduction to the lambda calculus", School of Computer Science, The University of Birmingham, 2004

[9] Hindley, J. R., & Seldin, “Lambda-calculus and combinators: an introduction”, J. P. Springer, 2008

[10] Simona Ronchi della Rocca, Ugo de’ Liguoro, “Teoria della programmazione e lambda-calcolo”, SSIS del Veneto, University “Ca’ Foscari”, Venezia, 2006

[11] Marco Maggesi, “Introduzione al lambda-calcolo”, Dipartimento di matematica Ulisse Dini, University of Firenze, 2008

High-Precision Temperature Sensor Architecture for Human-Mimicking Phantoms

Stanisław Nowak, Grzegorz Cichocki, Cyryl Prentki, Paweł Żelazek, Wojciech Leszczyński[1] and Krzysztof Cichoń[2]

[1]Science Club Spektrum, Poznan University of Technology, ul. J. Rychlewskiego 1, 61-131 Poznań
[2]Poznan University of Technology, ul. J. Rychlewskiego 1, 61-131 Poznań

***Abstract*—This paper presents a ten-channel, synchronized sensor architecture for monitoring temperature fluctuations in solid, semi-liquid, and liquid human-mimicking tissues. The architecture is a core instrumentation component of a system developed to validate the safety of 6G frequency bands by detecting localized heating in realistic biological models, offering a low-cost, scalable, and embeddable alternative to surface-restricted infrared thermography and to costlier, lower-channel-count fiber-optic or MR-based approaches. We discuss the technical challenges posed by various material forms and the stringent accuracy requirements inherent in such experiments. The proposed architecture multiplexes high-precision NTC thermistors, each integrated into an independent Wheatstone bridge, across ten synchronized channels digitized by 24-bit $\Sigma\Delta$ ADCs; a three-stage calibration procedure and a software-defined pulsed excitation scheme, which mitigates thermistor self-heating without additional circuitry, together yield calibrated temperature readings that are streamed in real time to a web-based operator interface. The system achieved single-millikelvin electrical-path accuracy in resistance-path calibration, a channel-to-channel consistency of $\pm 0.24\,^{\circ}\mathrm{C}$ across all ten channels over a 24-hour isothermal run, and stable dynamic tracking over a continuous 12-hour thermal test.**

***Index Terms*—NTC Thermistors, IoT System, Data Acquisition, Human-Mimicking Phantoms, Tissue Mimicking Materials**

I. Introduction

Over the past few decades, the biological impact of electromagnetic (EM) fields on the human body has emerged as a critical domain of interdisciplinary study, drawing substantial attention from medical researchers, biomedical engineers, and regulatory agencies [1]. Given the ubiquity of wireless communications, consumer electronics, and industrial equipment, extensive research has focused on quantifying the absorption and distribution of non-ionizing radiation within biological tissues. Because non-ionizing radiation cannot induce atomic ionization, it interacts with tissue primarily through dielectric heating, and exposure limits are therefore formulated to bound the resulting temperature rise. Below the $6\,\mathrm{GHz}$ transition frequency common to ICNIRP (2020) and IEEE C95.1-2019, the specific absorption rate (SAR) — the RF power absorbed per unit mass ($\mathrm{W\,kg^{-1}}$) — remains the basic measure of internal exposure [2], [3]. Above it, absorbed (or epithelial) power density replaces SAR as the internal dose metric, with absorbed energy density introduced for brief pulses; both limit sets were explicitly calibrated to avoid excessive tissue temperature rise [4]. This shift is directly relevant to the upper mid-band (FR3) spectrum considered for 6G, where absorption becomes increasingly superficial and the induced thermal gradient—rather than a volume-averaged SAR—is the quantity of practical interest. Since *in vivo* testing carries ethical and practical limits, dosimetric validation relies on anthropomorphic phantoms, ranging from physical tissue-equivalent liquids and gels to digital voxel models derived from MRI/CT scans. For physical phantoms, the value of the experiment is bounded by the instrumentation used to read the temperature field inside them: what is needed is not a single high-resolution point, but a spatially distributed, time-synchronized record of the gradient, acquired while the phantom is under illumination.

The precise evaluation of EM field exposure in phantoms requires reliable, continuous internal temperature monitoring. While the assessment procedures standardized above $6\,\mathrm{GHz}$ address power density [5], [6], the temperature rise those limits are calibrated against has no comparable measurement standard, leaving experimental thermal characterization dependent on instrument-specific solutions. Infrared thermography offers high-resolution data but is limited to the surface. EMI-immune fiber-optic probes and MR-based (PRF) thermometry are established alternatives, but typically instrument only one or two points per phantom or require an MRI scanner [7], [8]; fluorescent-thermoprobe and confocal-microscopy methods have recently pushed spatial resolution further, at the cost of offline, non-real-time acquisition [9]. Near-infrared solid-state sensing has likewise been demonstrated in tissue-simulating phantoms, but again at a single interrogation site [10]. Standard resistive sensors are cheap and embeddable but, in the reported literature, rarely offer synchronized multi-channel operation and, like the present work, are not intrinsically EMI-immune (Sec. V). Table I summarizes these trade-offs.

To address these limitations, this paper proposes a scalable, cost-effective NTC-thermistor sensor architecture for complex biological models, prioritizing embedded channel density and synchronization over the raw resolution of specialized optical techniques. Its main contributions are: (i) a synchronized, ten-channel multiplexed acquisition architecture, expandable to more channels; (ii) a three-stage calibration procedure with a software-defined pulsed excitation scheme that mitigates thermistor self-heating without extra hardware; (iii) a low-latency Redis/Grafana telemetry pipeline with a live web

TABLE I: Qualitative Comparison of Temperature-Sensing Modalities Used in EM-Exposure Phantom Studies

Modality	**Channels (typ.)**	**EMI**	**Reported accuracy**	**Real-time**
Infrared thermography	Surface, non-contact	Yes	70 mK NETD (FLIR ONE Pro [7])	At frame rate
Fiber-optic probes (GaAs/FBG)	1–2 (typ.)	Yes	± 0.3 K (Opsens TempSens [7])	Continuous
PRF MR thermometry	Volume, non-contact	Yes	Scanner-/sequence-dependent [8]	In-scanner, 41 s/map
Fluorescent probe + confocal	Point / 2D scan	Yes	$0.04\,^{\circ}$C, 2 µm (xy) [9]	No, offline
Near-infrared solid-state probe	Point (single)	Yes	$0.2\,^{\circ}$C–$0.3\,^{\circ}$C prediction error [10]	Per interrogation
This work (NTC bridge)	**10, expandable**	No*	Sub-mK; array $\pm 0.24\,^{\circ}$C (Sec. III)	**Live** web/SSE + Grafana

EMI: intrinsic immunity to electromagnetic interference. *Metallic elements; behavior under active RF exposure is future work (Sec. V). Accuracy figures are instrument specifications from the cited deployments, not modality-wide bounds.

panel and run-time reconfigurable per-channel calibration; and (iv) a metrological characterization comprising resistance-path calibration (single-millikelvin electrical-path accuracy), a 24-hour isothermal consistency run ($\pm 0.24\,^{\circ}$C across channels), and a 12-hour dynamic tracking experiment. The novelty claimed here lies not in the individual components, which are commercial off-the-shelf, but in their metrologically characterized integration: channel count scales with commodity converters and sensing elements rather than with dedicated interrogator hardware, which makes dense embedded instrumentation practical where the modalities of Table I constrain deployments to one or two interrogation points.

II. Self-Prepared Phantoms

The sensor architecture was developed to monitor thermal fluctuations within tissue-mimicking materials (TMMs) for the safety verification of the upper mid-band targeted for 6G [11], [12]. To be representative, TMMs must replicate the relative permittivity (ε_r) of human tissue, which varies across tissue types, with frequency, and with moisture content [13]–[16]; we adopted values from the IT'IS database [17], a comprehensive repository of dielectric properties. As shown in Table II, there is a nearly linear correlation between tissue water content and ε_r; higher moisture levels correspond to higher permittivity, Peyman et al. [18] present similar conclusions.

TABLE II: Permittivity and water content of selected tissues

Tissue	**Water (%)**	**Permittivity (ε_r) at Frequency (GHz)**				
		2	4	6	8	16
Skin	65.19	38.57	36.59	34.95	33.18	25.46
Fat	24.41	10.96	10.36	9.80	9.28	7.62
Brain	73.86	45.67	42.43	39.71	37.10	27.97
Bone	17.63	11.65	10.53	9.59	8.79	6.68

The TMMs use a common base of demineralized water and gelatin, with sunflower oil and glycol in varying proportions to replicate specific tissues. Following fabrication, the mixture was poured into 3D-printed molds encased in silicone, where it cooled and polymerized into a solid or semi-liquid state. The resulting TMMs were validated using dielectric resonators to confirm their ε_r values; this validation process is detailed in [19].

The physical properties of these phantoms—specifically their density and the requirement for multi-layered configurations—impose significant constraints on sensor integration. Because expected temperature fluctuations can range from subtle 0.01°C to a single degree Celsius, the measurement system must combine high precision with a large dynamic range and remain stable within a submerged, heterogeneous medium. These requirements necessitate a specialized multi-channel architecture capable of maintaining signal integrity across various depth levels.

III. Measurement System

A. Signal Conditioning Architecture

To achieve high precision within the phantom model [20], a multi-channel measurement system based on ultra-stable NTC thermistors was developed. Laboratory-grade CSP60BA252M sensors (Amphenol) were selected [21], featuring a nominal resistance (R_{25}) of $2500\,\Omega$ at $25\,^{\circ}$C and calibration accuracy up to $0.01\,^{\circ}$C. Because NTC thermistors exhibit an exponential sensitivity drop at higher temperatures [22], uncompensated integration into biological models introduces systematic nonlinear errors. Integrating each thermistor into a balanced Wheatstone bridge compensates this in hardware, transforming the exponential response into a quasi-linear S-shaped voltage transfer function [22] and optimizing dynamic range and resolution prior to digitization.

In practice, each of the 10 thermistors is integrated into an independent bridge with nominal arm resistances of $2.5\,\mathrm{k}\Omega$. High-precision resistors were meticulously matched to constrain maximum arm resistance mismatch to under $0.2\,\Omega$, verified using a digital multimeter with $0.1\,\Omega$ resolution. The resulting differential voltage, V_{diff}, is continuously digitized via a 24-bit $\Sigma\Delta$ ADC.

B. Signal Processing and Data Acquisition Chain

The overall signal processing and data acquisition architecture is illustrated in Fig. 1.

The 10 bridges are digitized by five 24-bit NAU7802 ADCs : each converter's dual differential inputs (VIN1, VIN2) are routed through its internal multiplexer, so five units serve all ten nodes. Their high effective number of bits (ENOB) resolves the minute resistance changes produced by subtle temperature shifts in the phantom.

To interface the ADCs with a single master controller, a TCA9548A 8-channel I^2C multiplexer resolves the addressing conflict between identical converters through software-driven channel selection . Control and logging routines execute on

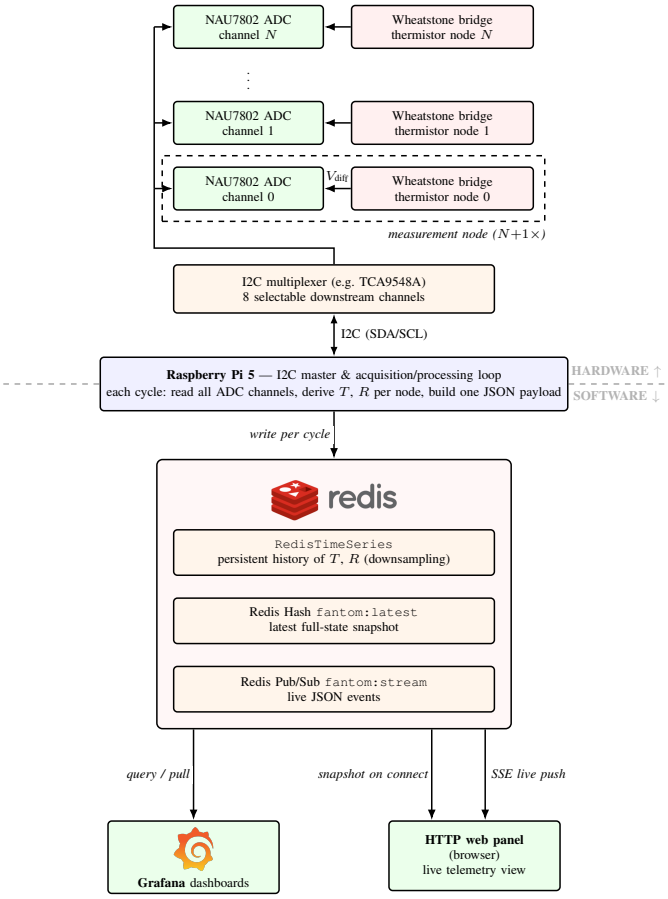


Fig. 1: End-to-end signal-acquisition and dissemination chain, read top to bottom. The dashed rule separates the hardware front-end (Wheatstone bridges, ADCs, I^2C multiplexer) from the software layer (three Redis structures feeding Grafana and the HTTP web panel), with the Raspberry Pi 5 as the hinge.

a Raspberry Pi 5, which runs a sequential readout across the multiplexer channels and dual ADC inputs, ensuring a continuous and synchronized record of the spatial temperature profile.

