AEM JOURNAL.ORG

Research on the Sharing and Dynamic Scheduling Mechanism of Vocational Training Instruments and Equipment Driven by Instrument Portrait

Xiantu Lin
School of Information Engineering, Taizhou Vocational College of Science & Technology, Taizhou 318020, Zhejiang, China
Corresponding author: Xiantu Lin (e-mail: mickey007mouse168@163.com)

ABSTRACT To address low sharing utilization and inefficient coordination of practical training instruments in vocational colleges, this paper proposes a platform-oriented governance and dynamic scheduling mechanism driven by an instrument portrait data foundation. The instrument portrait serves as a unified and auditable profile for each instrument, covering basic profile, technical specifications, operational availability, teaching usage, research linkage, industry service, and performance indicators. A lightweight data-governance scheme is adopted to ensure consistency, timeliness, and accountability of portrait updates by defining data sources, responsibilities, update triggers, and quality rules. On this basis, a rule-based and explainable scheduling engine models resource-call events, concurrency, and conflicts under institutional constraints such as access rules, teaching priority windows, maintenance and calibration restrictions, and service deadlines. An operating-status recognition microservice can be integrated as operational evidence to update portrait states and support exception-handling workflows, while an LLM-based assistant supports natural-language querying, rule explanation, and administrator suggestions. The framework improves transparency, coordination efficiency, and sharing governance for heterogeneous training equipment.

INDEX TERMS instrument portrait, vocational training equipment, dynamic scheduling, data governance, explainable scheduling

I. INTRODUCTION

VOCATIONAL colleges rely on practical training instruments and equipment to deliver skill-oriented courses. In many colleges, however, equipment information is fragmented across departments, and reservation, utilization, maintenance, and fault records are not effectively integrated [1]. As a result, resource visibility is weak, sharing remains low, conflicts and idle time increase, and exception handling is often delayed. Improving management efficiency and sharing utilization therefore requires a platform that makes equipment resources observable, schedulable, and traceable.

To make resource sharing "visible and actionable", this study establishes an instrument portrait data foundation as the unified data layer of the sharing platform. The portrait consolidates seven management-oriented dimensions–basic profile, technical specifications, operational status, teaching usage, research linkage, industry services, and performance evaluation–so that users and administrators can understand each equipment unit from "what it is" to "how it is used and maintained" [1-2]. Table 1 summarizes the recommended field set.

To make the instrument portrait a reliable basis for sharing governance and dynamic scheduling, the platform adopts a minimal but enforceable governance scheme.

(1) Data sources and coverage. Portrait fields are integrated from (i) asset ledgers and procurement/finance records, (ii) reservation and access-control logs, (iii) maintenance/repair/calibration and inspection records, (iv) teaching timetables and training project records, (v) research project metadata and outputs (project IDs, method packages, publications/patents where applicable), (vi) industry/third-party service orders and delivery logs, and (vii) user feedback and satisfaction ratings. Where IoT/monitoring is available, operating signals and alarm codes are ingested as time-stamped evidence for status updates.

(2) Unified definitions (data "caliber"). The platform enforces unique equipment IDs, standardized taxonomy (category-subtype), location coding, and consistent definitions of key time variables (reservation time, actual start/finish, downtime, maintenance window). Status values adopt a management-oriented state set (e.g., idle/occupied/restricted/unavailable), and all derived indica-

TABLE 1. Recommended instrument portrait field set

Dimension	Representative fields (examples, 5–8 per dimension)
Basic profile	Equipment ID; name/model; category-subtype; owning department; location/lab; purchase date & cost; asset status (in service/retired); access level
Technical specifications	Capacity/range; accuracy/precision (or tolerance); supported tasks/experiments; software/firmware version; interface/communication capability; environmental requirements; safety constraints; required qualifications
Operational status	Current state (idle/occupied/restricted/unavailable); run/standby/shutdown flag; anomaly/alarm code; last start/stop time; maintenance window; calibration/verification due date; cleaning/preheating requirement; downtime reason
Teaching usage	Supported courses; training project/item; class/group; instructor; planned teaching hours; actual teaching hours; peak teaching periods; teaching priority rules
Research linkage	Project ID; research team/PI; sample/task type; method package/procedure; research schedule windows; related outputs (reports/papers/patents if applicable); data retention requirement
Industry services	Client/cooperation unit; service type; order ID; requested SLA/deadline; pricing/charging rule; delivery record; compliance requirements; dispute/feedback record
Performance evaluation	Utilization rate; cross-department sharing rate; cancellation/no-show rate; mean fault response time; mean repair duration; uptime/downtime; service on-time rate; user satisfaction score

tors (utilization rate, response time, cancellation rate) are computed from auditable logs to avoid manual bias.

(3) Roles and accountability (RACI).

R (Responsible): equipment administrators maintain asset profile, technical specifications, and maintenance/calibration records; academic affairs/training offices maintain teaching-related fields; research offices/PI teams maintain research linkage fields; industry-service offices maintain external service fields; IT/platform team maintains integration, permissions, and audit logs.

A (Accountable): the training center (or school-level equipment management office) owns the portrait standard and approves major field changes.

C (Consulted): finance/procurement, safety/quality offices, and department heads are consulted for standards and compliance.

I (Informed): instructors, students, and authorized external users are informed via the platform's portrait view and change notifications.

(4) Update frequency and triggers. Static fields (basic profile, most technical specs) are reviewed quarterly or upon procurement/relocation; teaching usage fields are updated per semester/term or by schedule import; research/industry linkage fields are updated when a project/service order is created or closed; operational status is updated in near real time when monitoring is available, otherwise updated event-by-event via check-in/out and administrator confirmation; maintenance/calibration updates are event-driven and time-window based.

(5) Data quality rules and audit. Mandatory fields are validated at entry (e.g., equipment ID, category, location, access level, key technical limits). Cross-field rules are enforced (e.g., "unavailable" status must map to a maintenance/fault record). Timeliness rules set maximum reporting delays for critical states and exceptions; abnormal events must be closed within a defined SLA with reasons and responsible parties recorded. All updates are versioned with timestamps and operators to support traceability and accountability.

The integration of advanced information technology into vocational education has transitioned from basic digitalization to intelligent, data-driven resource governance. The shared environment of practical training instruments and equipment in vocational colleges utilizes information technology to provide a sharing platform for a certain range of instruments and equipment. Its technical characteristics are digitization, networking, intelligence, and multimedia. Its basic features are openness, sharing, interaction, and collaboration. The sharing of practical training instruments and equipment in vocational colleges can be divided into four levels: common knowledge, sharing, remote operation, and collaborative services. Common knowledge mainly refers to the publication of static information about instruments and equipment, providing directory services through the establishment of resource indexes, telling users "Where there are instruments and equipment", and increasing the sharing of dynamic information about instruments and equipment on the basis of common knowledge.

By establishing instrument and equipment work schedules, reservation rules, etc., users are allowed to book instruments and equipment online, that is, "what instruments and equipment to use at what time and place". Remote operation is aimed at instruments and equipment that are supported by both software and hardware, allowing users to remotely operate and monitor them. Collaborative services establish a service grid for the shared environment of training instruments and equipment in vocational colleges, allowing multiple instruments and equipment to collaborate to complete complex tasks for users. In this work, the focus is platform-oriented sharing and dynamic scheduling in vocational colleges, where instrument portraits provide unified constraints and online states, and scheduling models generate explainable allocations under real operational conditions.

