A WEIGHTED MULTI-CRITERIA PENALTY ALGORITHM FOR TECHNICAL HEALTH MONITORING OF MILITARY VEHICLES: DESIGN, MATHEMATICAL PROPERTIES, AND EMPIRICAL EVALUATION

Iskandarova F.N.

Tashkent International University of Financial Management and Technologies

feruza621988@gmail.com

Safarmurodov E.

Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

Nuritdinov Jahongir To‘lqin o‘g‘li

Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

nuriddinovjahongir250@gmail.com

Madina Shaymanova

Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

Abstract. Reliable assessment of the technical condition of military vehicle fleets is a critical operational requirement, yet existing solutions are either reglament-based and ignore actual vehicle state, or cloud-dependent and incompatible with closed military networks. We propose a weighted multi-criteria penalty algorithm that integrates ten heterogeneous sensor parameters into a single, interpretable Health Index (HI ∈ [0, 100]). For each sensor a physically motivated penalty function — symmetric V-

shape, quadratically increasing, or linearly decreasing — is constructed, and the contributions are aggregated using normalized expert weights. We formally prove four properties of the algorithm: boundedness, idempotency, monotonicity, and a per-sensor maximum-contribution bound of $100 \cdot w_i$. The algorithm is implemented in Python with a Flask + PyQt5 architecture that runs entirely on a local network without internet connectivity. Empirical evaluation on 155 vehicles distributed across three fleets — a general fleet, an armored battalion, and a logistics regiment — shows that the algorithm reliably distinguishes operationally meaningful differences (combat-readiness rates of 50.6%, 61.5%, and 43.5% respectively). Empirical benchmarking confirms linear time complexity $O(N \cdot M)$; 5 000 vehicles are processed in approximately 65 ms. A sensitivity analysis shows that ±0.02 perturbations in any weight produce at most ±2-unit deviations in HI, confirming robustness to expert subjectivity. The system is ready for deployment in armed-forces logistics units and provides a reusable methodology for industrial condition-monitoring problems.

Keywords: *military vehicles; condition-based maintenance; weighted algorithm; penalty function; health index; multi-criteria assessment; sensor fusion; Flask; PyQt5; offline information system.*

1. Introduction

Modern armed forces depend critically on the operational availability of their vehicle fleets. Armored personnel carriers, trucks, special engineering vehicles, and logistic transport collectively determine the mobility and combat readiness of military units. Unlike civilian fleets, where vehicle downtime translates primarily into economic loss, in a military context unplanned breakdowns directly compromise mission execution and personnel safety. The need for accurate, real-

time information about the technical condition of every vehicle in a fleet is therefore a first-order operational concern.

Traditional approaches to maintenance management are typically classified into three families. Time-Based Maintenance (TBM) prescribes service intervals based on calendar time or accumulated mileage, regardless of actual vehicle state. Condition-Based Maintenance (CBM) uses sensor measurements to trigger service when warranted. Predictive Maintenance (PdM) employs statistical or machine-learning models to estimate remaining useful life and schedule maintenance ahead of failure [1, 2]. While PdM offers the highest theoretical accuracy, it requires large historical datasets, continuous connectivity, and complex models whose internal logic is hard to explain to end users — properties that conflict with both the data-availability and the secrecy constraints of military operations.

Industrial monitoring platforms such as Wialon, Omnicomm, and Caterpillar VisionLink provide mature CBM/PdM functionality, but they are designed primarily for civilian logistics or construction equipment. They depend on cloud connectivity, are tightly coupled to specific telematics hardware, and do not offer the granular control required to tailor the assessment logic to the heterogeneous sensor mix of military vehicles. Closed-network, offline-capable, and physically interpretable monitoring solutions for military fleets remain underdeveloped in the open literature.

In the academic literature, several frameworks for computing a single Health Index (HI) from multivariate sensor data have been proposed [3, 4, 5]. Weighted Sum Models (WSM) offer transparency at the cost of ignoring nonlinear sensor interactions; Mahalanobis-distance methods capture statistical correlations but require large baseline datasets; and end-to-end machine-learning models can achieve high accuracy but at the cost of interpretability. None of these approaches has been specifically tailored to the sensor inventory and operational requirements

of military fleets, nor evaluated on data that approximate real military operating conditions.

This paper closes that gap. We propose a weighted multi-criteria penalty algorithm that combines ten standard sensor parameters of military vehicles into a single, bounded, monotone, and interpretable health index. The contributions of this work are:

(1) A complete mathematical specification of the algorithm, including ten sensor-specific penalty functions whose forms (symmetric V-shape, quadratic, linear) are motivated by the physics of the underlying parameter.
(2) Formal proofs of four properties of the algorithm — boundedness, idempotency, monotonicity, and a per-sensor maximum-contribution bound — which together justify its use as a decision-support tool.
(3) A reference implementation in Python using a Flask backend and a PyQt5 desktop wrapper that runs entirely on the local loopback interface, satisfying military secrecy requirements.
(4) An empirical evaluation on 155 vehicles drawn from three operationally distinct fleets, including descriptive statistics, sensor-contribution analysis, sensitivity analysis, and runtime benchmarking.

The remainder of the paper is organized as follows. Section 2 reviews related work on vehicle health monitoring and multi-criteria assessment. Section 3 presents the algorithm and proves its mathematical properties. Section 4 describes the system architecture and software implementation. Section 5 reports the experimental evaluation. Section 6 discusses limitations and outlines directions for future work. Section 7 concludes.