C. ADC Calibration and Metrological Optimization

Given the accuracy required here, calibration of the measurement path followed the compensation procedure specified for the 24-bit NAU7802 ADC. Combined with the low-drift MBB0207 precision resistors, this minimizes the overall thermal and temporal drift of the measurement system, ensuring high fidelity of the experimental data.

The calibration was performed in three main stages.

1) Internal Calibration: Conducted at every system startup to eliminate the converter's internal PGA offset and gain errors. The device disconnects the external Wheatstone bridge and shorts its differential inputs internally, verifying IC operation prior to active measurement.

2) System Offset Calibration: Compensates for asymmetries in the external analog circuit (bridge imbalances, thermistor variations, wire parasitic resistance) as well as internal amplifier errors. Determined from the resting bridge unbalance, this value is stored in the NAU7802 calibration register.

3) Resistance-Path Calibration: Performed as the final stage, since any repetition of the ADC calibration alters the raw output codes and would invalidate it. In each channel, the thermistor was temporarily disconnected and replaced by a precision reference resistor of $\approx 2500\,\Omega$ with an individually known value ($0.1\,\Omega$ resolution). From the measured-versus-reference pairs, a per-channel scale correction $R' = aR$ is derived and applied to every subsequent resistance reading (under the condition that the absolute offset has been nullified during Stage 2), and all calibration measurements are appended to a persistent log.

D. Measurement Execution and Signal Transformation

The core of the software operates as a synchronous, time-bound continuous loop, executing at a defined interval (T_{period}). During each cycle, the system iterates over the active measurement nodes. For each node, the I^2C multiplexer routes the bus to the appropriate NAU7802 ADC. A mandatory hardware settling time (t_{settle}) is enforced to eliminate transient electrical noise caused by bus switching before the software polls the ADC at its base nominal rate of 10 SPS. The system reads raw digital values and applies a software-level moving average over a specified window to mitigate transient noise while preventing blocking delays in the event of hardware unresponsiveness.

Following raw data acquisition, the software pipeline applies a two-stage mathematical transformation to derive physical values. First, the raw 24-bit integer is converted into a physical resistance value in ohms using the fixed reference resistor value (R_{fixed}) and the pre-configured ADC gain derived from a standard ratiometric bridge formula. Next, this calculated resistance—after applying the per-channel scale correction a determined during resistance-path calibration—is processed through a non-linear transfer function. The software utilizes the Steinhart–Hart equation based on calibrated thermistor coefficients to output the final temperature in degrees Celsius.

E. Database Dissemination and Telemetry

Each loop cycle ends with a low-latency distribution of the data through Redis. The computed temperature and resistance are injected into `RedisTimeSeries`, the persistent historical backend optimized for downsampling and Grafana visualization, while the instantaneous readings of all channels are compiled into a single JSON payload. That payload is written to a Redis Hash (`fantom:latest`) as an immediate state snapshot for newly connected clients and is simultaneously published on a Pub/Sub channel (`fantom:stream`), from which decoupled web panels receive Server-Sent Events (SSE) with near-zero latency. The cycle concludes by measuring its own execution time and sleeping for the remaining delta, maintaining a strictly periodic sampling frequency.

F. Operator Interface, Configuration, and Deployment

Beyond passively rendering live telemetry (Fig. 2), the HTTP web panel doubles as the primary configuration and



Fig. 2: Operator-facing FANTOM 6G web panel, refreshed live over Server-Sent Events. Each card reports one channel (Sb–CHn): temperature, bridge resistance, multiplexer/ADC routing, and a sparkline of the recent trend.

calibration interface for the instrument. The heart of this configuration is the per-channel calibration table binding each logical channel to its physical routing—the TCA9548A downstream port selecting the corresponding NAU7802—and to a thermistor set identified by serial number, with its Steinhart–Hart coefficients (b_0–b_3), the nominal bridge reference resistance R_{W}, and the length L and per-meter resistance ρ of the cable used for lead compensation. Because this mapping lives in editable configuration rather than in firmware, channels can be re-pinned, recalibrated, or swapped for a freshly characterized thermistor set at run time—a practical advantage when phantoms are re-instrumented between experiments.

G. Calibration Results and Metrological Evaluation

The resistance path of all 10 channels was verified against the precision reference resistors, as summarized in Table III. Before applying the correction, all channels agreed with the reference values to within $-0.033\,\%$ to $+0.026\,\%$ ($\leq 0.83\,\Omega$ at $2500\,\Omega$). Given the thermistor sensitivity of approximately $-4.3\,\%/\mathrm{K}$ at room temperature, this corresponds to a temperature error not exceeding $\pm 10\,\mathrm{mK}$: the electrical path (bridges, ADCs, cabling) is thus accurate at the level of single millikelvins, and the derived scale corrections primarily provide traceable documentation of this accuracy.

To assess the relative consistency of the array, all 10 thermistors were placed for 24 hours in a common isothermal enclosure together with an independent digital reference sensor, SHT45 (Sensirion, typical accuracy $\pm 0.1\,^{\circ}\mathrm{C}$), and 5805 full measurement cycles were recorded. The observed channel-to-channel spread of $\pm 0.24\,^{\circ}\mathrm{C}$ (ensemble standard deviation $0.18\,^{\circ}\mathrm{C}$) is unaffected by the resistance correction; consistently with the results above, it originates from the uncertainty of the individual Steinhart–Hart coefficients rather than from the electrical path. These deviations are systematic and stable in time and can therefore be removed in software by per-channel offsets referenced to the ensemble mean.

TABLE III: Resistance-Path Calibration Results per Bridge

Bridge	R_{ref} [Ω]	R_{meas} [Ω]	a	δT [mK]
0	2498.3	2498.61	0.999876	−2.9
1	2498.2	2497.64	1.000224	+5.2
2	2498.1	2497.75	1.000140	+3.3
3	2499.2	2498.63	1.000228	+5.3
4	2499.6	2499.40	1.000080	+1.9
5	2499.5	2499.20	1.000120	+2.8
6	2498.1	2497.27	1.000332	+7.7
7	2499.6	2498.83	1.000308	+7.2
8	2500.3	2500.94	0.999744	−6.0
9	2498.2	2498.38	0.999928	−1.7

The ensemble mean exceeded the SHT45 reading by $\approx 1.7\,^{\circ}\mathrm{C}$, attributed to thermistor self-heating under continuous $3.0\,\mathrm{V}$ bridge excitation ($\approx 0.9\,\mathrm{mW}$ per thermistor in still air). Since the excitation is supplied by each ADC's internal LDO, it was switched to a pulsed mode purely in software: the analog section of each converter is powered only during its read window ($\approx 17\,\%$ duty cycle at the $20\,\mathrm{s}$ measurement period). This reduced the offset to $0.64\,^{\circ}\mathrm{C}$ while leaving the channel-to-channel spread unchanged, confirming its thermal rather than electrical origin. Since the duration of the read window is fixed, the duty cycle — and with it the mean dissipated power — scales inversely with the measurement period, which is user-configurable in the range of $15\,\mathrm{s}$ to $3600\,\mathrm{s}$. This dependence was verified experimentally: extending the period to $60\,\mathrm{s}$ ($\approx 6\,\%$ duty cycle) further reduced the offset from $0.64\,^{\circ}\mathrm{C}$ to $0.51\,^{\circ}\mathrm{C}$, again with an unchanged channel-to-channel spread. In air, lowering the sampling rate is therefore a simple software-only mitigation of self-heating, limited only by the in-window heating component, which is independent of the period. In the phantom's tissue-mimicking medium — a gel based on distilled water and gelatin, with admixtures

of sunflower oil, sugar, NaCl, xanthan, and propylene glycol depending on the mimicked tissue — the dissipation constant is 1–2 orders of magnitude higher than in air, rendering the residual self-heating negligible; moreover, a common additive offset does not affect the measured temperature gradients, which are the primary quantity of interest in this application.

IV. Results

A. Initial Measurement Setup

A baseline experiment was conducted in a controlled homogeneous liquid medium: a large sealable plastic container filled with water, with holes drilled to mount the sensors (Fig. 3). The ten NTC thermistors were submerged and connected to the acquisition system, while an independent reference thermometer (Sensirion SHT45) placed nearby measured the ambient air temperature.

A controlled thermal drift was then induced: the laboratory HVAC (Heating, Ventilation, and Air Conditioning) system was cycled on and off with the set point lowered to 19°C, producing a periodic external forcing function and a slow cooling of the liquid volume. This transient regime exercises the analog front-end's resolution, its systematic noise, and the effectiveness of the calibration routines.

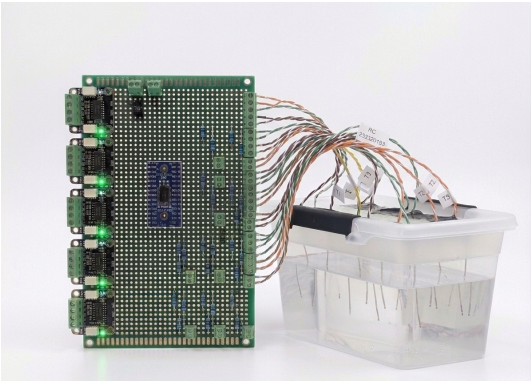


Fig. 3: Baseline setup: the acquisition board interfaced with ten NTC thermistors submerged in water.

B. Experimental Results

The architecture was evaluated over a continuous 12-hour observation window under this periodic forcing (Fig. 4).

Despite the extremely slow rate of temperature change inherent to this thermal mass, the response curves of the NTC sensors remain smooth. The absence of digital quantization steps or random signal jitter under these transient conditions empirically validates the high effective resolution of the 24-bit $\Sigma\Delta$ ADCs and the hardware-level linearization of the Wheatstone bridge topology.

The statistical evaluation of the measurement window, summarized in Table IV, quantifies this contrast. While the ambient reference sensor recorded a highly dynamic absolute temperature range of 4.29°C ($\sigma = 1.10$°C), the temperature of the water volume was heavily damped, spanning an average range of approximately 1.18°C ($\sigma \approx 0.31$°C).

TABLE IV: Statistical Summary of the Dynamic Thermal Response

Sensor	Min [°C]	Max [°C]	Range [°C]	Mean [°C]	σ [°C]
NTC 1	17.4095	18.5842	1.1746	18.0100	0.30659
NTC 2	17.4876	18.6632	1.1756	18.0862	0.30754
NTC 3	17.3671	18.5481	1.1810	17.9652	0.30665
NTC 4	17.4305	18.5978	1.1673	18.0257	0.30627
NTC 5	17.4204	18.5802	1.1599	18.0117	0.30569
NTC 6	17.4896	18.6594	1.1698	18.0869	0.30628
NTC 7	17.3612	18.5297	1.1685	17.9585	0.30196
NTC 8	17.4471	18.6306	1.1835	18.0497	0.30828
NTC 9	17.3814	18.5999	1.2185	17.9975	0.30591
NTC 10	17.5462	18.7164	1.1702	18.1445	0.30328
SHT45	15.4910	19.7848	4.2939	17.7086	1.09959

The SHT45 records the ambient air forcing function, not an accuracy reference for the submerged array: different medium and thermal time constant.

More importantly, the data reveals excellent intra-array consistency across the custom sensor network. The 10 distinct channels—each routed through its respective multiplexer path and independently calibrated—tracked the same forcing with closely matched statistics (σ within 2.1% across channels) and a constant inter-channel offset of $0.19\,^{\circ}\mathrm{C}$ peak to peak. The sequential readout introduces a temporal skew bounded by the cycle period, more than two orders of magnitude shorter than the thermal time constant of the medium, so the array samples an effectively simultaneous thermal state. This uniformity indicates that the sensor array can produce homogeneous spatial thermal maps when embedded in human-mimicking tissues for RF dosimetric analysis.

C. Comparison with Reported Sensing Modalities

Table I sets these figures against the alternatives, recalling that the $\pm 0.24\,^{\circ}\mathrm{C}$ inter-channel spread originates in the thermistor coefficients rather than in the acquisition chain. Against infrared thermography ($\approx 70\,\mathrm{mK}$, camera-dependent [7]) the proposed system trades non-contact operation for depth-resolved sampling inside the medium, and it does not reach the $0.04\,^{\circ}\mathrm{C}$ at $2\,\mu\mathrm{m}$ resolution of fluorescent-thermoprobe confocal microscopy [9], which is however inherently offline and instrument-bound. The decisive practical difference is channel density: the reported deployments of EMI-immune probes instrument one or two sites per phantom [7], [8], an order of magnitude fewer than the ten concurrent embedded points streamed in real time here.

V. Conclusion

The 12-hour experiment established the metrological stability and dynamic tracking of the architecture: single-millikelvin electrical-path accuracy, no perceptible quantization or jitter, and no channel-dependent skew across the ten multiplexed channels.