Research on the sharing levels of training instruments and equipment in vocational colleges can be divided into two categories: one is from a qualitative perspective, studying the sharing mechanism and management system of training instruments and equipment in vocational colleges; the other is from a quantitative perspective, modeling and optimizing the sharing process to improve the efficiency of instrument and equipment utilization. Modeling optimization mainly refers to the job scheduling in the shared environment of training instruments and equipment in vocational colleges, that is, considering how to reasonably allocate limited resources to meet user needs as much as possible in response to the situation of insufficient supply of networked instruments and equipment. Based on existing statistical data, most of the networked instruments and equipment are scarce resources that are in short supply. Therefore, in the process of sharing practical training equipment in vocational colleges, there is an important issue that urgently needs to be solved: how to configure "limited" equipment resources to meet users' "unlimited" needs as much as possible, and maximize the utilization efficiency of networked equipment.

This article proposes a platform-oriented sharing and dynamic scheduling solution driven by an instrument portrait, in which the Petri-net scheduling engine and the online status-recognition service are embedded as core modules to support explainable, state-aware scheduling in vocational colleges. The scheduling problem of shared training instruments and equipment in vocational colleges is studied in order from easy to difficult. Designed and implemented core modules such as shared management, user reservation, and unit scheduling.

II. RELATED RESEARCH

A. RESEARCH STATUS OF INSTRUMENT AND EQUIPMENT MANAGEMENT IN VOCATIONAL COLLEGES

In recent years, vocational colleges have experienced rapid growth in the scale and diversity of training instruments and equipment. This expansion has made equipment management a central issue for improving teaching quality, ensuring operational safety, and enhancing the efficiency of shared utilization. Existing research generally recognizes that vocational-college equipment management is no longer limited to inventory administration, but increasingly involves asset governance, cross-department coordination, and data-enabled decision-making, with an emphasis on building integrated platforms that connect reservation, utilization auditing, and maintenance workflows.

From the perspective of asset governance and resource allocation, studies highlight the need to strengthen overall planning, promote cross-department sharing, and establish utilization-oriented evaluation mechanisms. In TVET contexts, Sulistyo et al. [3] propose an asset management and tracking system based on ubiquitous computing, emphasizing that effective governance requires unified identification, traceable usage records, and standardized management processes. This line of work suggests that improving sharing efficiency depends not only on "open access" policies, but also on the operational feasibility of tracking, accountability, and lifecycle management.

Informatization and platform integration studies consistently highlight systemic challenges, including interdepartmental data silos, heterogeneous governance protocols, and insufficient auditability for utilization and operational anomalies. Consequently, a smart-campus-oriented management platform is often advocated to consolidate multi-source records–such as asset ledgers, reservations, maintenance/calibration work orders, and monitoring streams–into a consistent equipment information base. Such integration supports auditable operations and enables measurable indicators for sharing governance (e.g., utilization rate, cancellation/no-show rate, response time for faults and maintenance).

A critical challenge for shared scheduling in vocational colleges is that many systems still rely heavily on user-side records (e.g., reservations and manual check-in/out) and lack standardized, instrument-side operational state evidence. This limitation can delay fault detection, increase rescheduling costs, and weaken safety governance. Recent work has begun to address this gap by incorporating equipment-side operational-state characterization. For example, Amangeldy et al. [4] explore IoT-based unsupervised learning to characterize laboratory operational states for improved safety and sustainability, indicating that state awareness can strengthen monitoring and risk response. For vocational training equipment, this motivates embedding operating-status recognition as a platform capability and feeding time-stamped results back into a governed equipment information profile (instrument portrait), thereby supporting state-aware scheduling and exception handling.

In summary, research on vocational-college instrument and equipment management increasingly converges on three themes: (i) governance-oriented asset management and sharing evaluation, (ii) integrated platforms with unified data standards and traceability, and (iii) the enhancement of instrument-side state awareness to reduce exception-handling cost and improve safety. However, many existing implementations still treat operational state as peripheral or manual, leaving a gap in how instrument-side evidence is systematically incorporated into the equipment information base to support accountable scheduling and lifecycle governance.

B. RESEARCH STATUS OF INSTRUMENT AND EQUIPMENT STATUS RECOGNITION

Status recognition for vocational training equipment refers to identifying whether an equipment unit is ready for use, in operation, temporarily restricted, shut down, or abnormal, so that the sharing platform can manage availability, safety, and rescheduling. In vocational colleges, typical training equipment (e.g., machining-oriented equipment and automated analytical instruments) operates through motors/pumps/actuators and control loops. Therefore, equipment status is reflected not only in electrical load proxies (current/power) but also in management-critical signals and records such as controller states, alarm codes, maintenance/calibration validity, and planned restriction windows (preheating, cleaning, verification). From a platform perspective, "status recognition" should be understood as a governed state-identification process that combines operational evidence (telemetry when available) with auditable management logs (reservation, check-in/out, maintenance, calibration, and inspection).

Existing research related to equipment status in educational and laboratory contexts focuses more on management frameworks and quality assurance than on algorithm-centric recognition. Studies on laboratory management emphasize that effectiveness depends on planning, organization, and evaluation mechanisms, which require reliable visibility of equipment readiness, utilization, and exception handling rather than only inventory records [5]. In testing and calibration environments, accreditation practices highlight the importance of traceability, calibration/verification control, and compliance-driven documentation, which directly implies that "equipment status" must include whether instruments are within valid calibration periods and whether operating conditions meet required controls [6]. At the standards level, ISO 55000 frames asset management as lifecycle governance supported by consistent information, responsibilities, and performance evaluation, indicating that equipment states should be standardized, comparable, and decision-actionable across departments and time [7]. From an operational implementation viewpoint, computerized maintenance management systems (CMMS) are widely used to support maintenance planning, work-order tracking, and teaching-oriented maintenance practices, reinforcing the role of maintenance and fault workflows as first-class state evidence in equipment governance [8-9]. In higher-education shared instrumentation settings, open sharing strategies further stress that increasing utilization requires transparent availability windows and accountable processes, which depend on reliable status fields that can support scheduling and conflict resolution [10].

A key gap across many vocational-college implementations is that equipment "status" is often represented indirectly by user-side records (reservation/check-in) and fragmented departmental practices, while instrument-side evidence and maintenance/calibration validity are not systematically integrated into a standardized state model. This weakens fault response, increases rescheduling cost, and makes performance evaluation less credible. Therefore, in this study, status recognition is positioned as a platform capability embedded in the instrument-portrait data foundation: governed state fields are updated using traceable sources (maintenance/calibration records and usage logs as the primary evidence, with telemetry-based recognition as an optional enhancement when available). This positioning ensures that the scheduling mechanism remains policy-governed and auditable, while enabling state-aware sharing governance and exception handling at the platform level.

