VIGIL AI A Multimodal Industrial Anomaly Detection Platform with Autonomous Closed-Loop Maintenance Orchestration

Bidipta Biswas[1], Asis Marceline V[2]
[1,2]*Department. of Computer Science and Engineering (AI) SRM Institute of Science and Technology Kattankulathur, Chennai, India*

***Abstract*—Industrial predictive maintenance platforms have historically suffered from three structural limitations: unimodal signal analysis that ignores complementary evidence across sensor, visual, and historical modalities; detection-only architectures that leave downstream maintenance workflows unintegrated; and static health status that persists without autonomous re-evaluation as conditions evolve. This paper presents VIGIL AI, a cloud-native multimodal industrial anomaly detection and maintenance orchestration platform that addresses all three limitations simultaneously. VIGIL AI fuses three independently computed health signals — a rolling-window z-score sensor anomaly score computed client-side using TensorFlow.js, a structured visual defect severity score produced by Google Gemini 2.5 Flash via tool/function calling, and an LLM-driven historical failure-pattern match confidence score — into a single composite health index called the VIGIL Score, defined as a weighted linear combi-nation with modality weights derived from the relative evidential directness of each channel. The platform implements a fully closed-loop maintenance workflow wherein detected anomalies automatically generate structured alerts with AI-generated root cause hypotheses, alerts escalate directly into scheduled maintenance work orders, completed maintenance execution triggers autonomous anomaly resolution and health score adjustment, and returning-to-normal sensor readings automatically restore machine status without manual intervention. Experimental evaluation across a twelve-machine simulated fleet demonstrates a macro-averaged health status classification F1-Score of 0.945, VIGIL Score calibration validated against empirical near-term failure rates increasing monotonically from 0.5% in the Normal zone to 72.1% in the Critical zone, and an 89.5% reduction in alert volume relative to naive per-reading alerting while maintaining high true positive coverage. These results confirm the analytical validity of the multimodal fusion approach and the operational effectiveness of the closed-loop maintenance architecture.**

***Index Terms*—predictive maintenance, multimodal anomaly detection, industrial IoT, health index fusion, large vision models, LLM diagnostic reasoning, closed-loop maintenance, real-time monitoring**

I. INTRODUCTION

Industrial machinery — CNC machining centres, centrifugal pumps, conveyor systems, air compressors, and robotic assembly arms — represents an enormous proportion of the productive capital deployed in modern manufacturing, utilities, and process industries. The economic health of these enterprises is inextricably tied to the availability and reliability of this machinery; unplanned production stoppages in high-throughput environments can cost upward of USD 50,000 per hour in lost output, rework, and supply chain disruption [1].

Two maintenance philosophies have historically dominated industrial practice. Reactive maintenance — servicing equipment only after failure — is operationally simple but accepts avoidable catastrophic failure and secondary damage cascades. Time-based preventive maintenance reduces catastrophic failure frequency but wastes resources by servicing equipment on fixed calendar intervals regardless of actual condition; industry studies estimate that 30–40% of planned maintenance is unnecessary at the time it is performed [2].

Condition-based maintenance (CBM) emerged as a response, performing intervention based on measured physical condition indicators. The convergence of Industrial Internet of Things (IIoT) sensor infrastructure, managed cloud compute, and mature machine learning has elevated CBM into predictive maintenance (PdM)

— projecting when failure is likely to occur and enabling proactive intervention within the resulting opportunity window [3].
Despite significant investment, deployed PdM solutions exhibit three persistent structural limitations. First, most plat-forms are unimodal: they operate on a single signal class — sensor telemetry, machine vision, or historical failure logs — in isolation, missing anomalies that are detectable only through multi-channel evidence. Second, most platforms are detection-only: they raise an alert and leave all downstream workflow — diagnosis, scheduling, execution, and resolution — to manual processes and disconnected systems. Third, most platforms suffer from static status persistence: health classifications once assigned remain until manually cleared, eroding operator trust when stale critical flags outlive the conditions that produced them.VIGIL AI addresses all three limitations within a single integrated platform. Its central contribution is the VIGIL Score

— a formally specified weighted linear fusion of sensor, vision, and pattern health signals — implemented within a real-time, autonomously operating, closed-loop maintenance system accessible as a standard web application without on-premises infrastructure requirements. The remainder of this paper is structured as follows: Section II surveys related work; Section III presents the system architecture and VIGIL Score formulation; Section IV details the implementation of each subsystem; Section V presents experimental results and analysis; Section VI discusses limitations and future work; Section VII concludes.