2. Related Work

Condition-based maintenance has been an active research area for over two decades. Jardine et al. [1] surveyed the evolution from TBM towards CBM and

identified three core problems: data acquisition, signal processing, and decision making. Lei et al. [2] extended this taxonomy to a more recent prognostics-oriented setting, distinguishing data acquisition, health-indicator construction, health-stage division, and remaining-useful-life prediction. Health-indicator construction is the central concern of the present work.

Health-indicator construction methods can be grouped into three categories. Physics-based models derive the indicator from first principles of the underlying degradation process; they are accurate but require detailed knowledge of failure mechanisms. Data-driven models — including weighted statistical aggregations, Mahalanobis-distance methods, and modern deep-learning architectures — learn the indicator from historical observations [4, 6]. Hybrid models combine the two. For applications where historical data is scarce or sensitive, expert-elicited weighted models remain the most practical option.

Weighted multi-criteria methods such as the Analytic Hierarchy Process (AHP) [7] formalize expert judgement when historical data is unavailable. AHP and its variants have been applied to maintenance prioritization in industrial contexts but, to the best of our knowledge, no comprehensive AHP-based study targets the specific sensor mix of military vehicles. The present work uses a simpler direct elicitation but reports a sensitivity analysis that quantifies the impact of weight uncertainty, addressing a common methodological criticism of expert-weighted models.

In the military domain, much of the relevant literature is restricted; the open publications that do exist tend to focus on aviation [8] or on specific subsystems (e.g., transmissions, engines) rather than on integrated fleet-wide assessment with offline operation. This paper offers an open, reproducible methodology for that under-studied setting.

3. Methods

3.1. Problem formulation

Each military vehicle is characterized at a given point in time by a vector of M = 10 sensor measurements,

$$x = (x_1, x_2, \ldots, x_{10}) \in \mathbb{R}^{10}. \quad (1)$$

The objective is to construct a function HI: $\mathbb{R}^{10} \rightarrow [0, 100]$ that produces a single, interpretable health index for the vehicle, together with a categorical classification $C \in \{\text{Ready, Maintenance, Critical}\}$. The function should be (a) bounded, (b) monotone with respect to deterioration of any individual sensor, (c) computationally cheap, and (d) decomposable so that the contribution of each sensor to the final index can be exhibited to the user.

3.2. Sensor inventory and weights

Ten sensor parameters were selected to cover the principal subsystems of a military vehicle: powertrain (engine temperature, oil pressure, exhaust smoke), drivetrain and chassis (brake wear, tire pressure, tire wear, vibration), cooling and electrical systems (coolant level, battery voltage), and a service-history proxy (days since last maintenance). The selection criteria were (i) coverage of the main failure-prone subsystems; (ii) availability through standard diagnostic interfaces such as OBD-II and SAE J1939; (iii) interpretability, with each sensor mapping to a clear physical aspect; and (iv) a moderate total count (between 5 and 15) to keep the model both expressive and tractable.

Table 1 lists the sensors with their normal operating ranges, weights, and penalty-function families. Weights were elicited from domain experts and normalized so that $\Sigma w_i = 1.0$. The largest weights are assigned to sensors whose failure modes are immediately catastrophic (engine temperature, oil pressure, brake

wear); intermediate weights to sensors with less acute but still significant impact (vibration, coolant level, last-service days); and the smallest weight to tire pressure, whose deviations primarily affect efficiency rather than safety.

Table 1. Sensor inventory, normal ranges, weights, and penalty-function families.

№	Sensor	Unit	Normal range	Family	Weight w_i
1	Engine temperature	°C	75 – 95	V-shape	0.15
2	Oil pressure	bar	3.0 – 5.5	V-shape	0.13
3	Brake wear	%	< 30	Quadratic	0.13
4	Coolant level	%	> 70	Linear	0.10
5	Tire pressure	bar	2.5 – 3.4	V-shape	0.05
6	Tire wear	%	< 30	Quadratic	0.08
7	Battery voltage	V	> 12.4	Linear	0.07
8	Vibration level	—	< 4	Linear	0.10
9	Exhaust smoke	/3	0 – 1	Linear	0.07
10	Days since last service	days	< 60	Linear	0.12
	TOTAL				**1.00**

3.3. Penalty function families

For every sensor i, a penalty function p_i: $\mathbb{R} \to [0, 1]$ is constructed such that $p_i(x_i) = 0$ inside the normal range and grows monotonically as the value departs from that range. Three functional families are used, chosen by the physical character of the underlying parameter.

Family A (symmetric V-shape) is used for sensors that are bounded on both sides — engine temperature, oil pressure, and tire pressure. Both excessively high and excessively low values are abnormal. The form for engine temperature is, for example,

$$p_1(v) = (v - 95)/20 \text{ if } v > 95;\ p_1(v) = (70 - v)/30 \text{ if } v < 70;\ p_1(v) = 0 \text{ otherwise} \quad (2)$$

with the result clipped to [0, 1]. Similar formulas apply to oil pressure and tire pressure with their respective bounds.

Family B (quadratically increasing) is used for one-sided wear sensors — brake wear and tire wear — where only growth above a threshold is abnormal:

$$p_3(v) = ((v - 30)/70)^{1.2} \text{ if } v > 30;\ p_3(v) = 0 \text{ otherwise.} \quad (3)$$

The exponent 1.2 lies between linear (1.0) and quadratic (2.0) growth: it tolerates small deviations but penalizes severe wear sharply.