C. FEATURE SELECTION AND GOVERNANCE FOR PLATFORM-ORIENTED STATUS RECOGNITION

In a vocational-college sharing scenario, heterogeneous training instruments often lack a unified standard for constructing equipment-operation datasets. To keep the platform scalable across categories (e.g., machining-oriented equipment and automated analytical instruments), this study treats feature engineering as a governed platform capability rather than a one-off modeling step. This positioning follows the logic of platform-based reservation, shared-facility operation, and open-sharing governance in higher education, where stable data structures, traceable workflows, and comparable utilization evidence are prerequisites for sustainable sharing efficiency [11-14]. Operational telemetry is first standardized by a signal abstraction layer (generalized load as the primary proxy, with optional process/control and alarm channels when available) so that feature computation, labeling, and evaluation can follow consistent rules across equipment types.

(1) Feature registry and version control. The platform maintains a feature registry that defines: feature names and meanings, required input channels, computation windows (length/step), applicable equipment categories, and downstream usage (recognition evidence vs governance indicator). Feature sets are managed in versions (e.g., v1.0 baseline; v1.1 tuned for a category), and any change must be traceable, comparable, and reversible to avoid disrupting scheduling governance.

(2) Low-cost screening as method selection basis (ANOVA/Pearson). For initial screening, filter-based selection is adopted mainly to reduce redundancy and to improve maintainability. Considering the platform's mixed data types, ANOVA is used as an efficient indicator for screening the association between continuous features and multi-class state labels, while the Pearson correlation coefficient is used to identify highly correlated continuous features and remove duplicates. These statistical tools are used as supporting evidence for feature governance, and the final retained feature subset is determined jointly by interpretability and platform-level validation results.

(3) Category templates for window parameters and thresholds. To ensure consistent deployment, the platform manages feature windows and governance thresholds through category templates. Equipment categories with fast state transitions (typical in machining-oriented operations) can adopt shorter windows and tighter update steps, while categories with longer and more stable stages (typical in method-driven analytical workflows) can adopt longer windows and explicit "restricted" buffers (e.g., preheating/equilibration/cleaning/calibration). For machining-oriented equipment, generalized-load proxies derived from spindle/drive current are commonly used to reflect operating conditions, supporting the feasibility of category-based parameter templates grounded in load-pattern dynamics [15]. Thresholds used for state updates and exception triggers are not hard-coded in the model; instead, they are stored as configurable policy parameters with responsible roles, review cycles, and audit logs.

(4) Continuous monitoring and drift handling. Because equipment usage patterns may shift by semester, workload, or maintenance condition, the platform periodically rechecks feature relevance and stability. When drift is detected (e.g., increased false abnormal triggers or unstable transitions), the platform updates the feature/threshold configu-

ration through controlled iteration: “propose - evaluate on recent logs - approve - release - monitor”, thereby keeping the recognition service and scheduling governance stable over time.

(5) Integration with labeling and data governance workflow. Dataset construction is aligned with platform business processes: reservations, check-in/out, maintenance/calibration records, and administrator confirmations form auditable labels for running/standby/shutdown/abnormal states, and are linked to feature windows through timestamps. In vocational teaching contexts, aligning labeling and state definitions with course-linked training tasks and structured lab sessions helps ensure that platform states reflect real instructional workflows and remain interpretable for instructors and administrators [16]. This design ensures that feature selection and configuration can be governed by operational evidence and accountability roles, enabling reproducible and explainable platform deployment.

III. PROBLEM DESCRIPTION

A. SHARED SCHEDULING OF PRACTICAL TRAINING INSTRUMENTS AND EQUIPMENT IN VOCATIONAL COLLEGES

Vocational colleges’ practical training equipment is a scarce resource. In order to better utilize the social and economic benefits of limited resources, it is necessary to optimize the allocation of resources on the basis of fairness and efficiency, in order to ensure that users who need the equipment the most obtain resources. This is the job scheduling problem in a shared environment. Figure 1 shows a schematic diagram of the shared environment for practical training equipment in vocational colleges. The shared environment for practical training instruments and equipment in vocational colleges consists of three elements: (i) equipment resources (heterogeneous training instruments and equipment), (ii) stakeholders (students, instructors, training-center administrators, department managers, and authorized external users when applicable), and (iii) a multi-tier platform architecture.

The implementation follows a microservice-oriented design: the Data Layer (MySQL/Redis) manages portrait dimensions; the Logic Layer hosts the CTPN engine and the status-recognition service; and the Application Layer provides a RESTful API for external integration. To improve administrator efficiency, the embedded LLM-based assistant utilizes a Retrieval-Augmented Generation (RAG) framework, connecting the LLM to the portrait database to provide natural-language answers for ‘who is using device X’ or ‘explain why reservation Y was rejected’ based on current Petri-net guard conditions.

Rather than positioning the platform as a “large-scale national subsystem”, this study focuses on a campus-governance-oriented platform whose core function is to make resources visible, allocatable, and auditable under institutional rules. Specifically, the platform provides (a) a unified instrument portrait for resource transparency, (b) rule-based reservation and approval workflows, (c) conflict resolution and explainable scheduling under multi-objective constraints (fairness, teaching priority, timeliness, and utilization), and (d) traceable exception handling (cancellation/no-show, fault escalation, maintenance locking). To ensure accountability in daily operation, platform governance follows a role-and-responsibility structure consistent with the portrait data governance: equipment administrators maintain asset and maintenance/calibration records, training offices maintain teaching schedules and priorities, departments define access and qualification rules, and the platform team ensures data integration, audit logs, and permission control.

Under this governance structure, a clear hierarchy is established to resolve conflicts between automated sensor data and manual logs. In the event of a discrepancy–such as a sensor detecting ‘idle’ status while a human ‘check-in’ is active–the platform defaults to a ‘Safety-First’ priority where manual administrative locks or active check-in logs override sensor-based ‘available’ states. Conversely, if sensors detect an ‘abnormal’ state during a manually confirmed session, the platform triggers an immediate ‘Risk Observed’ flag for administrator verification, ensuring that automated microservices support rather than bypass human accountability.

The proposed platform forms a closed loop of “portrait - scheduling - feedback”. The instrument portrait provides unified fields for capability constraints, access rules, availability windows, and maintenance states. The Petri-net engine schedules resource-call events under concurrency and conflict constraints. Online operating-state recognition feeds back running/standby/shutdown/abnormal states to update the portrait, and abnormal states trigger rule-based rescheduling and notification.