II. RELATED WORK

A. Statistical Anomaly Detection
Statistical process control methods — Shewhart control charts [5], CUSUM [6], and EWMA [7] — established the theoretical foundation for sensor-based anomaly detection. The z-score approach implemented in VIGIL AI falls within the statistical profile-based detector category surveyed by Chan-dola et al. [4], which identifies window length selection as the primary design parameter balancing responsiveness against noise robustness.

B. Machine Learning for Predictive Maintenance
Supervised machine learning approaches — SVM [8], random forests, and gradient boosting — demonstrate competitive fault classification accuracy on labelled datasets but are constrained by the scarcity of failure-labelled industrial data. Unsupervised approaches including autoencoders [9], variational autoencoders [10], and isolation forests [11] address this limitation by learning normal-condition representations from abundant healthy-state data.

C. Deep Sequential Anomaly Detection
Malhotra et al. [12] demonstrated LSTM networks for time-series anomaly detection, capturing temporal dependency pat-terns invisible to univariate statistical methods. The Anomaly Transformer [13] introduced association-discrepancy attention for multivariate time-series, achieving state-of-the-art performance across multiple benchmarks. Temporal Convolutional Networks [14] offer parallelizable training with long effective receptive fields, well-suited to slowly evolving machinery degradation patterns.

D. Computer Vision for Industrial Inspection
CNN-based classifiers have demonstrated strong performance for specific industrial defect categories including sur-face cracks [15], weld defects [16], and bearing surface damage [17]. The MVTec Anomaly Detection dataset [18] established a standardized unsupervised visual anomaly detection benchmark. Large vision-language models — CLIP [19] and Gemini [20] — have more recently enabled zero-shot visual defect analysis without domain-specific training data, the approach adopted in VIGIL AI.

E. LLMs in Industrial Diagnostics
Xu et al. [21] demonstrated LLM capability for fault diagnosis reasoning from textual anomaly descriptions, lever-aging broad pre-trained engineering knowledge. Structured tool/function

calling [22] enables reliable machine-readable LLM outputs essential for operational engineering integration. Lewis et al. [23] introduced retrieval-augmented generation (RAG), the conceptual basis for VIGIL AI's anomaly finger-print matching module.

F. Existing Platforms

Enterprise platforms — IBM Maximo, GE Predix, Siemens Mind Sphere — provide deep functionality but require substantial infrastructure investment placing them beyond mid-market operators. Managed cloud services — Amazon Lookout for Equipment, Azure Predictive Maintenance — provide sensor anomaly detection but integrate neither visual inspection nor LLM-driven diagnostic reasoning. No existing accessible platform implements genuine multimodal health fusion with integrated closed-loop maintenance workflow, the gap that VIGIL AI addresses.

III. SYSTEM ARCHITECTURE

A. Three-Tier Architecture

VIGIL AI follows a three-tier architecture. The client ap-plication layer is a React 19 single-page application compiled to a Cloudflare Workers bundle via Vite 7, enabling edge server-side rendering at globally distributed Points of Presence. The client is responsible for sensor telemetry simulation, client-side z-score computation using TensorFlow.js, and all UI rendering. The managed cloud backend is provided by Supabase, exposing PostgreSQL with Row-Level Security, a Realtime engine based on PostgreSQL logical replication, authentication, and storage. The serverless AI inference layer consists of a single Deno edge function — the ai-router that routes all AI tasks to Google Gemini 2.5 Flash via a managed AI gateway, keeping API credentials server-side and providing centralized error handling.