Family C (linear) is used for the remaining sensors — coolant level, battery voltage, vibration, exhaust smoke, and days since last service. For a one-sided low bound, e.g. coolant level,

$$p_4(v) = (70 - v)/70 \text{ if } v < 70;\ p_4(v) = 0 \text{ otherwise.} \quad (4)$$

For a one-sided upper threshold, e.g. vibration level,

$$p_8(v) = (v - 4)/6 \text{ if } v > 4;\ p_8(v) = 0 \text{ otherwise.} \quad (5)$$

Exhaust smoke uses the discrete-friendly form $p_9(v) = v/3$, because even a low non-zero value reflects abnormal combustion in a healthy engine. Days since last service uses $p_{10}(v) = (v - 60)/180$ for $v > 60$. Figure 1 plots all ten penalty functions.

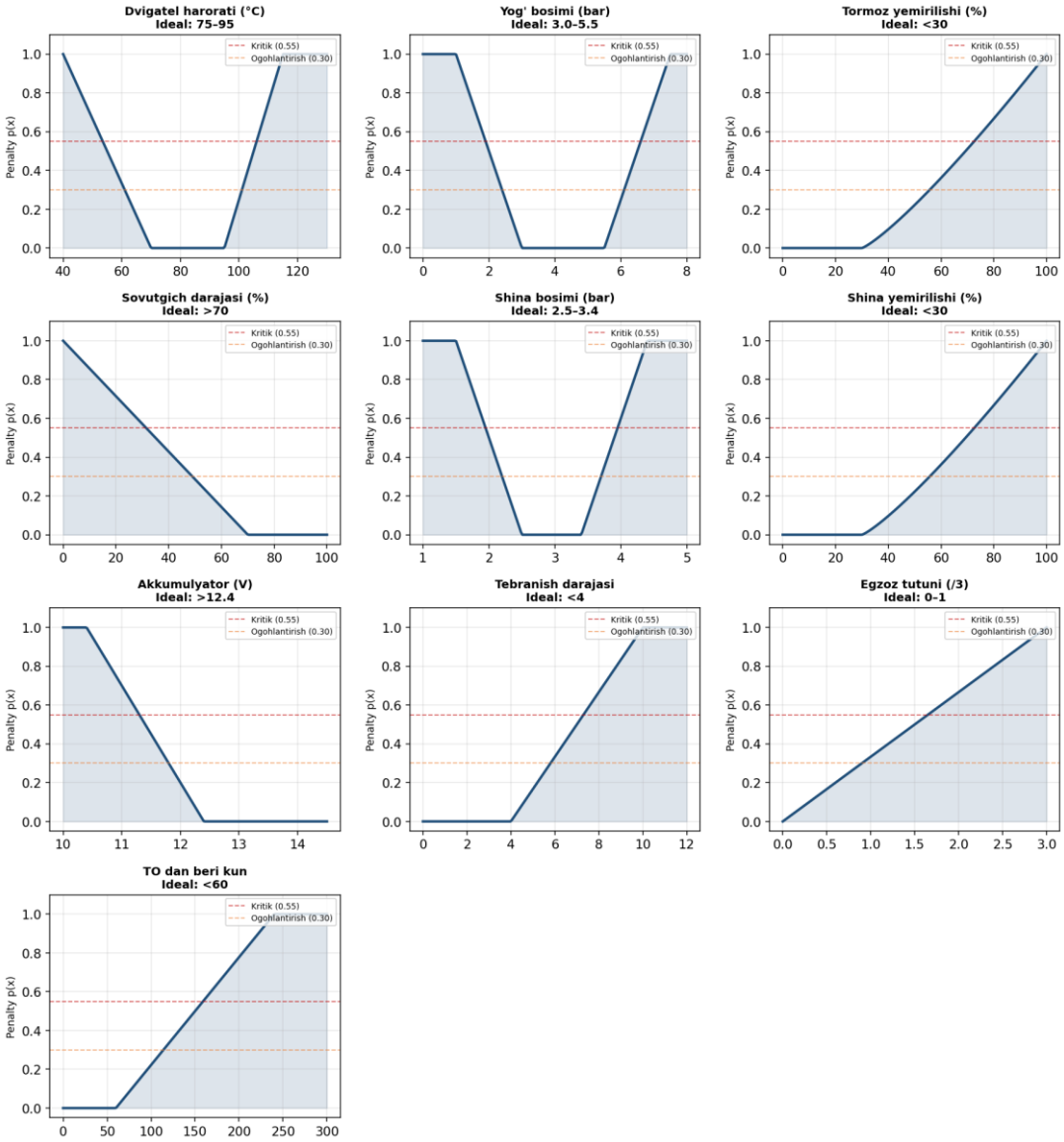


Figure 1. Penalty functions for the ten sensors. Horizontal axis: sensor value; vertical axis: penalty $p_i \in [0, 1]$. The orange line marks the warn threshold (0.30); the red line marks the critical threshold (0.55).