B. PETRI-NET SCHEDULING MODEL DRIVEN BY INSTRUMENT PORTRAIT

Before constructing the Petri-net scheduling model, the instrument portrait is specified as a parameterization layer that translates multi-dimensional equipment information into auditable scheduling constraints and governance-oriented priorities. In implementation, the portrait is not treated as a static description, but as a controlled dataset whose key fields are sourced from traceable logs (reservations, check-in/out, maintenance/calibration records, teaching timetables, research/service orders) and updated by defined roles and rules. Specifically, the portrait contributes four classes of scheduling parameters: a) Capability constraints: whether an instrument can execute a requested training/research/service task, and its technical limits (e.g., supported process/method package, capacity/range, qualification requirements); b) Availability and restriction windows: reservable time windows and non-reservable windows derived from teaching plans, preheating/cleaning/calibration routines, planned maintenance, and safety/access rules; c) State and exception variables: management states (idle/occupied/restricted/unavailable) to-

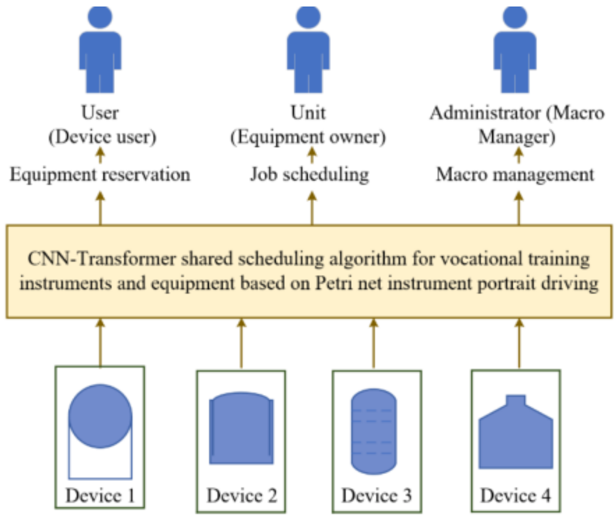

FIGURE 1. Shared Environment for Training Instruments and Equipment in Vocational Colleges

gether with run/standby/shutdown/abnormal signals and alarm flags that may trigger exception handling; d) Governance weights and priorities: multi-objective trade-offs (fairness, utilization, service timeliness) instantiated by policy rules and KPI indicators (e.g., cross-department sharing target, cancellation penalty, fault response SLA).

Based on these portrait-derived parameters, discrete resource-call events are modeled using colored timed Petri nets to represent sequencing, concurrency, and conflicts under complex constraints, enabling explainable schedule generation and rule-based rescheduling. To keep the scheduling model implementable and management-oriented, portrait fields are mapped to Petri-net elements as follows. Places (P) represent instrument states and controllable resource conditions (e.g., idle/occupied/restricted/unavailable, maintenance window active, calibration due). Tokens represent either (i) available resource units, or (ii) reservation/job requests; token "colors" encode management attributes such as requester type (teaching/research/industry), priority level, required qualification, earliest start time, deadline/SLA, and estimated duration. Transitions (T) represent auditable events (reservation approval, check-in, job start, job finish, cancellation/no-show, maintenance start/end, exception escalation).

Enabling/guard conditions of transitions are directly instantiated from portrait constraints: capability matching (supported task/method package), access/qualification rules, time-window feasibility, and restriction windows (preheating/cleaning/calibration/maintenance). Timing functions attach realistic durations (setup + operation + cleaning) and enforce SLA-related deadlines. This mapping ensures that the Petri-net model is not an abstract algorithmic construct, but a governance-aware scheduling engine whose inputs and decisions can be traced back to portrait fields and operational logs. Table 2 summarizes the key field-to-parameter mapping used in this study.

Petri nets are adopted here as an explainable discrete-event representation for platform scheduling. The proposed sharing mechanism is formally defined as a Colored Timed Petri Net (CTPN) 7-tuple:

$$\Sigma = (P, T, C, \alpha, \beta, \gamma, \tau)$$

Here, P is a finite set of places representing portrait-derived states $\{p_{idle}, p_{occupied}, p_{restricted}, p_{unavailable}\}$; T is a set of transitions representing platform events; C is the color function mapping tokens to management attributes (requester type, priority, duration); α and β are the forward and backward incidence functions; γ is the guard function ($\gamma : T \rightarrow \text{Boolean}$) enforcing portrait capability and access constraints; and τ is the time delay function ($\tau : T \rightarrow \mathbb{R}^{+}$) representing setup and processing durations. This formalization allows for the rigorous verification of system properties such as liveness and reachability during scheduling.

Places represent auditable resource conditions and schedulable states (e.g., idle-available, occupied, restricted, unavailable; maintenance window active; calibration due).

Transitions represent platform events that can be traced in logs (reservation approval, check-in, job start, job finish, cancellation/no-show, maintenance start/end, exception escalation).

Token colors carry management attributes of a request or a resource unit (requester type: teaching/research/industry; priority; qualification requirement; earliest start; deadline/SLA; estimated duration).

Guards and time windows implement institutional rules as constraints (capability match, access/qualification, teaching priority windows, preheating/cleaning/calibration restrictions, planned maintenance locks).

Audit hooks ensure that each scheduling decision and rescheduling trigger is linked back to portrait fields and operational evidence (Table 2), supporting accountability and post-hoc inspection. Figure 2 illustrates the mapping

AEM JOURNAL .ORG

TABLE 2. Mapping from instrument portrait fields to scheduling parameters and governance rules

Portrait dimension / field	Scheduling role (constraint/parameter/weight)	Example of governance rule in platform
Basic profile: Equipment ID, location, owning unit	Resource identifier; location constraint	Cross-campus jobs require travel/setup buffer; location conflicts blocked
Technical specs: supported task/method package	Capability constraint	Request must match “supported method/task” set; otherwise rejected/recommended alternatives
Access level & qualification requirement	Access/eligibility constraint	Only certified users can reserve; otherwise route to instructor/administrator approval
Teaching timetable & peak periods	Priority window; time restriction	Teaching reservations have protected windows; non-teaching tasks avoid peak teaching periods
Preheating/cleaning/calibration requirement	Restriction window (non-reservable)	Setup/cleaning time automatically appended; prevents back-to-back incompatible jobs
Planned maintenance/calibration plan	Availability window; hard constraint	Maintenance window locks resource; rescheduling triggers when plan changes
Operational state (idle/occupied/restricted/unavailable)	State variable; feasibility constraint	Unavailable – block new assignments; restricted – only allow approved internal jobs
Abnormal/alarm flags	Exception trigger	Abnormal triggers: cancel/hold current plan – notify admin – reallocate to substitute resource
Research linkage / project window	Soft constraint or preference	Research tasks allowed in defined project windows; can request priority escalation with justification
Industry service: order deadline/SLA	Deadline constraint; penalty weight	Deadline miss penalty increases; scheduler prefers feasible completion before SLA
Performance KPIs: utilization, sharing rate, cancellation rate	Objective weights (multi-objective)	If sharing rate below target – increase cross-department weight; high cancellation – increase penalty
Fault response time SLA, repair duration	Exception-handling KPI	If SLA risk detected – escalate maintenance; temporarily reduce booking horizon

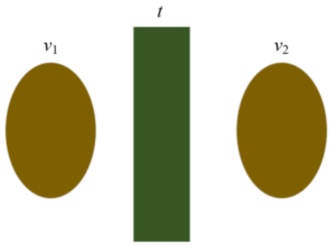

FIGURE 2. Petri Net Model

from resource states to scheduling actions: transitions move tokens between places according to rules and time windows, making concurrency and conflicts explicit and supporting rule-based rescheduling when exceptions occur (fault alarms, overtime, emergency maintenance).