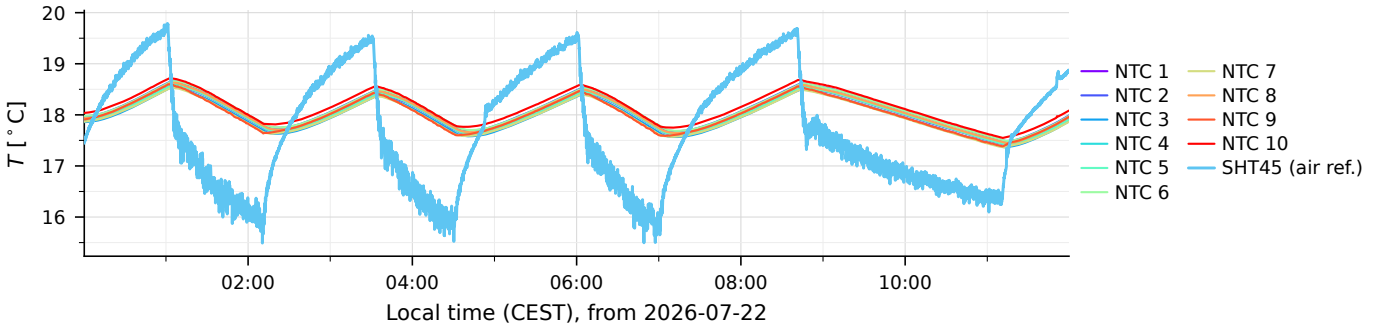


Fig. 4: Dynamic thermal response over the 12-hour window. The Sensirion SHT45 reference (blue) tracks the high-variance, periodic HVAC ambient air cycles, while the 10-channel NTC array resolves the damped, phase-shifted water volume response.

The validation reported here is metrological: it establishes the resolution, stability, and channel-to-channel consistency of the acquisition chain, but does not yet include a temperature measurement taken under active 6G-band RF exposure. Two limitations follow. First, unlike the intrinsically EMI-immune modalities of Table I, the sensing elements and cabling used here are metallic, and their behavior beside an active radiating antenna has not yet been characterized; this will be addressed by shielding evaluation and exposure-synchronized acquisition. Second, the absolute accuracy is currently bounded by the per-channel Steinhart–Hart coefficients rather than by the electronics, and can be improved by a bath calibration against a traceable reference.

Ultimately, this work delivers a robust, metrologically characterized sensor network for tissue-mimicking materials and establishes the instrumentation basis for the forthcoming RF-exposure dosimetric campaign.

Acknowledgment

The members of Science Club Spektrum were supported by the Polish Minister of Science and Higher Education under the program “Student Research Groups Create Innovations”.

References

[1] SCENIHR (Scientific Committee on Emerging and Newly Identified Health Risks), “Potential health effects of exposure to electromagnetic fields (EMF).” European Commission, https://health.ec.europa.eu/, 2015. Accessed: 10 July 2026.

[2] International Commission on Non-Ionizing Radiation Protection (ICNIRP), “Guidelines for limiting exposure to electromagnetic fields (100 kHz to 300 GHz),” *Health Physics*, vol. 118, pp. 483–524, May 2020.

[3] IEEE, “IEEE standard for safety levels with respect to human exposure to electric, magnetic, and electromagnetic fields, 0 Hz to 300 GHz,” 2019.

[4] A. Hirata, S. Kodera, K. Sasaki, J. Gomez-Tames, I. Laakso, A. Wood, S. Watanabe, and K. R. Foster, “Human exposure to radiofrequency energy above 6 GHz: review of computational dosimetry studies,” *Physics in Medicine & Biology*, vol. 66, no. 8, p. 08TR01, 2021.

[5] IEC/IEEE, “Assessment of power density of human exposure to radio frequency fields from wireless devices in close proximity to the head and body (frequency range of 6 GHz to 300 GHz)—part 1: Measurement procedure.” IEC/IEEE Std 63195-1:2022, 2022.

[6] IEC/IEEE, “Assessment of power density of human exposure to radio frequency fields from wireless devices in close proximity to the head and body (frequency range of 6 GHz to 300 GHz)—part 2: Computational procedure.” IEC/IEEE Std 63195-2:2022, 2022.

[7] J. Blackwell, G. Oluniran, B. Tuohy, M. Destrade, M. J. Kraśny, and N. Colgan, “Experimental assessment of clinical MRI-induced global SAR distributions in head phantoms,” 2020. arXiv:2009.04753.

[8] H. Jeong, M. C. Restivo, P. Jezzard, and A. T. Hess, “Assessment of radio-frequency heating of a parallel transmit coil in a phantom using multi-echo proton resonance frequency shift thermometry,” *Magnetic Resonance Imaging*, vol. 77, pp. 57–68, 2021.

[9] S. Yamazaki, M. Mizuno, and T. Nagaoka, “Development of a cornea-equivalent phantom in the terahertz frequency region for 3D temperature rise assessment,” *Scientific Reports*, vol. 15, 2025.

[10] A. Kauffman, J. Q. Nguyen, S. Parthasarathy, and M. A. Arnold, “Noninvasive temperature measurements in tissue-simulating phantoms using a solid-state near-infrared sensor,” *Sensors*, vol. 24, no. 12, p. 3985, 2024.

[11] 3GPP, “7–24 GHz frequency range.” 3GPP TR 38.820, V16.1.0, 2021.

[12] S. Kang, M. Mezzavilla, S. Rangan, A. Madanayake, S. B. Venkatakrishnan, and G. Hellbourg, “Cellular wireless networks in the upper mid-band,” *IEEE Open Journal of the Communications Society*, vol. 5, pp. 2058–2075, 2024.

[13] S. Gabriel, R. Lau, and C. Gabriel, “The dielectric properties of biological tissues: II. Measurements in the frequency range 10 Hz to 20 GHz,” *Physics in Medicine and Biology*, vol. 41, no. 11, pp. 2251–2269, 1996.

[14] K. R. Foster and H. P. Schwan, “Dielectric properties of tissues and biological materials: a critical review.,” *Critical Reviews in Biomedical Engineering*, vol. 17, no. 1, pp. 25–104, 1989.

[15] K. Sasaki, K. Wake, and S. Watanabe, “Measurement of the dielectric properties of the epidermis and dermis at frequencies from 0.5 GHz to 110 GHz,” *Physics in Medicine & Biology*, vol. 59, no. 16, pp. 4739–4747, 2014.

[16] S. Wood, N. Krishnamurthy, T. Santini, S. B. Raval, N. Farhat, J. A. Holmes, and T. S. Ibrahim, “Design and fabrication of a realistic anthropomorphic heterogeneous head phantom for MR purposes,” *PLOS ONE*, vol. 12, no. 8, p. e0183168, 2017.

[17] C. Baumgartner, P. A. Hasgall, F. Di Gennaro, E. Neufeld, B. Lloyd, M. C. Gosselin, D. Payne, A. Klingenböck, and N. Kuster, “IT’IS Database for thermal and electromagnetic parameters of biological tissues.” Version 5.0, https://doi.org/10.13099/VIP21000-05-0, Aug. 2025. Accessed via itis.swiss/database.

[18] A. Peyman, A. A. Rezazadeh, and C. Gabriel, “Changes in the dielectric properties of rat tissue as a function of age at microwave frequencies,” *Physics in Medicine & Biology*, vol. 46, no. 6, pp. 1617–1629, 2001.

[19] C. Prentki, P. Czekała, Ł. Nowicki, and K. Cichoń, “Dedicated dielectric resonator-based measurements for liquid samples for phantoms,” in *2026 26th International Microwave and Radar Conference (MIKON)*, pp. 261–263, 2026.

[20] M. Burfeindt, T. Colgan, R. Mays, J. Shea, N. Behdad, B. Van Veen, and S. Hagness, “MRI-Derived 3-D-Printed breast phantom for microwave breast imaging validation,” *Antennas and Wireless Propagation Letters, IEEE*, vol. 11, pp. 1610–1613, 2012.

[21] Amphenol Advanced Sensors, *Type CSP: Laboratory NTC Temperature Standards*. Datasheet AAS-920-615A.

[22] J. Jovanović and D. Denić, “NTC thermistor nonlinearity compensation using Wheatstone bridge and novel dual-stage single-flash piecewise-linear ADC,” *Metrology and Measurement Systems*, vol. 28, no. 3, pp. 523–537, 2021.

Software Defined Radio for Attacks in Wireless Networks

Nikola Danilović, Krešimir Grgić
Faculty of Electrical Engineering, Computer Science and Information Technology Osijek
Josip Juraj Strossmayer University of Osijek
Osijek, Croatia
nikola.danilovic@ferit.hr, kresimir.grgic@ferit.hr

***Abstract*—Wireless networks are a widespread in current technological landscape. Because of that it is worth researching and exploring different aspects of cybersecurity pertaining to it. There is a large number of different possible attacks on wireless networks such as signal spoofing, jamming, eavesdropping and replay attacks just to name a few. Because of that it is important to make an overview of those attacks, especially in the context of software defined radio, which proofed to be a valuable tool for examination of this scientific field.**

***Index Terms*—software-defined radio, cybersecurity, cyberattack, wireless networks, internet of things**

I. Introduction

In the current state of the world and technology, cybersecurity plays a crucial role in management and safety of any infrastructure that is connected to the internet. That dependency is even more exaggerated when observing all the wireless systems that are being used. Wireless systems, with the accelerated adoption of IoT devices, have increased connectivity but also, inevitably, increased attack surface of those same systems.

To see how important and how vulnerable to attacks wireless systems are currently, example incident can be provided. On April 7, 2017, 156 emergency sirens were triggered due to the cyberattack on emergency system infrastructure. In later case study of that cyberattack, it was concluded that the attacker has taken advantage of vulnerability inside the wireless infrastructure of the emergency system [1].

This example, and many others, can be used as a study case of how low level vulnerabilities in some wireless system can be exploited in some malevolent purposes. It must be acknowledged that high level software cyberattacks are not the only kinds of threats that can be utilized when malicious third party is trying to attack wireless system.

To combat these kinds of low level threats, study them or even simulate them in controlled and safe environment, software-defined radio technology (SDR) can be used which will be described in second section. SDR is flexible in a sense that platforms and devices that utilize it can be used as transmitter or a receiver. In that sense it can be used both as attack platform and as defensive security platform.

Purpose of this survey is to make a review of SDR technologies in wireless systems with an emphasis on different types of security risks that manifest in lower layers of network architecture, below network layer. Brief overview will be done on those threats in third section. Proposed methodology for research and comparison of relevant papers is to dissect each paper and its type and find which wireless technology has been studied and which SDR platform has been used. Furthermore, collected data and evaluation metrics will be compared in fourth section, as well as limitation of study in the context of SDR technology. In fifth section conclusion will be made based on previous sections.

II. Software defined radio background

Software defined radio technology as it is currently in use can be defined as a device that can behave as radio transmitter, but also as receiver if needed. It is a technology that enables RF operating parameters like frequency range and modulation type, just to name a few, to be manipulated by software.

Essential part of the SDR is the digital radio which consists of three blocks that enables needed functionality. Those are: RF, IF and baseband section. Encoded radio frequency is being received through antenna which is connected to RF front-end part of the SDR digital radio. That part is used to transmit or receive RF signals and change it to some constant Intermediate Frequency (IF).

As it can be seen in figure 1, Analog to Digital Converter (ADC) and Digital to Analog Converter (DAC) are parts of the architecture which enable conversion from analog to digital signal and from digital to analog signal respectively. Furthermore, modulation and demodulation blocks which are called Digital Up Conversion (DUC) and Digital Down Conversion (DDC) serve as modulation and demodulation blocks for transmitting and receiving path respectively.

Additionally there is the baseband section where equalization, correlation, frequency hopping, coding and decoding happens [2]. These qualities and capabilities of SDR technology and its architecture can enable many defensive and offensive tasks, while also serve as a great tool for spectrum monitoring

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, project: "Threat Detection System for Wireless Network Environments (SOS-BMO)". Project code: 581-UNIOS-33. This research was funded by the European Union - NextGenerationEU. However, the views and opinions expressed are solely those of the authors and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

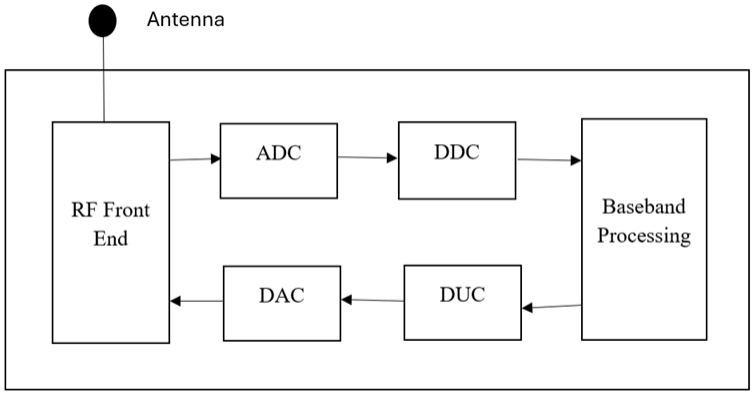


Fig. 1. Basic example of a SDR architecture scheme.

and analysis. In different words, it is a tool that can be used for simulating attacks, making a defensive analysis and observing frequency spectrum and reporting on its usage [1].