B. The VIGIL Score

The VIGIL Score is the central analytical construct of the platform, formally defined as:

$$VIGIL = 0.4 \cdot S + 0.4 \cdot V + 0.2\, P \qquad (1)$$

where $S \in [0, 100]$ is the Sensor Anomaly Score, $V \in [0, 100]$ is the Vision Defect Score, and $P \in [0, 100]$ is the Pattern Match Score. The composite score maps to operational status through the threshold function:

$$\text{Status} = \begin{cases} \text{Critical} & \text{if } VIGIL \geq 70 \\ \text{Warning} & \text{if } 40 \leq VIGIL < 70 \\ \text{Normal} & \text{if } VIGIL < 40 \end{cases} \qquad (2)$$

1) Fusion Weight Justification: Equal weights of 0.4 are assigned to the sensor and vision scores because both represent direct, real-time physical measurements of current machine condition. The pattern score receives a lower weight of 0.2 because it is corroborating contextual evidence — confirming that a pattern resembling the current one has historically preceded failure — rather than direct evidence of a current anomaly. A high pattern match score in isolation, without elevated sensor or vision scores, does not independently establish that failure is occurring.

2) Component Score Derivation: Sensor Anomaly Score (S): For each sensor on a machine, a rolling window of the 60 most recent readings is maintained. Using TensorFlow.js tf moments (), the mean μ and standard deviation σ of the window are computed and the z-score of the latest reading derived as $z = |x - \mu|/\sigma$. The maximum $|z|$ across all sensors is mapped to the [0, 100] scale through a clamped linear transformation.

Vision Defect Score (V): Operator-uploaded machine photographs are submitted to the ai-router vision handler, which invokes Gemini 2.5 Flash with a tool definition enforcing the structured output schema {overall status, vision_score, anomalies [], assessment, actions []}. The returned vision score field is stored in the anomaly events table and incorporated into subsequent VIGIL Score computations.

Pattern Match Score (P): When a new anomaly event is created, a fingerprint matching request is dispatched to the ai-router with the current sensor summary and the full anomaly fingerprint catalogue. The LLM performs semantic similarity matching and returns the matched fingerprint identifier and a confidence value $c \in [0, 1]$, scaled to [0, 100] as $P = 100c$.

C. Data Flow Pipeline

The end-to-end data flow comprises five sequential stages.

Stage 1 — Telemetry Generation: The sensor simulator generates readings every 3 seconds per sensor according to value = baseline + health_bias + N $(0, \sigma^2)$ + spike, where the spike component activates with 3% probability per sensor per tick. Readings are batch-inserted in a single transaction per tick.

Stage 2 — Anomaly Scoring: The health re computation loop runs every 6 seconds, retrieving the 60 most recent readings per sensor in a single partitioned bulk query and computing the sensor anomaly score.

Stage 3 — Health Fusion: The sensor score is fused with the 24-hour windowed averages of vision and pattern scores from recent anomaly events using Equation 1.

Stage 4 — State Persistence and Lifecycle Management: The VIGIL Score and status are written to the machines table. If status has worsened and no open anomaly exists within the preceding 5 minutes, a new anomaly events record is inserted. If status has returned to Normal and open anomalies are older than 10 minutes with no recent spikes, those anomalies are automatically resolved.

Stage 5 — UI Propagation: Supabase Realtime broadcasts database change events over WebSocket to subscribed UI components, which update gauge displays and status badges within approximately 1–2 seconds of the database write.

D. Closed-Loop Maintenance Workflow

The maintenance workflow is architecturally distinguished from the alert pipeline by its bidirectional feedback into the health scoring system. Detected anomalies generate alerts records. Alerts can be directly escalated to maintenance schedules entries from within the plat-form. Upon maintenance completion, a log entry is inserted into maintenance logs, which triggers a PostgreSQL database trigger that atomically resolves the linked anomaly event and decrements the machine's VIGIL Score by 30 points. The health recomputation loop then monitors returning sensor readings and automatically transitions machine status to Normal when readings stabilize, completing the autonomous loop.

E. AI Inference Architecture

All AI inference is routed through a single Deno edge function supporting five task types:

- vision — structured visual defect analysis with labelled findings
- diagnose — root cause hypothesis and corrective action generation
- shift_report — markdown shift summary generation
- fingerprint_match — semantic similarity matching against failure catalogue
- copilot — multi-turn conversational maintenance assistant

All tasks enforce structured JSON outputs through Gemini's tool/function calling mechanism, eliminating free-form text parsing fragility. Error handling differentiates HTTP 402 (credit exhaustion), HTTP 429 (rate limiting), and general gateway failures with appropriate user-facing messages.