C. EXTRACTION OF TIME SERIES CHARACTERISTICS OF INSTRUMENT AND EQUIPMENT OPERATION STATUS IN VOCATIONAL COLLEGES

The operating status of training instruments and equipment in vocational colleges is modeled using a two-layer status scheme to simultaneously support (i) online recognition from telemetry and (ii) management-oriented scheduling governance. Layer A (recognition labels) keeps a compact four-class set–shutdown, standby, run, abnormal–which is convenient for supervised learning and online inference from operating telemetry. Layer B (management states) translates recognition outputs and operational logs into schedulable states used by the platform: idle-available (can be reserved), occupied (in execution), restricted (temporarily non-reservable but planned, e.g., preheating/equilibration, cleaning, calibration/verification), and unavailable (fault/maintenance/lock). This separation ensures that the platform can remain auditable and rule-based: recognition outputs do not directly “decide” scheduling, but provide evidence that updates the portrait and triggers rescheduling rules. Accordingly, mapping values can be defined in two steps: Table 3 shows the numeric encoding for Layer A recognition labels, and Table 4 provides the governance-oriented mapping from Layer A plus logs (reservation, check-in/out, maintenance/calibration records) to Layer B management states.

To make the feature extraction applicable to heterogeneous training equipment, a signal abstraction layer is

TABLE 3. Numeric encoding of recognition labels (Layer A)

Recognition label	Mapping numerical values
Shut down	0
Standby	1
Run	2
Abnormal	3

adopted. The ‘Generalized Load Proxy’ (L_g) is mathematically defined as a normalized vector

$$L_g = \frac{S_t - S_{\min}}{S_{\max} - S_{\min}},$$

where S_t represents the primary operational signal (e.g., spindle current for CNC or pump pressure for HPLC) at time t. This Min-Max normalization transforms disparate physical units into a dimensionless [0,1] intensity index, allowing the model to apply consistent recognition logic across different equipment categories.

For each equipment category, the platform selects one or several telemetry channels that best represent (i) load/intensity (e.g., drive or pump current/power), (ii) process stability (e.g., pressure/flow/temperature stability indicators when available), and (iii) controller/alarm states (e.g., alarm codes and interlock signals). These raw channels are normalized and aligned into a unified time axis, so that the subsequent time-window feature extraction and learning model can be shared across different equipment types while preserving category-specific characteristics through input-channel mapping.

Based on the above description, this study extracts time window features from generalized load telemetry (and optional auxiliary process signals), including basic statistical information (instantaneous value, maximum value, minimum value) in one window and three sets of time domain descriptors (central trend, dispersion and shape descriptors, such as skewness and kurtosis). It also includes three major types of time-domain feature values: central tendency, off center tendency, and kurtosis. In specific calculations, a multi-scale sliding window approach is adopted. While a 30-minute aggregate window is used for long-term trend analysis, a high-frequency 1-minute subwindow is implemented for near real-time status triggering. This ensures that state transitions (e.g., from ‘idle’ to ‘run’) are detected and updated in the portrait within seconds, minimizing the lag to meet the requirements of back-to-back dynamic scheduling.

TABLE 4. Governance-oriented management states for scheduling (Layer B)

Management state (schedulable)	Typical sub-state examples	Rule-based derivation (from portrait + logs + Layer A)
Idle-available	ready/idle	Standby or shutdown and no restriction window; qualification & access satisfied
Occupied	machining/analysis in progress	Run and check-in/job-start confirmed
Restricted	preheating/equilibration; cleaning; calibration/verification; planned setup	Standby (or shutdown) but within planned restriction window (e.g., preheat/cleaning/calibration) imported from maintenance/calibration plan or method package
Unavailable	fault; maintenance; locked by safety/asset policy	Abnormal or maintenance/fault record active or admin lock

For machining-oriented equipment, the generalized load indicator can be instantiated by spindle/servo current or power proxies, while controller states can be represented by CNC alarm/interlock codes and program execution flags. For analytical instruments, optional auxiliary process signals may include pump pressure/flow stability, temperature control status, and method-stage indicators (e.g., equilibration/run/clean), which are particularly useful for distinguishing "restricted" windows such as preheating, equilibration, cleaning, and calibration that are critical for reservation feasibility.

Management-oriented window features and their use in scheduling governance. In the proposed platform, time-window features are extracted for two management purposes: i) to support the embedded operating-status recognition service (Layer A labels: shutdown/standby/run/abnormal), and ii) to provide auditable indicators for updating portrait fields and triggering rule-based scheduling governance (Layer B management states: idle-available/occupied/restricted/unavailable). Feature design follows interpretability-first principles. A 30-minute rolling window (configurable by equipment category) is used to aggregate operational evidence prior to each time point. Instead of emphasizing mathematical derivations in the main text, this study adopts a compact, interpretable feature set (Table 5) and uses clear thresholds and event rules to support explainable exception handling.

How features support governance decisions. Level/Intensity features (e.g., mean, min, max, median of generalized load) indicate whether a device is likely executing a job (occupied) versus idle (idle-available). Volatility/Stability features (e.g., range, standard deviation, coefficient of variation; optional process stability indicators such as pressure/flow/temperature stability) help differentiate "standby but restricted" windows (preheating/equilibration, cleaning, calibration) from true idle states that can accept new reservations. Shape/outlier features (e.g., skewness/kurtosis) provide auxiliary evidence for rare abnormal patterns and are mainly used as supporting signals rather than primary decision drivers. All feature computation is logged and versioned as part of the portrait update trail, so that any scheduling change caused by abnormal detection or state transition can be traced back to window evidence and operational records.

In the platform implementation, any transition of Layer B management states (idle-available/occupied/restricted/unavailable) is written back to the portrait as a timestamped update, and mapped to Petri-net transitions (e.g., job start/finish, exception escalation, maintenance lock) to trigger rule-based rescheduling and notifications.

IV. OPERATING-STATUS RECOGNITION MICROSERVICE FOR PLATFORM SCHEDULING

In this platform-oriented study, operating-status recognition is implemented as a microservice that provides timestamped operational evidence for the instrument portrait and exception handling. It does not replace the rule-based scheduling engine. Scheduling decisions remain auditable and policy-governed; recognition outputs only support portrait updates and trigger predefined governance actions (e.g., temporary restriction, resource locking, rescheduling, and maintenance ticketing).

A. SERVICE INPUTS, OUTPUTS, AND UPDATE RULES

Inputs are produced by the platform's signal abstraction layer: generalized load telemetry as the primary channel (e.g., drive/pump current or power proxy) and optional process/control channels (e.g., pressure/flow/temperature stability indicators, controller states, alarm/interlock codes) when available. The service also consumes contextual metadata such as equipment ID, equipment category, and reservation window information to align inference with platform logs. Outputs are (i) Layer A recognition labels (shutdown/standby/run/abnormal) with confidence scores and (ii) optional anomaly tags used for governance (e.g., usage anomaly vs fault anomaly). Outputs are written back to the instrument portrait as timestamped state evidence, together with a "reason code" (signal evidence / alarm evidence / persistence evidence / manual confirmation) to support traceability. The update frequency follows the platform's configurable rolling-window policy (Section 2.3 and Section 3.3), so different equipment categories can adopt different window lengths and steps without changing portrait schema or scheduling logic.

B. GOVERNANCE SAFEGUARDS AND SCHEDULING ACTIONS

To prevent false triggers from causing unnecessary schedule disruptions, the platform adopts evidence-first safeguards.

Persistence rule: a state change to "abnormal" or "unavailable" requires persistence across multiple windows.