A. SDR hardware

There are many SDR hardware platforms. They vary in ADC and DAC specifications, ability to transmit or just receive signal, frequency range and maximal RF bandwidth [1].

Through the development of aforementioned SDR hardware platforms, there can be found a clear improvement of their specifications and properties. That can be seen in USRP-1, designed in 2004, with DAC sample rate of 64 [MS/s], and subsequently in CRIMSON Cyan in which the sample rate can go up to 325 [GS/s] [1]. Improvements can also be seen in frequency range and in computational capacity, where the latter has been achieved by using Digital Signal Processor (DSP) and FPGA devices.

There are also more budget friendly options when it comes to SDR hardware platforms. Some of them include FUNCube, RTL, HackRF and Pluto, just to name a few. General Specifications of these, and some more advanced SDR platforms can be seen in Table 1. There can be seen their frequency ranges, maximal RF bandwidth and GNU Radio compatibility [1].

While some can be used with Matlab, almost all of the hardware solutions can be used with GNU Radio open source development toolkit [3] which is used to implement SDR technology [1]. The compatibility can be seen in Table 1.

These different hardware platforms can have a great significance in the context of cybersecurity. In that context it is important to take into account vulnerabilities of certain SDR platform, their attack potential and the diagnostics and analysis possibilities.

III. SDR Wireless Threats

There are myriad of cybersecurity related threats that are possible and that involve SDR technology analysis. Some of them involve technologies that were being analysed, such as Global System for Mobile communication (GSM), Long Term Evolution (LTE), Radio Frequency Identification (RFID), just to name a few. Threats that were being analysed were spoofing, sidechannel, eavesdropping, jamming and replay attacks.

A. Replay attacks

Replay attack is capture of access control information or authentication with the goal of reproducing and retransmitting it to achieve unauthorized access and effect. One of the possible examples of such attack can be found in automotive industry. Remote Keyless Entry (RKE) is used as a mechanism for remote locking or unlocking of the vehicles, among other things. Some of the techniques to avoid or detect it, such as short-term memory neural networks, show 80% of accuracy of detection of that kind of attack [4]. Furthermore, SDR has also been used in analysis of replay attack in RFID systems using USRP N210/SBX hardware [1].

B. IoT wireless vulnerabilities

There has been a plethora of examples of attack analysis using SDR technology in wireless IoT systems. One of those examples is man in the middle attack using Bluetooth 2.1 technology with USRP 2 SDR hardware.

In general, bluetooth is vulnerable in respect to various critical security risks [5]. There are some problems that can be caused by denial-of-service (DoS) attacks, where large amounts of traffic can overwhelm a device. Also eavesdropping and unauthorized access can be achieved with Bluetooth Low Energy (BLE) forced connection [6].

Furthermore Long Range (LoRa) technology, as one of main examples of Low Power Wide Area Network (LPWAN) technologies, has also been analysed for cybersecurity reasons. Threats have been categorized and studied like key reuse, eavesdropping, jamming and replay attacks [7]. All those are cybersecurity aspects of technology that can be used for malicious purposes, but it is also a chance for improvement of said technology .

In addition to LoRa and bluetooth, widespread IoT based technology such as WiFi can also have vulnerabilities among which there is unauthorized entry, malware and inadequate encryption, just to name a few [8].

TABLE I
SDR PLATFORMS AND THEIR PROPERTIES

SDR Platform	Frequency Range [MHz]	Maximum RF Bandwidth [MHz]	GNU Radio Support
FUNcube Dongle	64–1700	0.08	Yes
RTL	25–1750	3.2	Yes
HackRF One	1–6000	20	Yes
Pluto	325–3800	20	Yes
USRP-1	DC–6000	40	Yes
USRP-N210	DC–6000	40/80/120/160	Yes
USRP N320	3–6000	200	Yes
CRIMSON Cyan	0.5–18000	1000 (upgradable to 3 GHz)	Yes

C. Signal Spoofing and Jamming

Signal spoofing in its essence is an attack with which attacker tries to imitate some parts of the communication by taking advantage of loopholes, problems or inconsistencies in communication protocols. One of the examples of signal spoofing is GPS spoofing, which takes advantage of relatively weak GPS signal that is already prone to in-band interference [9]. RF interference can be used for jamming attack too. Jamming attack, while it has a few variations, is mainly an attack that uses continuous or sporadic and random injection of signal to disrupt or block wireless communication [10].

D. Fake access point attack

Fake access point attack is ever more prevalent danger in context of security. In Wi-Fi networks cyber attacks are much easier to counduct. During the attack malicious party can sniff, eavesdrop or interupt the frames [11].

IV. COMPARISON

For comparison four papers were taken into consideration because they neatly fit into proposed research method and because they fulfill criteria pertaining to said method. In table 2 it can be seen that there is a great variety in given examples spanning from wireless technology to evaluation metrics and limitations. Some studies evaluate larger amount of metrics like the one that studies replay attack [4], while others focus more on few carefully selected metrics like in the case of eavesdropping and GPS spoofing study. It has to be noted that there are multitudes of papers that don't thoroughly investigate the threat in the context of this overview methodology. For example, DoS and jamming attack paper [14] from table 2 does not go into detail regarding collected data, evaluation metrics and limitations of SDR technology. Instead it focuses more on simple distinction between those attacks that were successful and those that failed in research experiment.

V. CONCLUSION

Cybersecurity of wireless networks is dynamic and ever changing area of research. In this survey of possible physical layer network attacks in context of SDR technology it can be concluded that the papers taken into account for wireless network attacks span from replay attacks and eavesdropping to GPS spoofing. There is plethora of wireless technology types that these kinds of studies can be made upon, from RFID systems to car technology and unmanned aircraft systems. They vary in scope and depth of research, because they have different evaluation metrics and limitations, but they all use SDR technology in research which proves to be an efficient but not limitless tool for wireless network attack research. It also needs to be noted that future research should be in direction that this paper is lacking, and that is more thorough research papers on denial of service and jamming attacks for example, that include precise evaluation metrics, collected data and limitations.

TABLE II
COMPARISON OF SDR-BASED WIRELESS NETWORK ATTACK STUDIES

Attack Type	Wireless Technology	SDR Platform	Collected Data	Evaluation Metric	Limitations
Replay [4]	RKE (Remote Keyless Entry)	HackRF One, RTL	55 samples of the original signal	Sensitivity, specificity, PPV, NPV, classification error, accuracy, F1 score, and AUC	Spectrogram of phase, magnitude, and other radio-frequency parameters
Eavesdropping [12]	FHS RFID system	USRP1 and USRP-N210s	100,000 bits for each channel-hopping duration	Amplitude and BER	Considers only linear combining of the received signal and assumes that the attacker is close to the RFID system
GPS spoofing [13]	GPS on UAS (Unmanned Aircraft System)	HackRF	Relative positioning derivatives	MAVLink parameters (yaw, pitch, roll)	Not defined in the paper
DoS and jamming [14]	LoRaWAN	HackRF One	Success or failure of an attack	Success or failure of an attack	Not defined in the paper

REFERENCES

[1] J. de Jesús Rugeles Uribe, E. P. Guillen, and L. S. Cardoso, "A technical review of wireless security for the internet of things: Software defined radio perspective," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 4122–4134, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157821000896

[2] R. Krishnan, R. G. Babu, S. Kaviya, N. P. Kumar, C. Rahul, and S. S. Raman, "Software defined radio (sdr) foundations, technology tradeoffs: A survey," in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, 2017, pp. 2677–2682.

[3] R. Zitouni and L. George, "Output power analysis of a software defined radio device," in *2016 IEEE Radio and Antenna Days of the Indian Ocean (RADIO)*, 2016, pp. 1–2.

[4] J. C. Martinez Quintero, E. P. Estupiñan Cuesta, and L. J. Ramirez Lopez, "A new method for the detection and identification of the replay attack on cars using sdr technology and classification algorithms," *Results in Engineering*, vol. 19, p. 101243, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590123023003705

[5] G. Kwon, J. Kim, J. Noh, and S. Cho, "Bluetooth low energy security vulnerability and improvement method," in *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, 2016, pp. 1–4.

[6] M. V, V. R, and A. Shukla, "Bluetooth vulnerabilities and security," in *2025 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 2025, pp. 1–6.

[7] J. Chukwuemeka Osuorah, S. T. Ahmed, A. Annamalai, and M. F. Chouikha, "Vulnerabilities and layered security threats in lorawan systems: A comprehensive survey on physical, mac, and application layer weaknesses," *IEEE Access*, vol. 13, pp. 217 939–217 953, 2025.

[8] Shailendra and K. T. Joseph, "Analysis on iot networks security : Threats, risks, esp8266 based penetration testing device and defense framework for iot infrastructure," in *2023 3rd International Conference on Intelligent Technologies (CONIT)*, 2023, pp. 1–7.

[9] M. Ahmad, M. A. Farid, S. Ahmed, K. Saeed, M. Asharf, and U. Akhtar, "Impact and detection of gps spoofing and countermeasures against spoofing," in *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2019, pp. 1–8.

[10] B. Yu and L.-Y. Zhang, "An improved detection method for different types of jamming attacks in wireless networks," in *The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014)*, 2014, pp. 553–558.

[11] N. Lovinger, T. Gerlich, Z. Martinasek, and L. Malina, "Detection of wireless fake access points," in *2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2020, pp. 113–118.

[12] F. Huo, P. Mitran, and G. Gong, "Analysis and Validation of Active Eavesdropping Attacks in Passive FHSS RFID Systems," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 7, pp. 1528–1541, Jul. 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7432027/

[13] H. Romesburg, J. Wang, Y. Jiang, H. Wang, and H. Song, "Software Defined Radio based Security Analysis For Unmanned Aircraft Systems," in *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, Oct. 2021, pp. 1–5, iSSN: 2374-9628. [Online]. Available: https://ieeexplore.ieee.org/document/9679408/

[14] N. Torres, P. Pinto, and S. I. Lopes, "Exploiting Physical Layer Vulnerabilities in LoRaWAN-based IoT Networks," in *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*, Oct. 2022, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10152098/

GNSS Spoofing Detection Using Discrete Wavelet Transform and Support Vector Machines

Nika Ćatipović, and Katarina Babić

Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture
University of Split
Split, Croatia

***Abstract*—Reliable GNSS spoofing detection is important because counterfeit signals can manipulate receiver position, timing, and navigation decisions. This paper evaluates a discrete wavelet transform based approach combined with support vector machine classification. Two feature representations, A1M1 and A1M2, are compared using Haar and Daubechies 12 wavelets and linear and Gaussian kernels. Experiments are performed with five-fold cross-validation and leave-one-scenario-out evaluation on the OAKBAT GPS dataset. The results show that A1M2 provides almost perfect performance when training and test samples originate from the same recordings, whereas A1M1 generalizes substantially better to the unseen os4 scenario. The findings demonstrate that evaluation protocol and feature representation have a strong influence on the measured effectiveness of GNSS spoofing detection.**

Keywords—GNSS, spoofing detection, discrete wavelet transform, support vector machine, OAKBAT.

I. Introduction

Global navigation satellite systems (GNSS) provide position, velocity, and precise timing services and are used in transportation, telecommunications, geodesy, emergency response, and other critical applications. Their dependence on weak satellite signals also makes them vulnerable to spoofing, in which counterfeit signals are generated so that a receiver accepts manipulated measurements as authentic. Detecting such attacks is therefore an important part of maintaining the integrity of GNSS-based services [1], [2].

GNSS spoofing detection becomes particularly challenging when counterfeit signals are designed to resemble authentic signals closely. In such cases, simple signal-level indicators may be insufficient, while representations that preserve localized changes in the signal can provide additional discriminative information. Wavelet-based processing combined with machine learning has therefore been investigated as a practical detection strategy. Recent work has demonstrated the usefulness of discrete wavelet transform features for spoofing detection and has examined the influence of wavelet and machine-learning choices on classification performance [3].

This paper investigates spoofing detection using a two-level two-dimensional discrete wavelet transform (DWT) followed by feature-based support vector machine (SVM) classification. The experiments compare two feature representations, A1M1 and A1M2, two wavelets, Haar and Daubechies 12 (db12), and two SVM kernels, linear and Gaussian. The key objective is to determine whether a model that performs well when training and test samples originate from the same recordings also generalizes to an unseen spoofing scenario.

The remainder of the paper is organized as follows. Section II describes the OAKBAT signal data and the wavelet-based processing procedure. Section III presents the feature extraction, SVM classification and evaluation protocols used in the experiments. Section IV presents and discusses the results, including comparisons of the feature representations, wavelets, kernels and evaluation protocols. Section V summarizes the main findings and directions for further work.