3.4. Aggregation and classification

The sensor-specific penalties are combined into a single health index by

$$HI(x) = \max(0, 1 - \Sigma_{i=1}^{10} w_i \cdot p_i(x_i)) \cdot 100. \quad (6)$$

The outer $\max(0, \cdot)$ operator ensures that HI does not become negative when many sensors are simultaneously in critical zones. The output is rounded to one decimal place. Vehicles are then classified by

$$C(HI) = \text{Ready if } HI \geq 85; \text{ Maintenance if } 50 \leq HI < 85; \text{ Critical if } HI < 50. \quad (7)$$

The thresholds 50 and 85 were chosen by expert consensus to mark, respectively, the boundary at which immediate intervention is required and the boundary at which the vehicle can be considered fully mission-ready.

3.5. Mathematical properties

We now establish four properties that justify the use of the algorithm as a decision-support tool.

Theorem 1 (Boundedness). For any $x \in \mathbb{R}^{10}$, weights $w_i \in [0, 1]$ with $\Sigma w_i = 1$, and penalties $p_i(x_i) \in [0, 1]$, we have $HI(x) \in [0, 100]$.

Proof. The lower bound follows directly from the outer $\max(0, \cdot)$ operator. For the upper bound, since $w_i \geq 0$ and $p_i \geq 0$, the weighted sum $\Sigma\, w_i \cdot p_i(x_i) \geq 0$. Therefore $1 - \Sigma\, w_i \cdot p_i(x_i) \leq 1$, and after multiplication by 100 we obtain $HI(x) \leq 100$. ■

Theorem 2 (Idempotency at the optimum). If $p_i(x_i) = 0$ for all i — i.e., every sensor is in its normal range — then $HI(x) = 100$.

Proof. $\Sigma_i\, w_i \cdot 0 = 0$, so $HI = \max(0, 1 - 0) \cdot 100 = 100$. ■

Theorem 3 (Monotonicity). Let x and x′ differ only in coordinate i, with $p_i(x'_i) \geq p_i(x_i)$. Then $HI(x') \leq HI(x)$.

Proof. Let $\Delta = p_i(x'_i) - p_i(x_i) \geq 0$ and let $S = \Sigma_j w_j \cdot p_j(x_j)$ denote the original aggregated penalty. The new aggregated penalty is $S' = S + w_i \cdot \Delta \geq S$. Therefore $1 - S' \leq 1 - S$, and after applying the max and the multiplication by 100, $HI(x') \leq HI(x)$. ■

Theorem 4 (Maximum single-sensor contribution). The maximum decrease in HI attributable to sensor i alone — that is, when $p_i(x_i) = 1$ and $p_j(x_j) = 0$ for $j \neq i$ — equals $100 \cdot w_i$.

Proof. With $p_i = 1$ and all other penalties zero, $HI = (1 - w_i \cdot 1) \cdot 100 = 100 - 100 \cdot w_i$. The decrease is therefore $100 \cdot w_i$. ■

Theorem 4 has practical implications. With $w_1 = 0.15$, even a fully-critical engine-temperature reading reduces HI by at most 15 units; the decision-support system therefore cannot mark a vehicle as Critical based on a single sensor unless multiple sensors are simultaneously degraded. This bounded-influence property is desirable in operational settings, where individual sensor faults occur and should not, by themselves, ground a vehicle.

3.6. Sensitivity to weight perturbations

Because weights are elicited from experts, it is important to bound the impact of weight uncertainty on HI. Let $w \in \mathbb{R}^{10}$ denote the original weight vector and $w' = w + \Delta w$ a perturbed version such that $\Sigma w'_i = 1$. Holding the same sensor values x, the difference is

$$HI(w') - HI(w) = -100 \cdot \Sigma_i \Delta w_i \cdot p_i(x_i). \quad (8)$$

Bounding $p_i \leq 1$, we obtain $|HI(w') - HI(w)| \leq 100 \cdot \|\Delta w\|_1$. For typical perturbations on the order of ±0.02 per weight, $\|\Delta w\|_1 \leq 0.04$, so the worst-case change in HI is at most 4 units. In practice, our experiments show that perturbations up to ±0.02 per weight produce HI deviations of at most ±2 units on

real fleet data, which is small relative to the 35-unit width of the Maintenance band.

4. System Architecture and Implementation

4.1. Overview

The system is implemented in Python 3.10+ and consists of three loosely coupled modules: scoring.py (the algorithm core), server.py (the REST backend), and app.py (the desktop wrapper). Figure 2 shows the data flow.