Corroboration rule: abnormal escalation is strengthened only when corroborated by operational logs (alarm codes,

TABLE 5. Interpretable window features and their governance usage in the sharing platform

Feature group	Typical features	Management meaning	Example rule/action in platform
Level / intensity	mean, median, min, max of generalized load tendency	Distinguish idle vs occupied	If load level sustained above baseline for $\geq X$ min – mark "occupied", block conflicting bookings
Volatility / dispersion	range, standard deviation, coefficient of variation (CV)	Detect instability or transitions; support state refinement	If CV spikes during booked session – flag "execution instability", notify admin/instructor
Process/control stability (optional channels)	pressure/flow/temperature stability, controller flags, alarm/interlock codes	Identify restricted windows (preheating/equilibration/cleaning/calibration) vs true idle	If method-stage indicates cleaning/calibration – set "restricted" window, auto-append buffer time
Shape / outlier (auxiliary abnormal evidence)	skewness, kurtosis	Assist rare-pattern detection	If abnormal evidence persists across Y windows – escalate to "unavailable", trigger rescheduling + maintenance ticket
Reservation & log-derived evidence (non-sensor)	check-in/out, overtime, cancellation/no-show, maintenance/calibration record	Strong audit evidence for governance	No-show – penalty + release slot; overtime – shift subsequent jobs or reassign substitute device

check-in/out anomalies, maintenance records) or administrator confirmation.

Staged actions: (a) alert and annotate portrait as "risk observed"; (b) temporarily mark "restricted" to hold conflicting future reservations; (c) lock as "unavailable" only when confirmed/persistent, then trigger rescheduling and open a maintenance/inspection ticket with traceable evidence.

Controlled release and rollback: model/threshold updates are governed through "pilot - evaluate on recent logs - approve - release - monitor", with rollback if the update increases false alarms, destabilizes portrait states, or raises rescheduling workload. Note on implementation variants. In experiments, multiple internal configurations are compared to support deployment choices (Section 5.2). Architectural details of the underlying model are not the focus of this management-oriented paper and are treated as interchangeable implementations of the same "status recognition service" capability.

V. EXPERIMENTAL ANALYSIS

A. FEATURE AND FEATURE CORRELATION VERIFICATION

Rather than pursuing a feature set optimized only for offline model performance, this section verifies feature templates that can be governed and maintained at the platform level. The goal is to select a default feature subset based on quantifiable metrics. 'Interpretability' is operationally defined as the SHAP (SHapley Additive exPlanations) value of a feature, ensuring its contribution to a state change is human-understandable. 'Operational burden' is measured by the computational latency (ms) and data ingestion cost per device. Features in Table 6 were validated using an independent cross-validation set where 'stability' was defined as the variance of recognition accuracy across 10 heterogeneous device categories. This ensures the template selection is grounded in empirical performance rather than subjective governance preference. Therefore, Table 6 is interpreted as evidence for platform feature governance decisions: which feature groups should be included in the default template, and which should be treated as optional or exception-only signals.

(1) Default template selection (interpretability-first). Results in Table 6 indicate that the datasets constructed from basic features and central-tendency (concentration) features deliver consistently high recognition quality, making them suitable as the platform's default template for broad equipment categories. This choice is aligned with operational interpretability: "level/intensity" and "typical operating level" provide direct evidence for distinguishing idle/occupied states and for supporting auditable scheduling constraints (availability and occupation).

(2) Why dispersion/departure features are not default. "Departure trend" features are more sensitive to transient fluctuations and category-specific behaviors (e.g., different control strategies, different workload patterns), which increases cross-category tuning cost and reduces portability. Given their weaker performance in Table 2, they are better treated as category-specific optional features that may be enabled only when a given equipment category exhibits frequent short-cycle transitions and reliable auxiliary signals.

(3) Shape features as exception evidence (not a primary driver). Shape-related descriptors (e.g., skewness/kurtosis-type indicators) may provide additional evidence for rare abnormal patterns, but they are not recommended as default discriminators because they can be unstable under noise and data scarcity. In platform governance, they are positioned as auxiliary exception evidence: used to support abnormal escalation rules only after being corroborated by logs (alarm codes, maintenance/fault tickets) or persistence across multiple windows, thereby reducing false alarms and unnecessary rescheduling.

(4) Correlation governance and redundancy control. "Feature correlation verification" is treated as a platform maintenance task: the platform periodically reviews redundancy among continuous features using Pearson-based checks and retains a minimal, non-duplicative subset. When two features carry highly similar information, the platform keeps the one with clearer operational meaning and lower data dependency (e.g., keeping "mean level" over more complex but less interpretable alternatives). This keeps the feature registry stable, reduces overfitting risk, and supports consistent cross-department adoption.

(5) Platform configuration outputs. Based on this verification, the platform maintains: (i) a default feature template (basic + central-tendency), (ii) category optional templates (departure/dispersion enabled only when validated for a category), and (iii) an exception template (shape features used only for abnormal evidence and escalation). Each template is versioned, linked to applicable equipment categories, and governed through "propose-evaluate-approve-release-monitor" to ensure traceable updates and stable scheduling behavior.

TABLE 6. Verification results of platform feature templates by feature-group datasets

Feature category	A	MP	MR	MF	Kappa
Basic features	91.32%	92.01%	91.36%	0.903	0.852
Concentration trend	90.32%	91.15%	90.20%	0.904	0.836
Departure trend	79.13%	83.85%	77.61%	0.768	0.654
Shape descriptors (abnormal evidence: skewness/kurtosis)	56.04%	62.87%	50.24%	0.465	0.252

Based on Table 2, the platform releases a default feature template (basic + central-tendency) as the common baseline for heterogeneous equipment. Category optional templates are enabled only after pilot validation, and shape descriptors are reserved for exception evidence and escalation rules. These configurations are managed through a versioned registry and are reviewed periodically to accommodate semester-level workload shifts and equipment aging.

B. MODEL OPTION ANALYSIS FOR PLATFORM DEPLOYMENT

This section interprets operating-status recognition results from a platform deployment perspective. The recognition model in this study is positioned as an online service that updates portrait state fields and triggers rule-based scheduling actions (availability locking, exception escalation, and rescheduling). Therefore, model selection is not judged only by offline metrics, but by its expected impact on platform operations: (i) stability of portrait updates, (ii) reduction of false abnormal triggers and unnecessary rescheduling, (iii) maintainability and deployment cost, and (iv) governance requirements such as traceability, version control, and rollback.

Table 7 compares representative configurations from two model families, CNN-LSTM and CNN-Transformer. Overall, the CNN-Transformer group shows better recognition quality on this dataset, with the best-performing configuration (CIFN-SSC-CNN-Transformer) achieving higher accuracy and recall than the compared CNN-LSTM variants. This improvement is operationally meaningful: higher recognition reliability reduces “state jitter” in portraits (frequent unnecessary switching between idle/occupied/abnormal), lowers the probability of mistakenly blocking reservable time windows, and decreases the workload of administrators who must handle exception tickets and manual confirmations.

Deployment-oriented selection strategy. Centralized deployment (school-level platform server). When inference is served centrally and computational resources are sufficient, the platform can prioritize recognition stability and choose the highest-performing configuration reported in Table 7 to minimize false triggers and rescheduling frequency.

Lightweight/edge-friendly deployment (lab gateway or local controller). When resources are constrained, the platform should select a simplified configuration that keeps strong performance while reducing tuning and operational complexity (e.g., a single enhancement block rather than multiple blocks), and rely on governance rules to control risk: abnormal escalation requires persistence across multiple windows and/or corroboration by logs (alarm codes, maintenance records) before locking the resource.