II. Signal Data and Wavelet-based Processing

A. GNSS Signal Data

The experiments use the Oak Ridge Spoofing and Interference Test Battery (OAKBAT) GPS dataset [4]. The recordings are sampled at 5 MHz and contain interleaved 16-bit in-phase (I) and quadrature (Q) samples. The complex baseband representation is given by

$$x(t) = I(t) + jQ(t) \tag{1}$$

where I(t) and Q(t) denote the in-phase and quadrature components, respectively. To avoid processing complex-valued samples, the magnitude of the complex baseband signal is calculated before applying the wavelet transform:

$$|x(t)| = \sqrt{I^2(t) + Q^2(t)}. \tag{2}$$

The resulting real-valued magnitude signal is then reshaped into a two-dimensional matrix and used as the input to the two-level 2-D discrete wavelet transform.

Four static recordings are considered: cleanStatic, os2, os3, and os4. The spoofing recordings contain an authentic interval followed by a spoofed interval. The spoofing power advantage is +10 dB for os2, +1.3 dB for os3, and +0.4 dB for os4; therefore, os4 represents the most demanding scenario in the dataset.

B. Wavelet Decomposition

The DWT analyzes a signal using scaled and translated versions of a mother wavelet [5], [6]. In the discrete formulation, the coefficient at scale j and position k can be expressed as

$$W_{j,k} = \sum |x[n]| \psi_{j,k}[n] \tag{3}$$

where ψ denotes the wavelet basis function. A two-level two-dimensional decomposition is applied to each signal block. At each level, approximation and detail coefficients are obtained by low-pass and high-pass filtering and downsampling. Following the experimental procedure, only the approximation and horizontal detail coefficients A1, H1, A2, and H2 are used for feature extraction. Two wavelets are compared: Haar and Daubechies 12.

The overall processing concept follows the wavelet-and-machine-learning framework investigated in Babić, Balić, and Begušić [3], while the present study focuses on the A1M1/A1M2 feature representations and the comparison of within-recording and unseen-scenario evaluation.

Each recording is processed in 4 ms segments. Consecutive groups of 125 segments are combined into 0.5 s blocks, corresponding to 2,500,000 samples per block. The samples are rearranged into a 1581 × 1581 matrix, with zero padding used for the remaining positions. Blocks at and around the spoofing onset are excluded when they contain both authentic and spoofed data.

C. Feature Extraction and SVM Classification

The A1M1 and A1M2 algorithms used in this study are based on the methodology described in [7]. A1M1 uses textural descriptors extracted from coefficient images, whereas A1M2 uses statistical and spectral descriptors computed directly from wavelet coefficients [7]. A1M1 extracts 32 textural features from images of A1, H1, A2, and H2. The features include histogram statistics, gradient statistics, and four gray-level co-occurrence matrix descriptors. Before feature computation, coefficient matrices are normalized and converted to grayscale images.

A1M2 extracts 40 statistical and spectral features directly from the four coefficient sets. These include mean, standard deviation, variance, extrema, skewness, kurtosis, spectral centroid, spectral spread, and spectral entropy. Consequently, A1M2 preserves information about the absolute coefficient level that is intentionally suppressed in A1M1 [7].

The resulting feature vectors are classified using an SVM [8]. The SVM determines a decision boundary that separates the two classes while maximizing the margin between the boundary and the closest training samples. These closest samples are the support vectors. For the linear kernel, the decision boundary is constructed directly in the feature space, whereas the Gaussian radial basis function (RBF) kernel enables nonlinear separation by measuring similarity between feature vectors. The Gaussian kernel can therefore adapt to curved class boundaries but may also be more sensitive to parameter selection. In both cases, feature standardization is performed using only the training data of the corresponding evaluation split, so that information from the test data does not influence the model.

III. Evaluation Methodology

A. Five-fold Cross-validation

The first protocol uses five-fold cross-validation repeated three times. Three binary experiments are performed separately by pairing cleanStatic with the spoofed portion of os2, os3, or os4. Each experiment contains 720 blocks per class, resulting in 1440 blocks. Training and test samples are drawn from the same two recordings, so the model can observe the recording-specific distributions during training. Table I summarizes the data partition used in each cross-validation experiment.

TABLE I. Data Partition for Five-Fold Cross-Validation

Class	Source	Number of blocks
Authentic	cleanStatic	720
Spoofed	spoofed portion of os2, os3, or os4	720
Total		1440
Training per iteration	4/5 of the total set	1152
Testing per iteration	1/5 of the total set	288

B. Leave-one-scenario-out Evaluation

The second protocol is a leave-one-scenario-out (LOSO) evaluation. One spoofing recording is excluded completely from training and parameter selection and is used only for testing. The training set contains 2880 blocks from cleanStatic and the authentic and spoofed portions of the other two spoofing recordings. The test set contains 240 authentic and 240 spoofed blocks from the unseen recording. The regularization parameter is selected by an inner validation procedure using only the training recordings. This protocol is intended to measure generalization to an unseen spoofing scenario. Table II summarizes the training and test partition used for the LOSO protocol. The complete processing and classification flow is shown in Fig. 1.

TABLE II. Data Partition for Leave-One-Scenario-Out Evaluation

Class / subset	Source	Number of blocks
Training		
Authentic	cleanStatic	960
Authentic	os2 (before 120 s)	240
Authentic	os3 (before 120 s)	240
Spoofed	os2 (after 120 s)	720
Spoofed	os3 (after 120 s)	720
Total		2880
Testing		
Authentic	os4 (before 120 s)	240
Spoofed	os4 (after 120 s)	240
Total		480

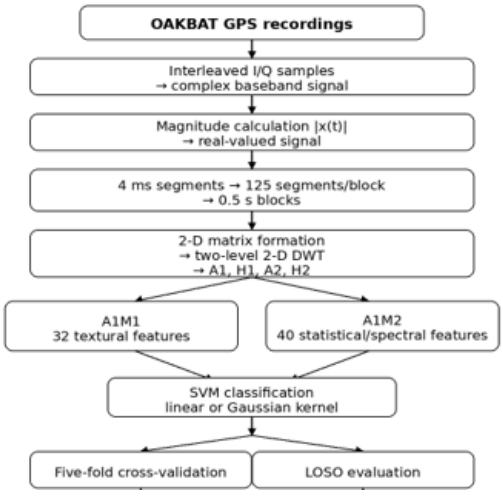


Fig. 1. Signal processing and classification flow used in the experiments.

C. Performance Measures

Performance is evaluated using accuracy, precision, recall, specificity, F1 score, ROC AUC, and PR AUC. Because the practical goal is to detect spoofed blocks, recall is particularly important: a low recall means that a large fraction of spoofed blocks is incorrectly accepted as authentic. Accuracy is defined as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \quad (4)$$

where TP and TN denote correctly detected spoofed and authentic blocks, respectively, while FP and FN denote false alarms and missed spoofed blocks.

Precision measures the proportion of predicted spoofed blocks that are actually spoofed:

$$Precision = \frac{TP}{TP+FP}. \quad (5)$$

Recall measures the proportion of spoofed blocks that are correctly detected:

$$Recall = \frac{TP}{TP+FN}. \quad (6)$$

Specificity measures the proportion of authentic blocks that are correctly identified:

$$Specificity = \frac{TN}{TN+FP}. \quad (7)$$

F1 score is the harmonic mean of precision and recall:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision+Recall}. \quad (8)$$

ROC AUC and PR AUC are threshold-independent summaries obtained as the areas under the ROC and precision–recall curves, respectively.

IV. Results and Discussion

A. Cross-validation Results

The five-fold results reveal a strong difference between the two feature representations in the hardest os4 scenario. A1M2 achieves almost perfect classification for all wavelet-kernel combinations, with accuracy from 99.93% to 100.00%. A1M1 is more dependent on the kernel: its linear-kernel configurations remain above 99.3% accuracy for os2-os4, whereas its Gaussian-kernel accuracy drops to 69.77% for Haar and 71.32% for db12 in os4. The corresponding recall values are 66.94% and 69.72%, showing that the main errors are missed spoofed blocks. The detailed os4 results are listed in Table III.

TABLE III. Five-Fold Results for the Hardest Scenario (OS4)

Configuration	Accuracy	Recall	F1	ROC AUC
A1M1 haar / linear	99.81%	99.68%	99.81%	100.00%
A1M1 haar / gaussian	69.77%	66.94%	68.90%	77.31%
A1M1 db12 / linear	99.31%	98.70%	99.30%	99.85%
A1M1 db12 / gaussian	71.32%	69.72%	70.81%	79.68%
A1M2 haar / linear	100.00%	100.00%	100.00%	100.00%
A1M2 haar / gaussian	100.00%	100.00%	100.00%	100.00%
A1M2 db12 / linear	99.98%	99.95%	99.98%	100.00%
A1M2 db12 / gaussian	99.93%	99.86%	99.93%	100.00%

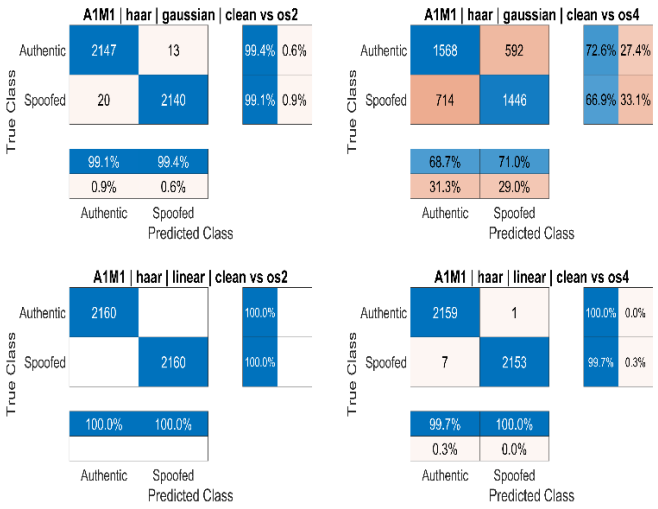


Fig. 2. Confusion matrices for A1M1 with the Haar wavelet and Gaussian or linear kernels for the os2 and os4 scenarios during five-fold cross-validation.

These results favor A1M2 under the k-fold protocol. For example, A1M2 with Haar achieves 100.00% accuracy, recall, and F1 score with both kernels, while A1M1 with Haar and the Gaussian kernel achieves only 69.77% accuracy and 66.94% recall. The difference is not caused by the wavelet alone; it is primarily associated with the feature representation and the chosen kernel.

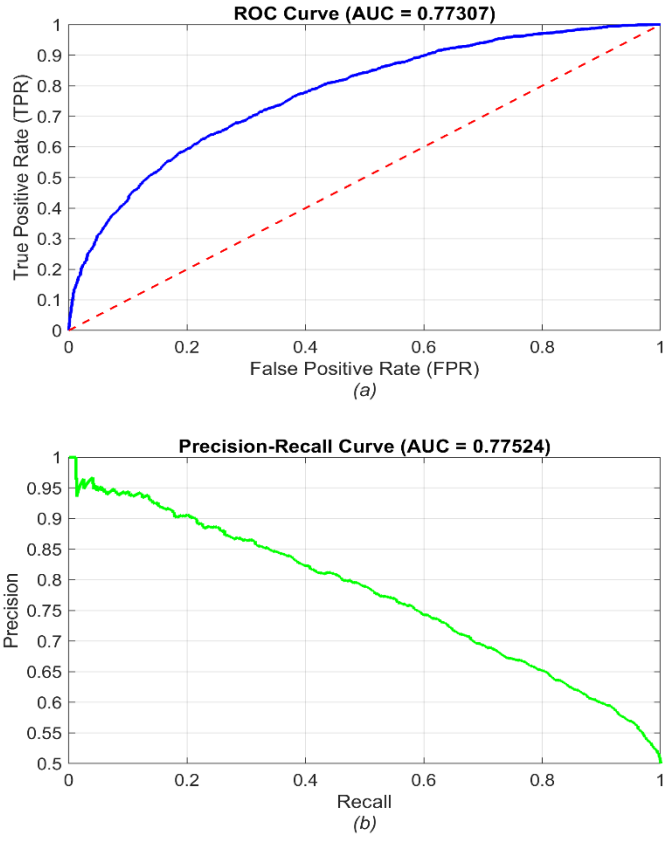


Fig. 3. ROC and precision-recall curves for the Haar wavelet, Gaussian kernel, and os4 scenario: (a) ROC curve; (b) PR curve.

The ROC and PR curves in Fig. 3 correspond to the same A1M1 configuration with the Haar wavelet, Gaussian

kernel, and os4 scenario shown by the os4 confusion matrix in Fig. 2. The ROC curve in Fig. 3(a) shows that the model retains a certain ability to distinguish authentic from spoofed signals, but class separability is considerably weaker than in the os2 scenario. The ROC AUC of 0.773 is nevertheless higher than 0.5, which corresponds to a random classifier. Therefore, the model is not completely ineffective, but provides a moderate ability to discriminate between the two classes. Similar behavior is observed in the precision-recall curve in Fig. 3(b). At lower recall values, precision remains high, indicating that, under a stricter decision threshold, most samples classified as spoofed are correctly identified. However, as recall increases, precision gradually decreases, indicating stronger overlap between authentic and spoofed samples in the extracted feature space. These results confirm that os4 is substantially more demanding for classification, particularly when the Gaussian kernel is used.