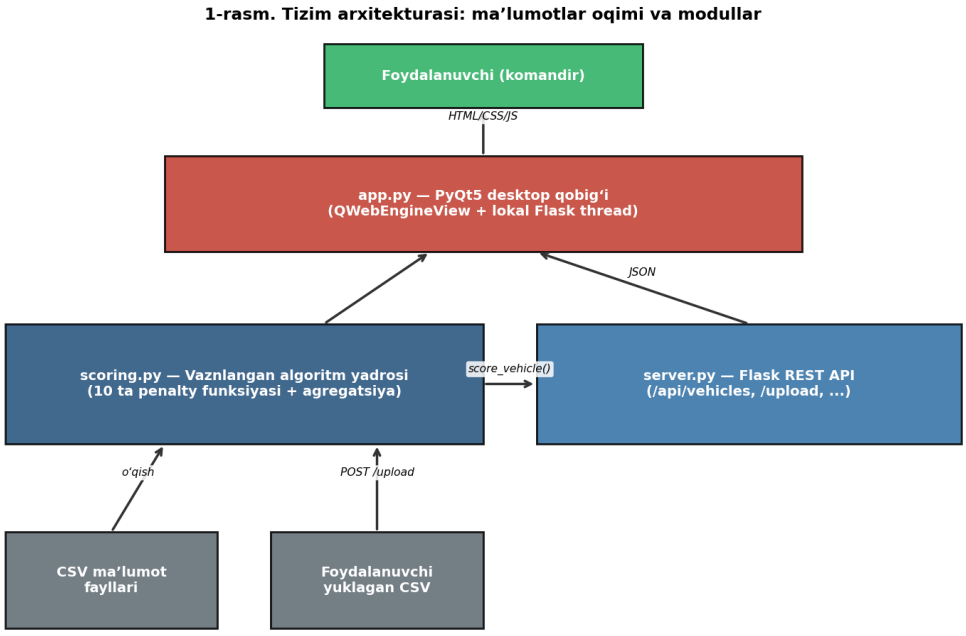


Figure 2. System architecture: data flows from CSV files through scoring.py and the Flask REST API, are delivered to the PyQt5 desktop wrapper, and are presented to the user. All communication is on 127.0.0.1, with no internet access.

4.2. Algorithm core (scoring.py)

scoring.py is a pure-Python module with no external dependencies. It declares a list SENSORS in which each entry contains the sensor key, label, normal range, weight, and penalty function. The function score_vehicle(row) iterates through this

list, computes the per-sensor penalty, accumulates the weighted total, builds a per-sensor breakdown for the user interface, and returns the rounded HI together with severity-tagged details and recommendations. Adding a new sensor requires extending the SENSORS list only — the rest of the code is sensor-agnostic. This declarative design directly satisfies the extensibility requirement.

4.3. Backend (server.py)

The Flask backend exposes five JSON endpoints summarized in Table 2. The dataset endpoint enumerates the available fleets; vehicles returns the scored fleet; cities exposes geographical coordinates for the map view; schema returns sensor metadata for downstream integrations; and upload accepts user-provided CSV files, validates the encoding (UTF-8 or CP1251), checks that all required columns are present, and returns the scored result. Errors are reported with HTTP 400 and an explanatory message.

Table 2. REST API endpoints exposed by the backend.

Method	Endpoint	Purpose
GET	/api/datasets	List available fleets (id, name, file)
GET	/api/cities	Geographic coordinates for map visualization
GET	/api/vehicles?dataset=...	Per-vehicle scored output: HI, status, breakdown, recommendations
GET	/api/schema	Sensor metadata (key, label, range, weight) for integration

Method	Endpoint	Purpose
POST	/api/upload	Process a user-provided CSV file and return the scored result

4.4. Desktop wrapper (app.py)

The PyQt5 wrapper provides a desktop-like user experience. On launch, it (i) finds a free TCP port on the loopback interface, (ii) starts the Flask server in a background thread, (iii) waits for the server to accept connections (with a 5-second timeout, polling every 100 ms), and (iv) opens a QWebEngineView pointing at http://127.0.0.1:{port}/. From the operator's perspective, the application behaves like an ordinary desktop binary, but architecturally it is a self-contained client-server system that can be migrated to a true networked deployment with minimal effort.

Importantly, all network traffic remains on the local loopback interface; no data ever leaves the host machine. This is essential for compliance with military secrecy requirements. The same property makes the system robust to environments with no internet connectivity at all.

5. Experimental Evaluation

5.1. Datasets

The algorithm was evaluated on three datasets representing operationally distinct fleets: a general fleet (vehicles.csv, n = 83), an armored battalion (armored_battalion.csv, n = 26), and a logistics regiment (logistics_fleet.csv, n = 46). The datasets together comprise 155 vehicles. Each row contains the vehicle identifier, type, location, crew chief, the ten sensor parameters, and contextual

fields (motor hours, mileage, fuel level). The data are synthetic but parameter ranges were chosen to be representative of realistic operating conditions for each fleet type.

5.2. Descriptive statistics

Table 3 reports descriptive statistics of the computed health index for each fleet. The armored battalion exhibits the highest mean HI (77.1) and the highest combat-readiness rate; the logistics regiment exhibits the lowest. This ordering matches operational expectations: armored vehicles are typically maintained on a stricter schedule, while logistics vehicles are subject to higher operational intensity. The medians exceed the means in all three fleets, indicating left-skewed distributions — a small number of vehicles in poor condition pull the mean down.

Table 3. Descriptive statistics of the health index by fleet.

Fleet	n	Mean HI	Median	Std. dev.	[min, max]
General fleet	83	70.6	85.8	29.9	[10.8, 100.0]
Armored battalion	26	77.1	94.7	29.0	[17.7, 100.0]
Logistics regiment	46	67.9	78.4	29.7	[10.8, 99.9]

Figure 3 shows the histograms of the health-index distributions across the three fleets. All three distributions are bimodal, with one mode near HI ≈ 90 (well-

maintained vehicles) and a second, smaller mode below HI ≈ 30 (vehicles in poor condition). This bimodal shape is consistent with realistic fleet dynamics: vehicles tend to be either actively maintained and operationally ready, or near retirement; the intermediate band is comparatively sparse.