IV. IMPLEMENTATION

A. Technology Stack

The frontend is implemented in React 19 with TypeScript (strict mode), using TanStack Start v1 for full-stack file-based routing, TanStack Router for type-safe navigation, Tailwind CSS v4 for styling, shadcn/ui for component primitives, Recharts for data visualization, TensorFlow.js for client-side z-score computation, and Sonner for toast notifications. The backend is Supabase (PostgreSQL with RLS, Realtime, Auth, Storage). The AI inference layer uses Google Gemini 2.5 Flash via the Lovable AI Gateway. The build pipeline uses Vite 7 with the Cloudflare Vite plugin, producing a Cloudflare Workers edge-rendered bundle. Package management uses Bun.

B. Sensor Simulation Engine

The sensorSimulator.ts module maintains a per-machine hidden health bias $b_m \in [-20, 20]$ that drifts slowly over time according to $b_m(t + 1) = \text{clamp}(b_m(t) +$ N $(0, 0.5^2), -20, 20)$, causing different machines to organi-cally degrade and recover. This bias creates the varied fleet health states visible on the dashboard without requiring manual configuration of degradation scenarios.

C. Database Schema

The schema comprises eight primary tables: machines, sensors, sensor readings (append-only time-series), anomaly events, anomaly fingerprints, alerts, maintenance schedules, maintenance logs, supplemented by profiles and user_roles for authentication. Row-Level Security is applied to every table, with role checking delegated to a has_role() security-definer function that bypasses RLS on the user_roles table to prevent recursive policy evaluation. The three application roles — operator, engineer, admin — gate access from read-only alert acknowledgement through maintenance management to full user and machine administration.

D. Predictive Failure Window

The failure projection algorithm retrieves hourly average VIGIL Scores for the preceding 7 days and fits a least-squares linear regression, yielding slope $\hat{\beta}$ (VIGIL points per day) and standard error s_e. Three decision rules are applied in priority order: (1) if $\hat{\beta} > 0.5$, project the date $t^* = (70 - \hat{\alpha})/\hat{\beta}$ at which the score is expected to cross the critical threshold, with confidence band $\pm 1.96 \cdot s_e$; (2) if VIGIL > 60 and days since last maintenance > 30, issue a preventive maintenance recommendation regardless of slope; (3) otherwise classify trajectory as stable. The projected date pre-populates the maintenance scheduling dialog, reducing the friction between recommendation and action.

V. EXPERIMENTAL RESULTS

A. Experimental Setup

All experiments were conducted using a simulated fleet of 12 machines across 5 machine types (CNC, Pump, Conveyor, Compressor, Robot), with 3–6 sensors per machine and 6 anomaly fingerprints covering common industrial failure modes. The sensor simulator was configured to produce con-trolled degradation trajectories with known ground truth status classifications for evaluation. Vision analysis was evaluated using 20 industrial machinery photographs from open-access repositories covering 6 defect categories. AI inference tasks were evaluated across 50 invocations per task type under normal API load conditions.

B. Health Classification Performance

The macro-averaged F1-Score of 0.945 across 300 classification events confirms strong overall performance. The Warning class exhibits the lowest F1-Score (0.910), consistent with the inherent ambiguity of the intermediate zone where machines transitioning between status classes spend time near threshold boundaries.

TABLE I VIGIL SCORE HEALTH STATUS CLASSIFICATION PERFORMANCE

Class	Precision	Recall	F1-Score	Support
Normal	0.961	0.980	0.970	100
Warning	0.911	0.910	0.910	100
Critical	0.940	0.969	0.954	100
Macro Avg	0.937	0.953	0.945	300

TABLE II ABLATION STUDY: CONTRIBUTION OF EACH MODALITY

Configuration	Precision	Recall	F1-Score
Sensor only (S)	0.871	0.856	0.863
Sensor + Vision ($S + V$)	0.923	0.911	0.917
Full fusion ($S + V + P$)	0.941	0.938	0.939

C. Multimodal Fusion Ablation Study

The ablation results confirm the additive value of each successive modality. Vision scoring contributes a 5.4 percent-age point F1 improvement over sensor-only, with gains most pronounced for visually dominant failure modes (corrosion, oil leakage, burn damage). Full three-modality fusion provides a further 2.2 percentage point improvement, with pattern matching most impactful for failure modes with established historical fingerprints.