B. Leave-one-scenario-out Results

The interpretation changes substantially under LOSO evaluation. When one spoofing recording is excluded completely from training, A1M1 retains much stronger detection performance than A1M2, especially for os4. The best A1M1 configuration reaches 95.63% accuracy and 91.25% recall (Haar, linear), whereas the best A1M2 configuration reaches 62.71% accuracy and 25.42% recall (Haar, Gaussian). Thus, the difference is not merely an overall accuracy effect: A1M2 misses most spoofed blocks in the unseen scenario, while A1M1 detects the majority of them. The complete os4 comparison is given in Table IV.

TABLE IV. LOSO RESULTS WITH OS4 AS THE UNSEEN TEST SCENARIO

Configuration	Accuracy	Recall	F1	ROC AUC
A1M1 haar / linear	95.62%	91.25%	95.42%	100.00%
A1M1 haar / gaussian	86.04%	74.58%	84.24%	97.88%
A1M1 db12 / linear	92.92%	85.83%	92.38%	100.00%
A1M1 db12 / gaussian	75.00%	52.92%	67.91%	93.98%
A1M2 haar / linear	54.17%	8.33%	15.38%	93.38%
A1M2 haar / gaussian	62.71%	25.42%	40.53%	99.99%
A1M2 db12 / linear	51.25%	2.50%	4.88%	97.21%
A1M2 db12 / gaussian	60.21%	20.42%	33.91%	100.00%

The confusion matrices show why accuracy alone is insufficient in this setting. For A1M2 with Haar and a linear kernel, 240 authentic blocks are correctly identified, but only 20 of 240 spoofed blocks are detected. The resulting 8.33% recall is therefore much more informative for spoofing detection than the 54.17% accuracy by itself. A similar pattern occurs for db12, where the linear A1M2 configuration detects only 6 of 240 spoofed blocks, producing 2.50% recall.

The direct comparison in Fig. 4 further illustrates the effect of the evaluation protocol. For A1M2 with db12 and a linear kernel in the os4 scenario, five-fold cross-validation produces almost perfect classification, whereas LOSO evaluation detects only 6 of 240 spoofed blocks. This sharp deterioration shows that performance estimated from within-recording partitions can be overly optimistic when the detector is applied to a previously unseen spoofing recording.

C. Comparison of Algorithms, Wavelets, and Kernels

The comparison between A1M1 and A1M2 depends on the evaluation protocol. In k-fold cross-validation, A1M2 is clearly superior because it achieves almost perfect performance in all tested configurations. However, this advantage does not transfer to the unseen-scenario setting. Under LOSO, A1M1 provides substantially higher recall and F1 scores for os4 across both wavelets and both kernels. Thus, A1M2 should not be interpreted as the more effective detector solely from its k-fold results.

The comparison between wavelets does not reveal a universal winner. For A1M1 with a linear kernel, Haar provides a slightly higher recall than db12 in the difficult LOSO os4 case (91.25% versus 85.83%), while db12 remains competitive in easier cases. For A1M2, differences between Haar and db12 are small compared with the much larger performance change caused by switching from k-fold to LOSO evaluation. The results therefore suggest that wavelet selection is less influential than the choice of feature representation and the evaluation protocol for generalization.

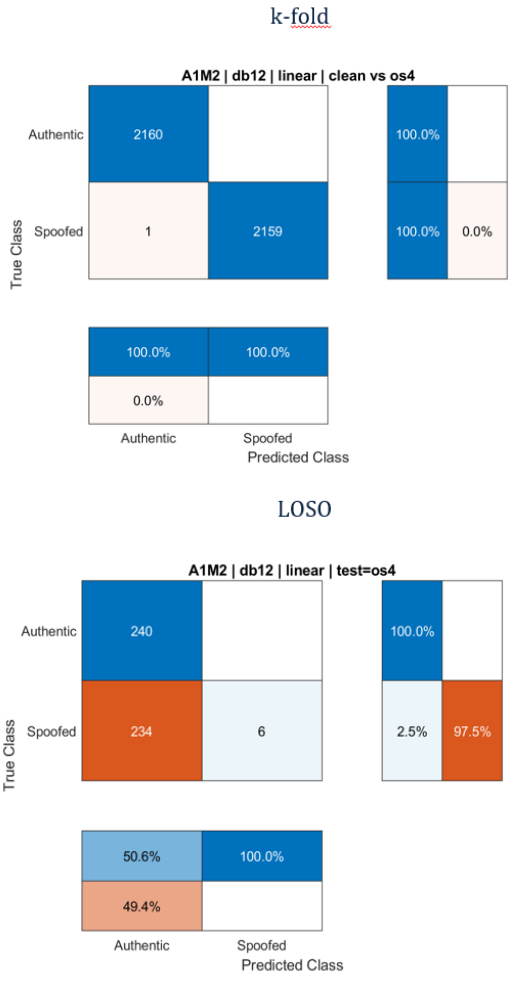


Fig. 4. Comparison of five-fold and LOSO evaluation for A1M2 with db12 and a linear kernel in the os4 scenario.

The kernel comparison is also protocol-dependent. For A1M1, the linear kernel is consistently stronger than the Gaussian kernel in the difficult scenarios, particularly for os4. For A1M2, neither kernel resolves the unseen-scenario problem: recall remains low for os4 with both wavelets. The main performance loss occurs when the evaluation setting changes from within-recording k-fold validation to an unseen

recording. The best recall configuration for each algorithm and scenario is summarized in Table V.

TABLE V. BEST SPOOFED-CLASS RECALL CONFIGURATION BY ALGORITHM AND SCENARIO

Algorithm / protocol / test	Wavelet / kernel	Accuracy	Recall
A1M1 / k-fold / os2	haar / linear	100.00%	100.00%
A1M1 / k-fold / os3	haar / linear	99.86%	99.72%
A1M1 / k-fold / os4	haar / linear	99.81%	99.68%
A1M2 / k-fold / os2	haar / linear	100.00%	100.00%
A1M2 / k-fold / os3	haar / linear	100.00%	100.00%
A1M2 / k-fold / os4	haar / linear	100.00%	100.00%
A1M1 / LOSO / os2	haar / linear	97.50%	100.00%
A1M1 / LOSO / os3	db12 / linear	99.38%	98.75%
A1M1 / LOSO / os4	haar / linear	95.62%	91.25%
A1M2 / LOSO / os2	haar / linear	97.92%	100.00%
A1M2 / LOSO / os3	haar / linear	99.58%	99.17%
A1M2 / LOSO / os4	haar / gaussian	62.71%	25.42%

For each algorithm and scenario, the table reports the configuration with the highest recall, using recall as the primary criterion because missed spoofed blocks are the most critical classification error for detection.

ACKNOWLEDGEMENT

This work was supported by the project "Advanced modeling and measurement procedures in engineering multiphysics problems (MULTIMOD)".

V. CONCLUSION

This paper evaluated GNSS spoofing detection using two-level two-dimensional DWT processing and SVM classification on the OAKBAT GPS dataset. Two feature-based algorithms, A1M1 and A1M2 [7], were compared using Haar and db12 wavelets and linear and Gaussian SVM kernels. The study follows the broader DWT-and-machine-learning direction described by Babić, Balić, and Begušić [3], while focusing on the behavior of the two feature representations under different evaluation protocols.

The main result is the strong discrepancy between k-fold and LOSO evaluation. A1M2 achieves almost perfect classification when training and test samples originate from the same recordings, but its performance deteriorates sharply when os4 is presented as an unseen recording. A1M1 generalizes substantially better, particularly with the linear kernel. The ROC and precision-recall analysis for the Haar-Gaussian os4 case also shows that the difficult scenario still retains moderate class separability (ROC AUC about 0.773), although the overlap between authentic and spoofed samples increases and precision decreases as recall is raised. Thus, os4 remains the most demanding scenario, especially for the Gaussian kernel.

For practical spoofing detection, the LOSO results are therefore more informative than the near-perfect k-fold results because they test whether the detector can recognize an attack that was not present during training. Future work should evaluate more recordings and attack types, optimize kernel parameters more extensively, and investigate feature normalization or threshold adaptation methods that reduce the influence of recording-specific coefficient levels.

REFERENCES

[1] M. L. Psiaki and T. E. Humphreys, "GNSS spoofing and detection," Proc. IEEE, vol. 104, no. 6, pp. 1258–1270, 2016.

[2] K. Radoš, M. Brkić, and D. Begušić, "Recent advances on jamming and spoofing detection in GNSS," Sensors, vol. 24, no. 13, p. 4210, 2024.

[3] K. Babić, M. Balić, and D. Begušić, "GNSS Spoofing Detection Based on Wavelets and Machine Learning," Electronics, vol. 14, no. 12, p. 2391, 2025, doi: 10.3390/electronics14122391.

[4] A. Albright, S. Powers, J. Bonior, and F. Combs, "Oak Ridge Spoofing and Interference Test Battery (OAKBAT) – GPS," Oak Ridge National Laboratory, 2020.

[5] C. S. Burrus, R. A. Gopinath, and H. Guo, Introduction to Wavelets and Wavelet Transforms: A Primer, 2nd printing. Prentice Hall, 1999.

[6] I. Daubechies, Ten Lectures on Wavelets. Philadelphia, PA, USA: SIAM, 1992.

[7] K. Babić, "Detekcija lažiranog signala u globalnom navigacijskom satelitskom sustavu," Ph.D. dissertation, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, Split, Croatia, 2025.

[8] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, pp. 273–297, 1995.

[9] N. O. Fedorova, D. A. Andreev, D. A. Tsapaev, and M. A. Kiryakina, "Detecting GPS Spoofing on Different Devices Using RaspberryPi with LimeSDR," in Proc. 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2020.

Integrated Technologies for Wildfire Prevention and Emergency Response

Slavko Tucaković
Chief Fire Commander
Croatian Firefighting Association
Zagreb, Croatia
slavko.tucakovic@gmail.com

***Abstract*—Wildfire management increasingly requires rapid acquisition, reliable transmission and effective interpretation of heterogeneous information from the field. Recent research has shown the potential of unmanned aerial vehicles (UAVs), optical and thermal sensing, artificial intelligence (AI), satellite observations, and geographic information systems (GIS) to improve wildfire prevention, detection, and response [1]–[5]. However, the operational value of these technologies depends on integrating them into resilient communications, decision-support processes, and professional procedures rather than on individual technological components alone [6], [7]. This paper analyses the need for advanced technologies from the perspective of fire and rescue services and presents an integrated operational concept developed within the SmartProtect project. The approach covers prevention and preparedness, active firefighting operations, and post-fire assessment, combining UAVs, resilient communications, GIS, computer vision (CV), AI, and high-performance data processing. Practical use cases include real-time UAV reconnaissance, thermal hot-spot detection, AI-assisted analysis and satellite-based burn-area mapping. The paper argues that future wildfire-management systems should be designed as integrated human-in-the-loop decision-support environments in which automated technologies support, but do not replace, professional operational decision-making.**

Keywords—wildfire management, UAV, AI, computer vision, GIS, emergency communications, thermal imaging, decision support.

I. Introduction

Wildfires are among the most demanding natural hazards affecting Mediterranean and other fire-prone regions. Their rapid development, large spatial extent, difficult terrain, and strong dependence on meteorological conditions create a constantly changing operational environment. Fire and rescue services must therefore make decisions under time pressure while maintaining personnel safety and coordinating multiple agencies and resources. Traditional field observation, professional radio communication, and incident-command experience remain essential, but increasingly need to be complemented by remote sensing, digital communications, and decision-support tools.

UAV-based remote sensing has become particularly important because it enables rapid observation of areas that may be inaccessible or dangerous for ground personnel. Bouguettaya et al. [1] reviewed the use of UAV imagery and deep-learning computer vision for early wildfire detection, while Allison et al. [2] highlighted the value of airborne optical and thermal sensing for wildfire detection and monitoring. More recent reviews show that deep learning has substantially expanded visual fire and smoke analysis, but also underline persistent limitations related to datasets, false alarms, generalization, and real-world validation [3], [4].

At the same time, advanced sensing alone does not provide effective decision support. Operational value depends on whether data can be transmitted reliably, geolocated, interpreted, verified, and presented to commanders in a usable form. GIS-based wildfire decision-support systems have demonstrated the benefits of integrating multi-sensor data and collaborative spatial information for faster, better-informed suppression decisions [5]. Emergency communications research likewise shows that no single wireless technology can satisfy all requirements of public-safety operations [6], [7]. The SmartProtect project adopts this integrated perspective. Instead of treating UAVs, AI, communication systems, or GIS as isolated technologies, it connects sensing, communication, processing, verification, and command-level decision support.

This paper therefore focuses on the operational need for technology across the wildfire lifecycle, presents the resulting integrated concept, and discusses practical use cases and lessons relevant to future digitalization of the fire service.

II. Operational Needs Throughout the Wildfire Lifecycle

A. Prevention and Preparedness

Before a wildfire occurs, technology can support inspections of fire-protection roads, vegetation, and infrastructure corridors; detection of illegal waste deposits; monitoring of high-risk areas; and maintenance of current GIS databases. UAVs are particularly useful when very high spatial resolution or rapid local inspection is needed, while satellite systems provide repeated, large-area observations. The two approaches are therefore complementary rather than competing.