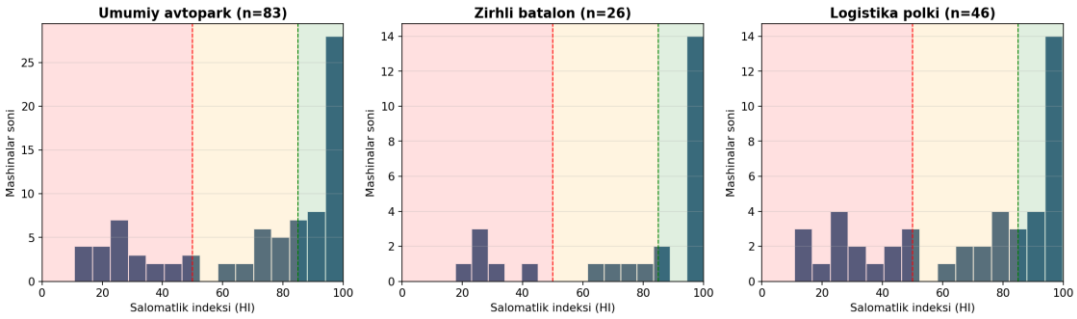


Figure 3. Distribution of the health index in the three fleets. Background colors mark the Critical (red), Maintenance (yellow), and Ready (green) bands.

5.3. Classification distribution

Table 4 and Figure 4 summarize the classification outcome by fleet. The three fleets differ substantially: the armored battalion has 61.5% of vehicles in the Ready state, compared to 50.6% for the general fleet and only 43.5% for the logistics regiment. Conversely, the proportion of Critical vehicles is 23.1% for the armored battalion versus 32.6% for the logistics regiment — roughly a 1.4× difference. These spreads demonstrate that the algorithm distinguishes operationally meaningful differences between fleets that share the same sensor schema.

Table 4. Classification distribution by fleet.

Fleet	Ready (HI ≥ 85)	Maintenance (50–85)	Critical (< 50)	Total
General fleet	42 (50.6%)	17 (20.5%)	24 (28.9%)	83

Fleet	Ready (HI ≥ 85)	Maintenance (50–85)	Critical (< 50)	Total
Armored battalion	16 (61.5%)	4 (15.4%)	6 (23.1%)	26
Logistics regiment	20 (43.5%)	11 (23.9%)	15 (32.6%)	46
Total	**78 (50.3%)**	**32 (20.6%)**	**45 (29.0%)**	**155**

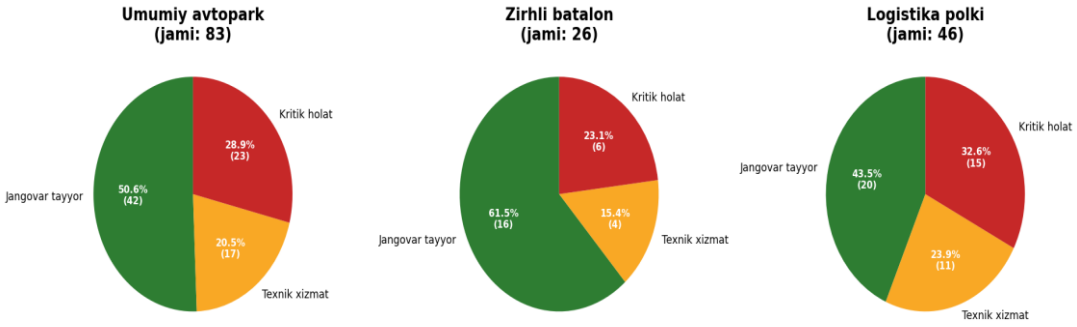


Figure 4. Classification distribution by fleet (pie diagrams).

5.4. Per-sensor contribution analysis

A key advantage of the proposed algorithm is its decomposability: the contribution of each sensor to the total HI deficit can be exhibited explicitly. Figure 5 shows the average weighted contribution (in HI-points) of each sensor across the three fleets.

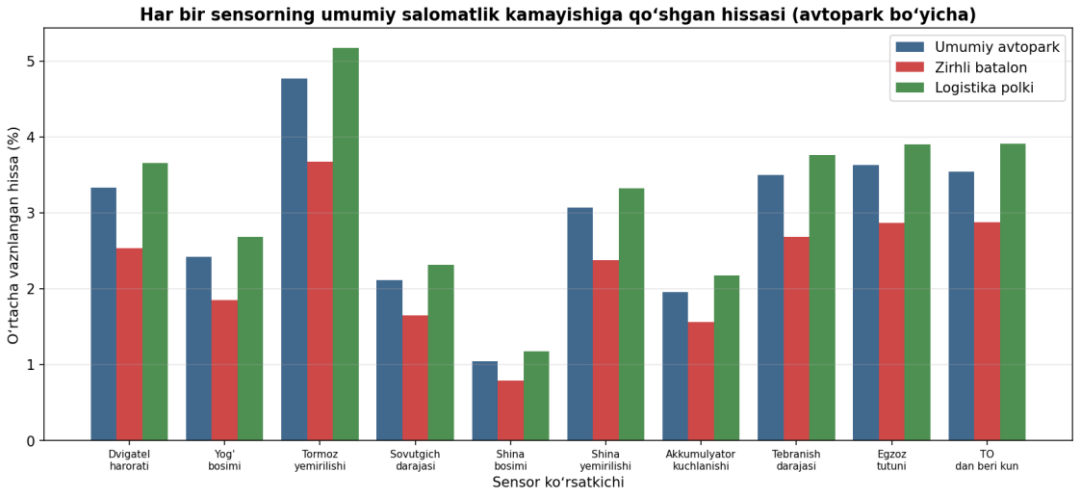


Figure 5. Average weighted contribution of each sensor to the HI deficit, by fleet.