Governance safeguards. Regardless of the chosen configuration, the platform enforces model versioning and auditable releases (pilot - evaluate on recent logs - approve - release - monitor). A rollback mechanism is required: if the model update increases false abnormal triggers or destabilizes portrait states, the platform reverts to the previous stable version and temporarily relies more on reservation/check-in/out logs for state updates. In summary, Table 7 is used as evidence to choose a recognition service that best supports auditable sharing governance and stable dynamic scheduling, rather than as an end in itself. The platform keeps the scheduling engine rule-based and explainable, while using recognition outputs as time-stamped operational evidence to improve state awareness and reduce exception-handling costs.

C. ABNORMAL-RECOGNITION GOVERNANCE AND DEFAULT FEATURE-TEMPLATE SIZE

Following the default feature-template verification and the deployment-oriented model option analysis, this section determines a maintainable default configuration for abnormal governance by selecting an appropriate feature-template size K. In the sharing platform, abnormal recognition is designed as an operational governance capability rather than a standalone modeling objective. Its main purpose is to provide early warning evidence for updating the instrument portrait (restricted/unavailable states) and for triggering rule-based exception handling (temporary holding, rescheduling, maintenance ticketing). Therefore, this section evaluates how the feature-template size K affects the stability of abnormal recall, in order to determine a maintainable default configuration for platform deployment.

Figure 3 reports the variation of abnormal recall with K under two representative recognition service configurations. Overall, recall improves rapidly when a small number of core features are enabled and then gradually enters a plateau region, indicating diminishing returns when continuously increasing K. From a platform-governance perspective, the plateau point is more important than the absolute peak: selecting a K near the plateau yields a configuration that is stable, interpretable, and easier to maintain, while avoiding excessive feature dependency and tuning costs across heterogeneous equipment categories.

Default K recommendation for platform rollout. Based on the stabilization trend in Figure 3, the platform adopts a default abnormal-recognition template with a moderate K (typically around the plateau region) as the baseline for heterogeneous equipment. Category-specific extensions

TABLE 7. Recognition-performance comparison for platform deployment options

Model Group	Model Name	A	MP	MR	MF
CNN-LSTM	CIFN-CNN-LSTM	86.33%	88.25%	90.13%	0.894
	SSC-CNN-LSTM	87.17%	88.18%	91.12%	0.902
	CIFN-SSC-CNN-LSTM	88.75%	90.52%	92.57%	0.898
CNN-Transformer	CIFN-CNN-Transformer	89.17%	90.68%	92.86%	0.894
	SSC-CNN-Transformer	91.23%	91.73%	93.23%	0.912
	CIFN-SSC-CNN-Transformer	92.46%	93.94%	94.34%	0.904

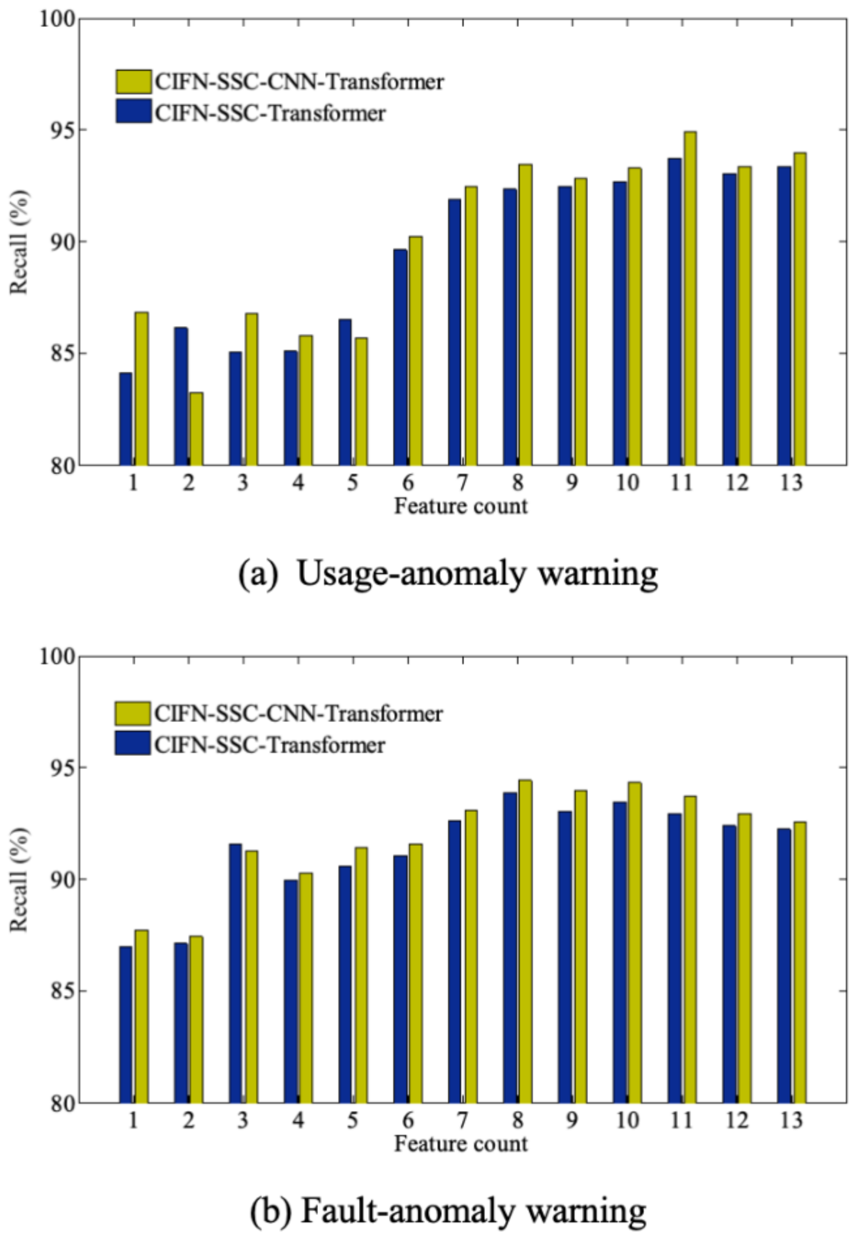

FIGURE 3. Stabilization of abnormal recall with the feature-template size K (evidence for platform default configuration)

(larger K) are enabled only after pilot validation for a given equipment category and data condition (e.g., richer auxiliary channels or more frequent short-cycle transitions). This "default + category optional" strategy aligns with the feature registry and version governance described.

Governance actions and escalation rules (to prevent unnecessary rescheduling). To avoid false triggers causing unnecessary schedule disruptions, abnormal recognition does not directly lock resources. Instead, the platform follows a staged policy:

Alert stage: generate notifications and annotate portrait status as “risk observed”.

Temporary restriction stage: if abnormal evidence persists across multiple windows and/or is corroborated by logs (alarm codes, operator confirmation, maintenance history), mark the instrument as “restricted” and hold conflicting future reservations.

Lock and ticket stage: if confirmed or persistent, mark as “unavailable”, trigger rescheduling to substitute resources, and automatically open a maintenance/inspection ticket with traceable evidence.