B. Active Firefighting Operations

During active suppression, the priority shifts from systematic mapping to timely situational awareness. Incident commanders need information on fire location and behavior, inaccessible fire fronts, smoke conditions, critical infrastructure, access routes, and deployed resources. UAVs can provide RGB and thermal imagery while reducing the need to expose personnel to hazardous reconnaissance tasks [1], [2]. AI-assisted analysis may further reduce operator workload by prioritizing potential smoke, flame, or other relevant events, but automated outputs remain sensitive to environmental variability and dataset limitations [3], [4].

C. Post-Fire Operations

After the main fire is suppressed, residual hot spots and possible reignition become the primary concern. Thermal UAV sensors can support local inspection, while satellite before/after analysis can support mapping of the affected area and burn severity. Combined Landsat and Sentinel-2 data have been shown to provide useful initial burn-severity products based on NBR-related indices [8].

TABLE I
TECHNOLOGICAL SUPPORT THROUGHOUT THE WILDFIRE LIFECYCLE

Phase	Operational need	Technology
Prevention	Risk and infrastructure monitoring	UAV, GIS, AI, satellite
Preparedness	Updated spatial information	GIS, UAV mapping, HPC
Response	Real-time situational awareness	UAV, thermal, communications
Response	Command and coordination	GIS, live video, radio/data
Post-fire	Hot-spot detection	Thermal UAV, GIS
Post-fire	Burn-area assessment	Satellite, GIS, EO analytics

III. INTEGRATED TECHNOLOGY FRAMEWORK

A. UAV and Remote Sensing

Different UAV classes serve distinct operational roles. Smaller systems are suited for rapid deployment and short-duration local reconnaissance, whereas professional platforms equipped with optical zoom, radiometric thermal sensors, and RTK positioning support more demanding mapping and post-fire missions. The SmartProtect concept therefore treats UAVs as mobile sensing platforms within a broader information system. This aligns with the literature, which identifies high spatial resolution, rapid deployment, and flexible sensor integration as key advantages but also notes constraints related to endurance, weather, communication range, and airspace coordination [1], [2]. Fig. 1 summarizes the role of these technologies across prevention, preparedness, active response, post-fire assessment, and recovery.

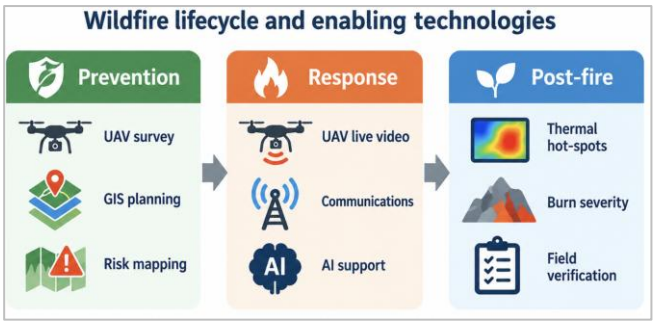


Fig. 1. Wildfire lifecycle and the role of UAV, GIS, AI and communication technologies. *Generated using ChatGPT Images 2.5 (OpenAI)*

B. Resilient Communications

Communication is the technical backbone of an integrated response system. Voice traffic, command messaging, GIS data, and live UAV video impose distinct requirements for coverage, bandwidth, latency, and resilience. Pervez et al. [6] showed that emergency-response systems benefit from selecting different wireless technologies for different operational functions. Research on 5G for Public Protection and Disaster Relief similarly confirms the potential of broadband cellular networks while recognizing the ongoing importance of proven professional mobile radio for mission-critical voice [7]. For that reason, SmartProtect separates mission-critical radio, local command-vehicle LAN/Wi-Fi, terrestrial LTE/5G broadband and satellite connectivity. Satellite communications and emerging non-terrestrial networks can provide an additional resilience layer when terrestrial infrastructure is unavailable or overloaded [9]. The principle is not to replace one technology with another, but to combine complementary communication paths.

C. GIS and Common Operational Picture

GIS provides the spatial integration layer for heterogeneous data sources. Relevant information may include UAV products, thermal anomalies, roads, power lines, crew locations, AI-generated candidates, and post-fire burn maps. The objective is not merely to display more data but to transform sensor data into a common operational picture. This approach aligns with GIS-based wildfire decision-support research, in which interoperability and collaborative data sharing are central system requirements. [5].

D. Computer Vision, AI and Human Verification

Computer vision can automatically extract candidate information from large image and video streams using classification, object detection, segmentation, and tracking. Current reviews highlight strong technical progress in visual fire detection but also indicate that robustness and generalization remain challenging when operating conditions differ from the training data [3], [4]. In safety-relevant settings, model confidence should therefore be interpreted alongside visual evidence, sensor context, and the consequences of errors. SmartProtect uses a human-in-the-loop approach: AI candidate → human verification → GIS → operational decision. Human-in-the-loop machine learning is a broad class of methods in which domain experts remain involved in learning, correction, or decision-making [10]. In wildfire operations, this principle preserves professional responsibility while using AI to accelerate screening and prioritization.

E. HPC and Data Processing

High-resolution UAV imagery, orthophotos, satellite products, and AI models can generate data volumes that are difficult to process efficiently on field devices. Centralized HPC/GPU resources can support photogrammetry, large-image tiling, AI inference, model training, and satellite image processing. The field layer can therefore focus on acquisition and communication, while computationally demanding processing occurs centrally. The system then shares the resulting information through GIS and web applications.

F. Operational Data States and Traceability

An integrated system should preserve the status and provenance of information as it moves from sensing to decision support. SmartProtect distinguishes among raw sensor data,

automatically generated candidates, reviewed information, and confirmed operational findings. Each result should retain basic metadata, including the source sensor, acquisition time, geolocation, processing method or model version, reviewer status, and relevant thresholds. This traceability matters for AI-assisted outputs because two visually similar results may stem from different model versions or acquisition conditions. The same principle applies to field communication. A live UAV stream may support immediate situational awareness, whereas a georeferenced orthomosaic or burn-severity product may be generated later for planning and documentation. The architecture should therefore support both time-critical, low-latency products and slower analytical products without conflating their intended operational roles. This separation reduces cognitive overload and supports later review of how a decision was reached.

IV. SmartProtect Integrated Operational Concept

The resulting SmartProtect concept integrates sensing, communication, processing, and command functions into a single operational data flow. UAV, satellite, and field-sensor data are collected at the edge, transmitted over available communication channels, processed by central computing services, integrated with GIS and AI/CV applications, and finally presented to command personnel and field teams as verified operational information. The information flow is bidirectional: a commander may request a new UAV task, a closer thermal inspection, or field verification based on previously received information. The complete end-to-end information flow is illustrated in Fig. 2.

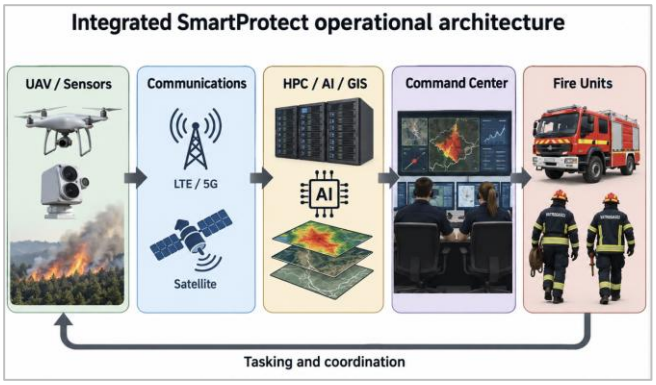


Fig. 2. Integrated SmartProtect architecture: sensing → communications → HPC/GIS/AI → command → field teams. *Generated using ChatGPT Images 2.5 (OpenAI)*

A key characteristic of the concept is the separation between raw or automatic data and operationally verified information. Automated outputs are marked as candidates until reviewed under the defined workflow. This avoids a common weakness in technology-centric systems, where an algorithmic result is presented without sufficient context to support a safe operational decision.

V. Practical Use Cases

A. UAV-Based Wildfire Reconnaissance

Rapid UAV reconnaissance follows the chain of mission planning → launch → RGB/thermal acquisition → communication → command display → assessment. UAVs can provide high-resolution data at low altitude without directly exposing a reconnaissance crew to the hazard [1], [2]. Their operational limitations remain significant: wind, smoke, battery endurance, airspace coordination, and data-link quality must all be considered by the incident commander.

B. Post-Fire Hot-Spot Detection

After suppression, thermal sensors can identify anomalous surface temperatures that may indicate residual combustion. In SmartProtect, the workflow is thermal candidate → RGB/zoom verification → operator review → GIS geolocation → field verification. A thermal anomaly is not automatically treated as an active fire because sun-heated rocks, metal structures or other warm objects can generate misleading signatures. Multi-sensor and human verification are therefore integral parts of the workflow. Fig. 3 illustrates this post-fire verification chain.

C. Burn-Area Mapping

For large-area post-fire assessment, satellite analysis provides a scalable complement to UAV inspection. The workflow uses pre-fire and post-fire Sentinel-2 scenes, calculation of NBR/dNBR-type indicators, severity classification and GIS export. Previous studies demonstrate that combining Landsat and Sentinel-2 can support rapid initial burn-severity mapping [8]. The resulting maps should be interpreted as analytical products rather than direct measurements of every local fire effect, and the method, thresholds, and acquisition dates should remain traceable.

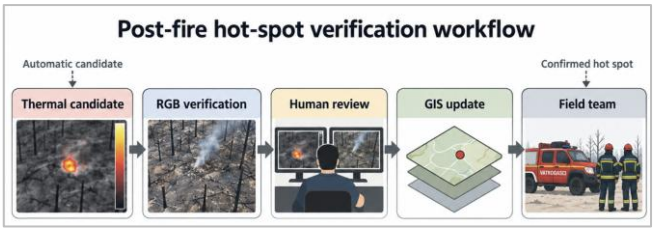


Fig. 3. Post-fire hot-spot workflow: thermal candidate → RGB verification → human review → GIS → field team. *Generated using ChatGPT Images 2.5 (OpenAI)*

D. AI-Assisted Preventive Monitoring

The same infrastructure supports preventive applications such as waste detection, corridor inspection, and vegetation monitoring. A typical workflow is UAV acquisition → annotation → dataset → model training → inference → human review → GIS. The key lesson is that AI performance depends on the target domain: models trained on generic datasets should be validated against the actual geographic area, platform, sensor, and acquisition conditions before operational use. Recent fire-detection surveys likewise emphasize the importance of realistic datasets and validation scenarios [3], [4].

VI. Operational Lessons and Training Requirements

Several lessons emerge from the SmartProtect implementation. First, integrate technology into operational procedures. Key questions include who receives the

information, who verifies it, how it reaches command staff, and what decision it is expected to support. Recent research on wildfire decision-support tools identifies usability, trust, training, organizational capacity, and integration into existing practice as major factors influencing real-world effectiveness [11]. Second, communication must be redundant. Live video and high-resolution imagery require broadband capacity, but essential command and safety functions cannot depend on a single broadband service. A layered architecture combining professional radio, terrestrial broadband and satellite backup is more resilient than a single-network solution [6], [7], [9]. Fig. 4 shows the corresponding resilient field-to-command communication concept.

Third, AI outputs require controlled interpretation. False positives, false negatives, and domain shift are not exceptional edge cases but normal engineering concerns in real environments [3], [4]. Human verification is therefore part of the system architecture, not an optional post-processing step [10].

Fourth, training should be role-based. Not every firefighter needs to become an AI specialist or UAV pilot. Field users, UAV operators, communication operators, GIS/AI specialists and commanders require different levels of competence. SmartProtect WP3 therefore developed a general training curriculum and training materials together with specialized UAV, communications and CV/AI manuals. This structure is intended to ensure that new technologies can be used safely and consistently within established command structures.

For commanders, the key competence is not operating every sensor in detail but understanding what information a technology can provide, how quickly it can provide it, and its principal limitations. For example, an incident commander should understand the difference between a thermal anomaly and a confirmed hot spot, between an AI candidate and a verified event, and between a live video view and a georeferenced mapping product. These distinctions directly affect whether information can support tactical decisions or should instead trigger additional verification.

Training must also include degraded-mode operation. A realistic exercise should consider loss of broadband connectivity, reduced UAV endurance, incomplete GIS data, false AI alarms, and the need to continue mission-critical communication through professional radio. Such scenarios test whether the technology actually strengthens resilience or merely adds dependencies. The recent literature on decision-support adoption similarly indicates that user trust and organizational fit depend on practical experience, transparency, and integration into established work practices [11].