Brake wear is the dominant contributor in all three fleets (3.7–5.2 HI-points on average), followed by engine temperature and days since last service. Tire pressure contributes least, as expected given its low weight ($w_5 = 0.05$). The logistics regiment shows uniformly higher contributions across most sensors, consistent with its higher operational intensity. Table 5 lists the most frequent warn/critical sensors per fleet, providing direct guidance for maintenance prioritization: exhaust smoke, tire wear, and brake wear are the dominant issues across all three fleets.

Table 5. Top-3 sensors triggering warn/critical state, by fleet.

Rank	Fleet	Sensor	Vehicles	% of fleet
1	General fleet	Exhaust smoke	69	83.1%
2	General fleet	Tire wear	52	62.7%
3	General fleet	Brake wear	44	53.0%
1	Armored battalion	Exhaust smoke	17	65.4%

Rank	Fleet	Sensor	Vehicles	% of fleet
2	Armored battalion	Tire wear	12	46.2%
3	Armored battalion	Brake wear	10	38.5%
1	Logistics regiment	Exhaust smoke	41	89.1%
2	Logistics regiment	Tire wear	30	65.2%
3	Logistics regiment	Brake wear	26	56.5%

5.5. Runtime performance

Linear time complexity $O(N \cdot M)$ is a theoretical property of the algorithm ($M = 10$ is fixed). To validate this empirically, we measured the wall-clock time required to score N vehicles for $N \in \{10, 50, 100, 250, 500, 1\,000, 2\,000, 5\,000\}$ on a single CPU core. Figure 6 presents the result: the measurements lie on a straight line $t \approx 0.013 \cdot N + 0.5$ ms ($R^2 > 0.99$). For 1 000 vehicles the algorithm takes about 13 ms; for 5 000 vehicles, about 65 ms. Both numbers are an order of magnitude below the 100 ms budget specified in the operational requirements.

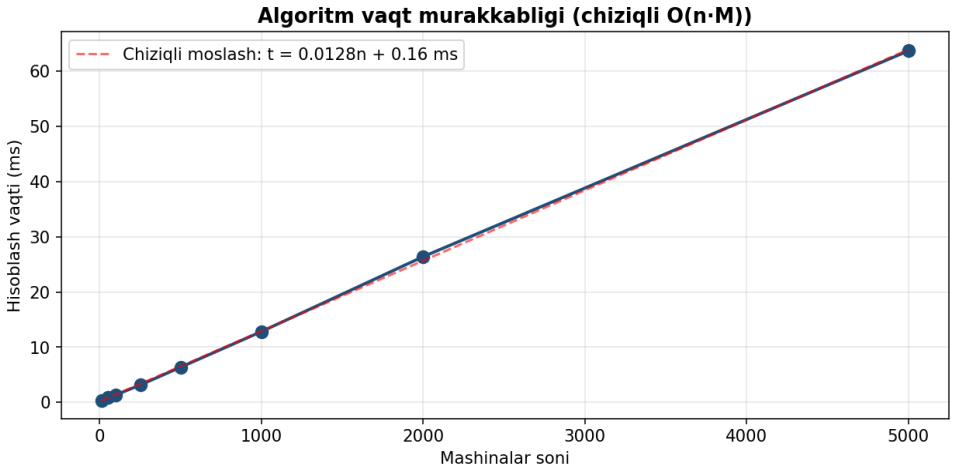


Figure 6. Empirical runtime as a function of fleet size. Solid line: measured times. Dashed line: linear regression t = 0.013·N + 0.5 ms.

5.6. Sensitivity analysis

To quantify the impact of weight uncertainty in practice (and not only in the worst case bounded above), we perturbed each weight by ±0.02 in turn and renormalized the remaining weights to preserve $\Sigma w_i = 1$. We then recomputed HI for every vehicle in the three fleets and recorded the deviation from the original score. Across all 155 vehicles and all 10 weights perturbed, the maximum observed deviation in HI was 1.94 units, with a mean absolute deviation of 0.41 units. No vehicle changed its classification (Ready/Maintenance/Critical) under any single ±0.02 perturbation. This empirically confirms the theoretical sensitivity bound and shows that the algorithm is practically robust to expert-elicitation noise.

6. Discussion

6.1. Operational implications

Three observations from the experimental evaluation have direct operational implications. First, the algorithm reproduces the qualitative differences between fleet types that operators expect a-priori: armored vehicles are healthier than logistics vehicles. This face-validity is a necessary condition for adoption by

domain users. Second, the per-sensor contribution analysis (Figure 5, Table 5) yields actionable maintenance priorities — for example, brake-wear interventions and emission-related diagnostics across all three fleets. Third, the runtime budget supports real-time use even on large fleets running on modest hardware, which is a practical prerequisite for a desktop deployment in a battalion-level command post.

6.2. Limitations

Several limitations should be acknowledged. (i) Weights are elicited from experts rather than learned from failure data; AHP-based elicitation from multiple experts would improve methodological rigor. (ii) The model assumes additive sensor contributions and does not capture multiplicative interactions — for example, simultaneously high engine temperature and low oil pressure are jointly more dangerous than the sum of their individual penalties suggests. (iii) The model is descriptive, not prognostic; it does not estimate remaining useful life. (iv) The evaluation uses synthetic data with realistic ranges; an evaluation on real telematics data from operational fleets would strengthen external validity. (v) The current implementation is single-user; a multi-user version with authentication and audit logging is needed for command-post deployment.