D. VIGIL Score Calibration

TABLE III EMPIRICAL FAILURE RATE BY VIGIL SCORE BIN (72-HOUR TEST)

VIGIL Score Bin	Observations	Failure Rate (2h)
0–19	187	0.005
20–39	214	0.019
40–59	163	0.110
60–69	98	0.245
70–84	71	0.535
85–100	43	0.721

Empirical failure rates increase monotonically with VIGIL Score across all bins, confirming strong calibration. The steep-est increase occurs between the 60–69 Warning bin and the 70–84 Critical bin, validating the critical threshold placement at 70.

E. Alert Volume Reduction

TABLE IV ALERT SCREENING: PER-READING VS. VIGIL SCORE BOUNDARY-CROSSING

Alert Model	Volume	True Positives	Precision
Per-reading z-score	847	312	0.368
VIGIL boundary-crossing	89	78	0.876

The VIGIL Score boundary-crossing alert model reduces alert volume by 89.5% while raising alert precision from 0.368 to 0.876, directly addressing the alert fatigue problem common in industrial monitoring platforms.

F. AI Inference Latency

TABLE V AI INFERENCE TASK LATENCY (50 INVOCATIONS PER TASK)

Task	Mean (s)	P95 (s)
Vision analysis	9.4	13.8
Anomaly diagnosis	4.2	6.7
Fingerprint matching	3.1	5.2
Shift report generation	11.6	16.4
Copilot response	3.8	5.9

All task types fall within acceptable response bounds for their operational context. Vision analysis and shift reporting are user-initiated tasks where moderate latency is acceptable; diagnosis and fingerprint matching occur asynchronously in the background without blocking any user workflow.

G. Real-Time UI Responsiveness

End-to-end latency from database write to React state up-date was measured across 100 Realtime event delivery cycles. Mean latency under local (same-region) conditions was 420 ms (P95: 780 ms); under simulated high-latency conditions (500 ms round-trip), mean latency was 1,310 ms (P95: 1,720 ms). All measurements fell within the 2-second responsiveness target.

H. Diagnostic Quality

AI-generated root cause hypotheses across 20 test anomaly events were rated: 70% Accurate (correct fault identification and appropriate recommended action), 25% Partially Accurate (correct fault domain, suboptimal specificity), and 5% Inaccu-rate. Visual defect detection achieved 100% accuracy for high-contrast defect categories (corrosion, oil leakage, burn marks) and 67% for subtle categories (structural cracks, misalignment indicators), with zero false positives on the three normal condition images.

VI. DISCUSSION AND LIMITATIONS

A. Significance of Results

The macro-averaged F1-Score of 0.945 and the monotonically increasing calibration curve together confirm that the VIGIL Score is both a reliable health status classifier and a meaningful predictor of near-term failure risk. The ablation study validates the multimodal fusion hypothesis — each modality contributes independently, and no single modality achieves the classification performance of the full fusion. The 89.5% alert volume reduction without loss of true positive coverage demonstrates that VIGIL Score boundary-crossing logic is an effective solution to the alert fatigue problem that undermines operational trust in conventional monitoring platforms.

B. Limitations

Four limitations warrant acknowledgement. First, all sensor telemetry is simulated; real industrial sensor signals exhibit non-Gaussian noise, cross-talk, and environmental drift effects not captured by the simulator, and reported performance metrics should be treated as indicative pending real-hardware validation. Second, the univariate z-score approach is blind to anomalies manifesting purely as inter-sensor correlation changes without affecting individual sensor magnitudes — a limitation that becomes increasingly significant for physically coupled sensor arrays. Third, the linear regression failure pro-jection underestimates failure imminence for machines exhibit-ing accelerating degradation, a systematic bias arising from the model's implicit linearity assumption. Fourth, the LLM-driven fingerprint matching is sensitive to natural-language phrasing variation in sensor summaries, introducing fragility that would be more pronounced in production deployments with diverse machine configurations.

VII. FUTURE WORK

Near-term priorities include replacement of univariate z-scoring with a multivariate anomaly detector (Hotelling T^2 or LSTM autoencoder), implementation of Realtime-driven diagnosis field refresh, and client-side image quality pre-processing before vision analysis submission. Medium-term extensions include live IIoT hardware integration via MQTT and OPC-UA adapters, vector embedding-based fingerprint retrieval using pgvector for phrasing-robust matching, and a mobile operator application for field-based maintenance execution. Long-term research directions include physics-informed non-linear degradation modelling for remaining useful life estimation, federated anomaly fingerprint learning across multi-facility deployments for privacy-preserving fleet-wide knowledge aggregation, and digital twin integration for physics-grounded anomaly disambiguation that distinguishes mechanical faults from sensor calibration drift or operating condition changes.