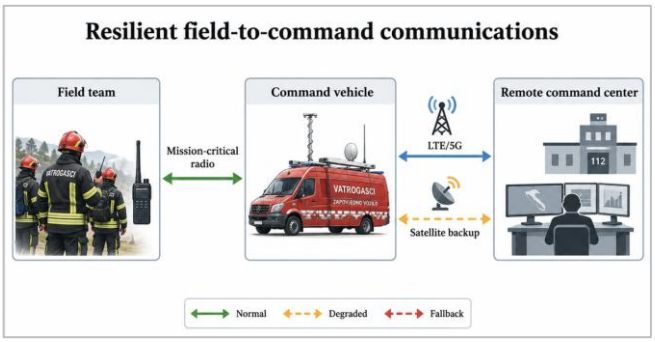


Fig. 4. Resilient field-to-command communications: mission-critical radio + LTE/5G + satellite backup. *Generated using ChatGPT Images 2.5 (OpenAI)*

VII. Proposed Operational Evaluation Framework

A. From Component Metrics to End-to-End Performance

A technology may perform well in isolation and still fail to improve an emergency operation. For that reason, evaluation should include both component-level and end-to-end criteria. UAV platforms can be characterized by endurance, sensor resolution, thermal capability and deployment time. Communication links can be measured by coverage, throughput, latency, availability and recovery time. AI models can be measured by precision, recall and robustness. GIS applications can be evaluated by update time, positional accuracy and usability. These metrics remain necessary, but they do not directly describe whether the complete system delivers useful information to a commander at the right time.

For operational evaluation, the SmartProtect concept therefore proposes measuring the complete information chain from observation to verified operational display. A representative metric is time-to-information: the interval from sensor acquisition to the moment when a verified result is available in the command environment. A second metric is continuity under degradation, defined by whether essential information and mission-critical voice remain available when one communication path fails. A third metric is verification effort, expressed through the number of operator actions, cross-checks, or additional sensor requests needed before a candidate becomes a confirmed finding. These measures connect technical performance with operational workload.

B. Scenario-Based Validation

Realistic scenarios are required because wildfire technologies interact with terrain, weather, smoke, infrastructure, and organizational procedures. A laboratory network test cannot fully represent a mountainous incident area, and an AI model validated on curated imagery may behave differently when haze, compression, motion blur or unfamiliar vegetation are present. Field exercises should therefore reproduce the conditions under which the system is expected to operate. Scenarios can include a rapidly developing wildfire, a post-fire hot-spot search, preventive corridor inspection, and a communication-degraded command-post operation.

Each scenario should contain predefined injects that test system resilience. Examples include loss of the primary

LTE/5G path, temporary satellite obstruction, reduction of UAV battery reserve, a false AI candidate, incomplete georeferencing, or a request for a second sensor view. The aim is not to make the exercise artificially difficult, but to verify that operators understand fallback procedures and that the integrated system degrades gracefully rather than failing abruptly. Lessons from such exercises should feed back into communication plans, software configuration, training materials and standard operating procedures.

C. Decision Quality and User-Centred Assessment

The system's final purpose is to support decisions, so evaluation should also consider information quality from the user perspective. Commanders need concise, spatially referenced and clearly qualified information. An interface that displays many raw feeds can increase rather than reduce cognitive load. Useful assessment criteria include whether the source and status of a finding are visible, whether uncertain information is clearly distinguished from confirmed information, whether operators can trace a result back to its source image, and whether the system enables operators to request a new task or field verification without leaving the operational workflow.

User-centered evaluation should involve the actual roles that will operate the system. UAV pilots can assess mission planning and sensor workflows; communication personnel can assess failover and network monitoring; GIS/AI specialists can assess data quality and model outputs; and incident commanders can assess relevance, timeliness and decision support. This role-based evaluation aligns with the SmartProtect training concept and recent wildfire decision-support research that emphasizes usability, trust, organizational capacity, and integration into established practices [11].

VIII. Implementation Considerations and Limitations

A. Safety, Airspace and Mission Planning

The introduction of UAVs into wildfire operations does not remove the need for conventional aviation safety procedures. Wildfire airspace may include helicopters, fixed-wing firefighting aircraft, police or medical aviation and other unmanned systems. UAV missions must therefore be planned within the applicable regulatory and incident-command framework, with clear coordination of launch areas, maximum altitude, operating zones and termination criteria. The operational benefit of a thermal or zoom sensor cannot justify creating an additional aviation hazard. Mission planning should also include lost-link behavior, reserve battery requirements, emergency landing areas, and criteria for suspending the flight because of wind, smoke, or visibility.

A second limitation is that UAV endurance and sensor performance can change rapidly during an incident. High wind increases power consumption, smoke reduces optical contrast, thermal interpretation may be affected by sun-heated surfaces, and the best observation point may not have reliable broadband connectivity. The command process should therefore treat UAV information as one source among several and should preserve alternative observation methods when unmanned operations are temporarily unavailable.

B. Communications and Degraded Operations

Broadband connectivity is highly valuable because it enables live video, map updates, and access to centrally processed products, but it should not become a single point of operational failure. A command vehicle may have several WAN options and still experience degraded connectivity because of terrain shielding, network congestion, local infrastructure damage, or temporary satellite obstruction. The system design must define which services are essential, which can tolerate delay, and which can be suspended when capacity is limited.

In practical terms, mission-critical voice and basic command messages should remain available even if high-bandwidth video is lost. Local recording can preserve UAV evidence for later transfer, while selected still images or low-rate metadata may provide more operational value than attempting to maintain a full-resolution stream over a constrained link. This prioritization should be part of communication planning and not an improvised decision made only after network degradation occurs.

C. Data Quality, AI Limitations and Accountability

AI-assisted wildfire systems also require explicit limits on how automated outputs are presented and used. A confidence value is not a universal probability that a result is correct; it depends on the model, dataset composition, threshold settings, and the target domain. Operators should therefore see the original image or sensor context together with the automated overlay whenever possible. The status of a result should be clear, for example, automatic candidate, reviewed-probable, confirmed, or rejected. Such status labels prevent automated detections from being silently transformed into operational facts.

Manage data quality throughout the lifecycle. Separate training and validation set to avoid leakage between nearly identical UAV frames or repeated missions. New sensor types, seasons, geographic locations, and acquisition heights may introduce domain shift and should trigger targeted validation. Model versions, thresholds, and preprocessing settings should be recorded so results can later be reproduced or explained. These measures are especially important if AI outputs influence resource prioritization or field verification tasks.

D. Organizational Adoption and Sustainability

Finally, technology adoption requires an organizational model for maintenance, updates, and support. Fire services should know who is responsible for UAV firmware and batteries, radio programming, GIS layers, AI model releases, server infrastructure and user accounts. A system that performs well during a pilot project but cannot be maintained after project completion has limited long-term value. SmartProtect therefore links technical deployment with training materials, maintenance procedures and specialist roles that can support operational users.

Sustainability also requires controlled evolution. New AI models, communication equipment or software versions should not be introduced directly into operational use without validation. A practical governance process should distinguish development versions from approved operational versions and record test results, known limitations, and release dates. This approach reduces configuration drift between partner organizations and supports cross-border exercises in which all participants need to understand the status and capabilities of the systems being used.

IX. Discussion

The SmartProtect experience suggests that future digital firefighting systems should focus less on individual devices and more on interoperability. An advanced UAV without a reliable data link has limited operational value. Similarly, an AI model without human verification and GIS integration may generate technically plausible outputs that do not support the operational process. A useful system must therefore connect sensing, communications, computing, spatial context and trained personnel. This integrated approach also has implications for cross-border response. Wildfires do not respect administrative boundaries, while neighboring services may use different communication infrastructures, data formats and procedures. Modular architectures and standard interfaces can reduce these barriers. In future work, the SmartProtect concept should therefore be evaluated through realistic field exercises that measure not only sensor accuracy but also end-to-end latency, communication continuity, workload, decision quality and the usability of information under degraded conditions.

Cross-border interoperability should be considered at three levels. The technical level concerns network reachability, data exchange, coordinate systems and common geospatial services. The operational level concerns terminology, status labels, responsibility for verification and procedures for requesting additional sensing or field checks. The organizational level concerns training, access rights, information ownership and the authority to act on shared information. Addressing all three levels is necessary if an integrated digital platform is to remain useful when an incident involves several agencies or countries.

A further research need is a quantitative assessment of the complete information chain. Many studies evaluate individual components such as detector precision or communication throughput. For fire-service adoption, however, end-to-end measures are also important: how long it takes from observation to verified display, how many operator actions are required, whether data remain available after a network failure, and whether the resulting information changes or improves a tactical decision. These measures should complement conventional technical metrics in future SmartProtect field trials.

X. Conclusion

Modern wildfire management requires fire and rescue services to complement traditional procedures with advanced digital technologies. The literature demonstrates substantial potential for UAV-based remote sensing, deep-learning fire detection, GIS decision support, broadband public-safety communications and satellite-based post-fire analysis [1]–[9]. At the same time, recent research stresses that successful adoption depends on usability, interoperability, organisational integration and training [10], [11].

The SmartProtect project addresses these requirements through an integrated operational concept combining UAV and satellite sensing, resilient communications, HPC-supported processing, GIS, CV/AI and human verification. The main conclusion is that the value of new technology depends less on individual devices than on the quality of the complete information chain from sensor to operational decision. Future work will focus on field validation, quantitative performance assessment, cross-border interoperability and continued integration of these tools into standard operating procedures and training.

Acknowledgment

This work was supported by the SmartProtect – Contemporary Technologies for Natural and Man-made Disaster Management project, implemented within the Interreg VI-A IPA Croatia–Bosnia and Herzegovina–Montenegro Programme 2021–2027, project ID HR-BA-ME00392.

References

[1] A. Bouguettaya, H. Zarzour, A. M. Taberkit, and A. Kechida, “A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms,” Signal Processing, vol. 190, Art. no. 108309, 2022, doi: 10.1016/j.sigpro.2021.108309.

[2] R. S. Allison, J. M. Johnston, G. Craig, and S. Jennings, “Airborne optical and thermal remote sensing for wildfire detection and monitoring,” Sensors, vol. 16, no. 8, Art. no. 1310, 2016, doi: 10.3390/s16081310.

[3] G. Cheng, X. Chen, C. Wang, X. Li, B. Xian, and H. Yu, “Visual fire detection using deep learning: A survey,” Neurocomputing, vol. 596, Art. no. 127975, 2024, doi: 10.1016/j.neucom.2024.127975.

[4] D. Gragnaniello, A. Greco, C. Sansone, and B. Vento, “Fire and smoke detection from videos: A literature review under a novel taxonomy,” Expert Systems with Applications, vol. 255, Art. no. 124783, 2024, doi: 10.1016/j.eswa.2024.124783.

[5] M. Lourenço, L. B. Oliveira, J. P. Oliveira, A. D. Mora, H. Oliveira, and R. Santos, “An integrated decision support system for improving wildfire suppression management,” ISPRS Int. J. Geo-Inf., vol. 10, no. 8, Art. no. 497, 2021, doi: 10.3390/ijgi10080497.

[6] F. Pervez, J. Qadir, M. Khalil, T. Yaqoob, U. Ashraf, and S. Younis, “Wireless technologies for emergency response: A comprehensive review and some guidelines,” IEEE Access, vol. 6, pp. 71814–71838, 2018, doi: 10.1109/ACCESS.2018.2878898.

[7] M. Volk and J. Sterle, “5G experimentation for public safety: Technologies, facilities and use cases,” IEEE Access, vol. 9, pp. 41184–41217, 2021, doi: 10.1109/ACCESS.2021.3064405.

[8] C. Quintano, A. Fernández-Manso, and O. Fernández-Manso, “Combination of Landsat and Sentinel-2 MSI data for initial assessing of burn severity,” Int. J. Appl. Earth Obs. Geoinf., vol. 64, pp. 221–225, 2018, doi: 10.1016/j.jag.2017.09.014.

[9] O. Kodheli et al., “Satellite communications in the new space era: A survey and future challenges,” IEEE Commun. Surveys Tuts., vol. 23, no. 1, pp. 70–109, 2021, doi: 10.1109/COMST.2020.3028247.

[10] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, and Á. Fernández-Leal, “Human-in-the-loop machine learning: A state of the art,” Artif. Intell. Rev., vol. 56, pp. 3005–3054, 2023, doi: 10.1007/s10462-022-10246-w.

[11] C. E. Rapp, K. L. Courtney, S. T. Franz, M. M. Colavito, T. L. Aldworth, et al., “Wildfire decision support tools: Barriers, facilitators, and future directions for effectiveness,” Fire Ecology, vol. 22, Art. no. 71, 2026, doi: 10.1186/s42408-026-00484-6.

REVIEWERS LIST

B

Babić, Katarina
Begusic, Dinko

C

Čoko, Duje

G

Galinac Grbac, Tihana
Grubisic, Ani

J

Jakubec, Maroš

K

Kraus, Daniel

L

Lovrek, Ignac

M

Mikuc, Miljenko

P

Perkovic, Toni

R

Radić, Joško

S

Šarolić, Antonio

AUTHORS INDEX

B
Babić, Katarina
Begusic, Dinko

C
Cichocki, Grzegorz
Cichoń, Krzysztof
Ćatipović, Nika

D
Danilović, Nikola

G
Grgić, Krešimir

L
Leszczyński, Wojciech

M
Mazzini, Gianluca

N
Nowak, Stanisław

P
Prentki, Cyryl
Pullin, Mauro

R
Rosandic, Snjezana

S
Škec, Edita

T
Tucaković, Slavko

Z
Żelazek, Paweł