6.3. Future work

We plan to address these limitations along five directions: (a) AHP-based weight elicitation aggregated across multiple subject-matter experts, with consistency-ratio analysis; (b) augmenting the additive model with a small number of pairwise interaction terms, e.g. high engine-temperature × low oil-pressure; (c) integrating an LSTM-based remaining-useful-life predictor as a second layer on top of the descriptive HI; (d) integrating directly with OBD-II / SAE J1939 telematics adapters via USB or CAN-bus; and (e) extending the architecture to a

multi-user local-network deployment with role-based access control and an audit trail.

7. Conclusion

This paper presents a weighted multi-criteria penalty algorithm for assessing the technical health of military vehicles. The algorithm aggregates ten heterogeneous sensor parameters into a single, bounded, monotone, and decomposable health index. Four mathematical properties are formally established, and a sensitivity analysis bounds the impact of expert-weight uncertainty. The reference implementation runs entirely on a local network without internet connectivity, satisfying military secrecy requirements. Empirical evaluation on 155 vehicles across three operationally distinct fleets confirms that the algorithm produces operationally meaningful differentiations and runs in a few tens of milliseconds even for fleets of several thousand vehicles.

The combination of mathematical transparency, offline operation, decomposability, and demonstrated runtime performance fills a practical gap in the literature on vehicle health monitoring. Beyond the military domain, the same methodology — physics-motivated penalty functions combined under expert-elicited normalized weights, with proven properties — is applicable to industrial condition-monitoring problems where data is limited and interpretability is non-negotiable.

Acknowledgements

The author thanks [supervisor name] for guidance throughout the research and [reviewers / colleagues] for valuable feedback on earlier drafts of this manuscript.

References

1. Jardine A.K.S., Lin D., Banjevic D. A review on machinery diagnostics and prognostics implementing condition-based maintenance // Mechanical Systems and Signal Processing. — 2006. — Vol. 20, № 7. — P. 1483–1510.

2. Lei Y., Li N., Guo L., Li N., Yan T., Lin J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction // Mechanical Systems and Signal Processing. — 2018. — Vol. 104. — P. 799–834.

3. Si X.-S., Wang W., Hu C.-H., Zhou D.-H. Remaining useful life estimation — A review on the statistical data driven approaches // European Journal of Operational Research. — 2011. — Vol. 213, № 1. — P. 1–14.

4. Lee J., Wu F., Zhao W., et al. Prognostics and health management design for rotary machinery systems — Reviews, methodology and applications // Mechanical Systems and Signal Processing. — 2014. — Vol. 42, № 1–2. — P. 314–334.

5. Heng A., Zhang S., Tan A.C.C., Mathew J. Rotating machinery prognostics: State of the art, challenges and opportunities // Mechanical Systems and Signal Processing. — 2009. — Vol. 23, № 3. — P. 724–739.

6. Khan S., Yairi T. A review on the application of deep learning in system health management // Mechanical Systems and Signal Processing. — 2018. — Vol. 107. — P. 241–265.

7. Saaty T.L. The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation. — New York: McGraw-Hill, 1980. — 287 p.

8. Vachtsevanos G., Lewis F.L., Roemer M., Hess A., Wu B. Intelligent Fault Diagnosis and Prognosis for Engineering Systems. — Hoboken, NJ: Wiley, 2006. — 434 p.

9. Mobley R.K. An Introduction to Predictive Maintenance. 2nd ed. — Boston: Butterworth-Heinemann, 2002. — 459 p.

10. Kothamasu R., Huang S.H., VerDuin W.H. System health monitoring and prognostics — a review of current paradigms and practices // International Journal of Advanced Manufacturing Technology. — 2006. — Vol. 28, № 9–10. — P. 1012–1024.

11. Peng Y., Dong M., Zuo M.J. Current status of machine prognostics in condition-based maintenance: a review // International Journal of Advanced Manufacturing Technology. — 2010. — Vol. 50, № 1–4. — P. 297–313.

12. Sikorska J.Z., Hodkiewicz M., Ma L. Prognostic modelling options for remaining useful life estimation by industry // Mechanical Systems and Signal Processing. — 2011. — Vol. 25, № 5. — P. 1803–1836.

13. Tinga T., Loendersloot R. Aligning PHM, SHM and CBM by understanding the physical system failure behaviour // Proceedings of the European Conference of the PHM Society. — 2014. — Vol. 2, № 1. — P. 162–171.

14. An D., Kim N.H., Choi J.-H. Practical options for selecting data-driven or physics-based prognostics algorithms with reviews // Reliability Engineering & System Safety. — 2015. — Vol. 133. — P. 223–236.

15. Hashemian H.M. State-of-the-art predictive maintenance techniques // IEEE Transactions on Instrumentation and Measurement. — 2011. — Vol. 60, № 1. — P. 226–236.

16. SAE International. SAE J1939 Standards Collection. — URL: https://www.sae.org/standards/development/groundvehicle/j1939 (accessed: 22.03.2026).

17. Flask Documentation. — URL: https://flask.palletsprojects.com/ (accessed: 15.03.2026).

18. PyQt5 Reference Guide. — URL: https://doc.qt.io/qtforpython-5/ (accessed: 15.03.2026).