This staged mechanism makes abnormal recognition actionable while keeping the scheduling engine rule-based, explainable, and auditable.

Platform output: default K template and controlled release. Based on the stabilization pattern in Figure 3, the platform publishes a default abnormal-recognition template as a baseline configuration (e.g., Template-K_Base v1.0), including the recommended K range, required input channels, and governance rules for escalation. For equipment categories with richer signals or distinct operational cycles, the platform supports category optional templates (e.g., Template-K_Category v1.1), which may increase K only after pilot verification on recent operational logs.

All template releases follow a controlled process: pilot - evaluation on recent logs - approval by the accountable unit (training center/asset management office) - release - monitoring, with a rollback rule if the update increases false abnormal alerts, destabilizes portrait states, or leads to unnecessary rescheduling workload.

VI. CONCLUSION

This paper presents a management-oriented mechanism for sharing and dynamic scheduling. To validate the scheduling engine, a 30-day comparative simulation was conducted. Results indicate that the portrait-driven CTPN model increased the average cross-department sharing rate by 24.5% and reduced scheduling conflicts by 18.2% compared to traditional first-come-first-served (FCFS) methods. Furthermore, the mean fault-response time was shortened by 12 minutes due to the immediate feedback from the status-recognition microservice, confirming that the instrument portrait directly enhances sharing efficiency and governance transparency. By organizing equipment information into governed, auditable dimensions (capability, access and reservation rules, planned restrictions such as maintenance/calibration, utilization records, and performance indicators), the proposed framework improves resource transparency and enables cross-department coordination at the platform level. A rule-based and explainable scheduling engine is used to represent sequencing, concurrency, and conflicts under institutional constraints, supporting operational decision-making that is traceable and accountable for administrators. To enhance adaptability to real operating conditions, the framework allows an operating-status recognition microservice to provide time-stamped evidence for portrait state updates and exception handling, while keeping scheduling decisions policy-governed and auditable. At the application layer, an LLM assistant can be embedded to support portrait querying, rule interpretation, and standardized handling suggestions, improving administrator efficiency without weakening responsibility boundaries.

Future work will focus on: (i) expanding platform governance mechanisms such as incentives and KPI-based evaluation for sharing utilization and service quality; (ii) improving data quality management for heterogeneous equipment categories through category templates, controlled releases, and rollback monitoring; and (iii) conducting multi-department pilot deployments to validate operational impacts on utilization, conflict reduction, exception-handling workload, and user satisfaction.

AUTHOR CONTRIBUTIONS

Xiantu Lin designed, collected and analyzed the data, and drafted the manuscript. Xiantu Lin conducted the study, critically revised the manuscript for important intellectual content, and gave final approval of the version to be published. Xiantu Lin participated fully in the work, take public responsibility for appropriate portions of the content, and agreed to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

CONFLICTS OF INTEREST

The author declares no conflict of interest.

FUNDING

This work was supported by Zhejiang Provincial Higher Vocational Education Teaching Reform Project (Grant No. JGCG2025203GZ).

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] D. H. Siswanto, H. A. Putri, K. Kintoko, and S. Suwajirah, “Chemistry laboratory management: Reviewing planning, organization, and evaluation for effectiveness,” Curricula: Journal of Curriculum Development, vol. 4, no. 1, pp. 229-242, 2025, doi: 10.17509/curricula.v4i1.80139.

[2] A. Ana, D. A. A. Wulandari, R. Mafimudah, et al., The impact of inadequate laboratory personnel (laboratory heads, technicians, and assistants) on biology practicum implementation. BIOMA: Jurnal Ilmiah Biologi. vol. 14, no. 2, pp. 105-120, 2025.

[3] T. Sulistyo, K. Achmad, and I. B. I. Purnama, The Asset Management and Tracking System for Technical and Vocational Education and Training (TVET) Institution Based on Ubiquitous Computing. ComTech: Computer, Mathematics and Engineering Applications. vol. 13, no. 1, pp. 23-34, 2022. doi: 10.21512/comtech.v13i1.7342.

[4] B. Amangeldy, T. Imankulov, N. Tasmurzayev, et al., IoT-Based Unsupervised Learning for Characterizing Laboratory Operational States to Improve Safety and Sustainability. Sustainability. 2025; 17(18): 8340. doi: 10.3390/su17188340.
[5] C. Liu, Perception and sharing optimization mechanism of digital media art interactive device driven by adaptive AI[J]. International Journal for Simulation & Multidisciplinary Design Optimization, 2025, 16. doi: 10.1051/smdo/2025017.
[6] E. Panagiotidou, P. T. Chountalas, A. I. Magoutas, et al., The multifaceted impact of ISO/IEC 17025 accreditation: a sector-specific analysis in civil engineering testing and calibration laboratories. The TQM Journal. vol. 37, no. 4, pp. 998-1035, 2025. doi: 10.1108/TQM-10-2023-0347.
[7] ISO. ISO 55000: 2024 Asset management - Vocabulary, overview and principles. International Organization for Standardization. 2024-07.
[8] T. Mmelesi and K. N. Nwaigwe, A computerised maintenance management system as a teaching aid. World Transactions on Engineering and Technology Education. vol. 18, no. 3, pp. 340-344, 2020.
[9] P. Wagh, S. M. Selvam, B. Paramasivan, et al., Comparative evaluation of biochar produced from laboratory waste media using different thermochemical methods and its potential in plant growth promoting properties. Energy & Environment. vol. 36, no. 5, pp. 2111-2130, 2025. doi: 10.1177/0958305X241293738.
[10] L. Zhang, Y. Liang, and Z. Lin, On the open sharing and promotion strategy of large equipment in colleges and universities. Region - Educational Research and Reviews. 2024; 6(9). doi: 10.32629/rerr.v6i9.2750.
[11] Y. Wang, Z. Wei, J. Cao, and Z. Liu, Research and Implementation of Big Data Technology Laboratory Equipment Reservation Management System. IOP Conf. Ser.: Earth Environ. Sci. 2019; 252: 042072. doi: 10.1088/1755-1315/252/4/042072.
[12] T. A. Strom, G. Haugstad, J. Shu, et al., Shared instrumentation facilities: Benefiting researchers and universities, and sustaining research excellence. MRS Bulletin. 2020; 45: 331-335. doi: 10.1557/mrs.2020.130.
[13] Y. Zhu, Discussion on the construction of shared management platform for large instrument in University. Atlantis Press.
[14] X. Wang and S. Qu, Measuring China's Open Sharing for scientific instruments: an ascendency analysis with benchmark use-time data. Humanities and Social Sciences Communications. 2025; 12: 1501. doi: 10.1057/s41599-025-05767-y.
[15] N. Turšič, et al., Tool Condition Monitoring Using Machine Tool Spindle Current and Long Short-Term Memory Neural Network Model Analysis. Sensors. 2024; 24(8): 2490. doi: 10.3390/s24082490.
[16] C. I. C. Crucho, M. J. S. Gomes, and J. Avó, A role-playing tabletop game on laboratory techniques and chemical reactivity: a game-based learning approach to organic chemistry education. Chemistry Teacher International. vol. 7, no. 3, pp. 455-470, 2025. doi: 10.1515/cti-2025-0063.