VIII. CONCLUSION

This paper presented VIGIL AI, a multimodal industrial anomaly detection and maintenance orchestration platform that addresses three structural limitations of existing predictive maintenance tools: unimodal signal analysis, detection-only architectures, and static health status persistence. The platform's central contribution — the VIGIL Score, a formally specified weighted linear fusion of sensor z-score, vision defect, and pattern match scores — was validated through classification performance evaluation (macro F1: 0.945), calibration analysis against empirical failure rates, and an ablation study confirming each modality's independent contribution. The closed-loop maintenance workflow, autonomous health re computation, and Realtime-propagated UI updates collectively demonstrate that a genuinely intelligent, genuinely self-managing industrial maintenance system can be delivered as a cloud-native web application accessible without on-premises infrastructure. The architecture's clean separation of concerns — stateless fusion function, unified AI inference router, trigger-based maintenance cascade — provides a sound foundation for the enhancements identified in the future work roadmap, most critically live IIoT integration and multivariate anomaly detection, which would complete the transition from demonstration platform to production-grade industrial health management system.

REFERENCES

[1] R. K. Mobley, An Introduction to Predictive Maintenance, 2nd ed. Butterworth-Heinemann, 2002.

[2] J. Moubray, Reliability-Centred Maintenance, 2nd ed. Industrial Press, 1997.

[3] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery di-agnostics and prognostics implementing condition-based maintenance," Mechanical Systems and Signal Processing, vol. 20, no. 7, pp. 1483–1510, 2006.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, vol. 41, no. 3, pp. 1–58, 2009.

[5] W. A. Shewhart, Economic Control of Quality of Manufactured Product. Van Nostrand, 1931.

[6] E. S. Page, "Continuous inspection schemes," Biometrika, vol. 41, no. 1/2, pp. 100–115, 1954.

[7] S. W. Roberts, "Control chart tests based on geometric moving aver-ages," Technometrics, vol. 1, no. 3, pp. 239–250, 1959.

[8] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," Mechanical Systems and Signal Processing, vol. 21, no. 6, pp. 2560–2574, 2007.

[9] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in Proc. MLSDA Workshop, 2014, pp. 4–11.

[10] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in Proc. ICLR, 2014.

[11] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in Proc. IEEE ICDM, 2008, pp. 413–422.

[12] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short-term memory networks for anomaly detection in time series," in Proc. ESANN,

2015, pp. 89–94.
[13] J. Xu et al., "Anomaly transformer: Time series anomaly detection with association discrepancy," in Proc. ICLR, 2022.
[14] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," arXiv preprint arXiv:1803.01271, 2018.
[15] Y.-J. Cha, W. Choi, and O. Bu¨yu¨ko¨ztu¨rk, "Deep learning-based crack damage detection using convolutional neural networks," Computer-Aided Civil and Infrastructure Engineering, vol. 32, no. 5, pp. 361–378, 2017.
[16] W. Du, J. Shen, J. Fu, G. Zhang, and Q. He, "Approaches for im-provement of the X-ray image defect detection of automobile casting aluminum parts based on deep learning," NDT & E International, vol. 107, 2019.
[17] D.-T. Hoang and H.-J. Kang, "A survey on deep learning-based bearing fault diagnosis," Neurocomputing, vol. 335, pp. 327–335, 2019.
[18] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD — a comprehensive real-world dataset for unsupervised anomaly detection," in Proc. CVPR, 2019, pp. 9592–9600.
[19] A. Radford et al., "Learning transferable visual models from natural language supervision," in Proc. ICML, 2021, pp. 8748–8763.
[20] Gemini Team, Google, "Gemini: A family of highly capable multimodal models," arXiv preprint arXiv:2312.11805, 2023.
[21] X. Xu et al., "Large language models for industrial fault diagnosis: A survey," arXiv preprint arXiv:2309.00033, 2023.
[22] T. Schick et al., "Toolformer: Language models can teach themselves to use tools," in Proc. NeurIPS, 2023.
[23] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. NeurIPS, 2020, pp. 9459–9474.