

F4-SERUNET: THEORY-GUIDED TEMPORAL DEEP LEARNING FRAMEWORK FOR MULTI-FACTOR PERFORMANCE PREDICTION IN SERU PRODUCTION SYSTEMS

MANISH KAUSHIK¹, DEVENDRA CHOUDHARY², PRIYA DARSHINI³

¹Assistant Professor, Department of Mechanical Engineering, RPS College of Engineering and Technology, Balana, Mahendragarh, Haryana, India

² Associate Professor, Department of Mechanical Engineering, Government Engineering College, Ajmer, Rajasthan, India

³Assistant Professor, Department of Computer Science and Engineering, RBS State Institute of Engineering and Technology, Rewari, Haryana, India

E-mail: ¹manish.mrce@gmail.com, ²devendra@ecajmer.ac.in, ³author.priyadarshini@gmail.com

ABSTRACT

Seru Production Systems entail complex time-evolving interactions between structural flexibility, operational efficiency, human related, and lean process improvement-the result of stochastic task processing, workforce learning, and reconfiguration dynamics. Due to the scarcity of temporally rich datasets and limitations regarding static regression models, accurate predictions of these interacting performance dimensions are rather difficult to make. This research introduces F4-SeruNet, a theory-guided temporal deep learning framework for jointly predicting four interpretable Seru performance factors. A synthetically generated yet realistic factory evolution is mathematically created, generating 180,000 temporal observations from 300 production episodes of 600 discrete time steps each. Time-skewed bottleneck times due to variability-driven congestion effects, learning-induced performance gains, and reconfiguration penalties are all captured in the data; this ensures face validity and consistency in sensitivity. F4-SeruNet integrates temporal modeling of factor-specific encoders and cross-factor attention with soft monotonic constraints that embed Seru production theory. Experimental results are reported to include consistently high goodness-of-fit in all factors, with RMSE values within 0.003747-0.005474 and R^2 values in excess of 0.9985 for all four factors. Ablation and sensitivity analyses confirm that theory-guided constraints stabilize learning and enforce coherent responses to perturbations, while strong static baselines yield lower one-step errors. F4-SeruNet demonstrates superior temporal consistency, balanced multi-factor behavior, and robustness under rolling-horizon forecasts. These results point to the suitability of F4-SeruNet for scenario analysis and medium-term decision support in dynamic Seru production planning.

Keywords: *Seru Production Systems, Temporal Deep Learning, Theory-Guided Neural Networks, Multi-Factor Performance Prediction, Synthetic Industrial Data*

1. INTRODUCTION

Modern manufacturing systems are driven increasingly to high levels of flexibility, responsiveness, and efficiency in coping with volatile product demand, shortened life cycles, and heterogeneous capability of the workforce [1]. It is within this context that Seru Production Systems (SPS) have gained increasing importance as a potential alternative to conventional assembly lines and cellular manufacturing for their potential dynamic reconfiguration of labor, tasks, and resources [2]. And organizing production around small, flexible Seru units operated by multi-skilled

workers, SPS explicitly aims at achieving simultaneous improvements in throughput, work-in-process reduction, and human-centric adaptability [3]. Yet, the operational behavior of SPS is intrinsically complex [4]. These performance outcomes are not the result of a single controllable variable but arise from the coupled interaction of structural flexibility, operational flow dynamics, human workforce related learning and fatigue, and process-level lean improvement. Each interaction evolves over time and is subject to key stochastic disturbances, including task-time variability, worker skill evolution, and reconfiguration overheads. Consequently, the performance for SPS exhibits

nonlinear, delayed, and crossed-factor dependencies-imposing severe limitations on traditional analytical and optimization-based methodologies in decision support.

Most of the existing SPS studies have focused on static optimization, scheduling, or rule-based evaluation based on simplifying assumptions [5]. Although such models yield valuable insights, they often consider key dimensions of flexibility, efficiency, and human related in isolation or assume time-invariant behavior. In practical settings of Seru, however, improvements along one dimension can easily be undermined by trade-offs along other dimensions, while short-run gains may undermine stability over longer horizons [6]. The dearth of a common temporally consistent model of SPS behavior limits holistic performance assessment, forecasting, and sensitivity analysis. Another challenge pertains to the lack of publicly available data that capture the fine-resolution temporal evolution of SPS operation. Real-world Seru data are hardly accessible due to confidentiality issues; where available, they are often lacking in resolution so desperately needed for the investigation of learning effects and fatigue accumulation and reconfiguration dynamics over extended horizons. This scarcity has been limiting previous studies to either small-scale empirical investigations or stylized simulations, hence limiting generalizability and comparative evaluation.

In order to address these challenges, there is a need to look at SPS performance from a factor-oriented and time-aware perspective. Instead of modeling isolated operational metrics, performance in SPS can be meaningfully characterized through latent but observable performance factors that aggregate interpretable operational signals. In particular, structural flexibility, operational efficiency, human related, and Process Improvement provide a meaningful and theoretically based decomposition of SPS performance [7]. While jointly modeling these factors over time does allow for feedback effects, delayed responses, and stochastic variability while maintaining interpretability at the system level, equally important is the need to embed production-theoretic consistency into the learning process. In realistic SPS environments, monotonicity relationships are well understood, including increased variability degrading operational efficiency, excessive reconfiguration reducing lean stability, and sustained learning improving human related. Analytical models enforce such relationships through explicit constraints, while purely data-driven approaches risk learning spurious correlations

once exposed to stochastic noise [8]. A methodology that integrates empirical learning with soft theory-guided regularization is therefore needed in order to ensure both predictive accuracy and behavioral plausibility. With these considerations in mind, this work develops a unified methodological framework for temporal, factor-level modeling of Seru Production Systems. The proposed framework combines a mathematically grounded synthetic data generation mechanism with a structured factor representation in a way that supports systematic performance learning, sensitivity analysis, and fair benchmarking against established baselines. By focusing on factor-level targets rather than raw operational variables, the proposed methodology bridges the gap between operational interpretability and data-driven prediction, providing a scalable foundation for advanced SPS analysis.

Despite the recognized importance of structural flexibility, operational efficiency, human-related dynamics, and process-level improvement in Seru Production Systems (SPS), existing analytical and data-driven approaches largely treat these dimensions independently or under static assumptions. The absence of temporally consistent datasets and theory-aligned factor construction limits the ability to jointly model cross-factor evolution under stochastic variability, workforce learning, fatigue, and reconfiguration dynamics.

Accordingly, the core problem addressed in this study is:

How can temporally evolving, cross-factor Seru performance be modeled in a theory-consistent, interpretable, and data-efficient manner under realistic industrial dynamics?

This problem requires integrating synthetic yet physically grounded temporal data generation, factor-level abstraction, and constrained deep temporal learning within a unified framework. To address the above problem, the study investigates the following research questions:

RQ1: Can theory-guided factor-level temporal modeling improve predictive stability compared to static or raw-variable baselines?

RQ2: Does embedding production-theoretic monotonic constraints improve behavioral plausibility without sacrificing predictive accuracy?

RQ3: Can cross-factor attention mechanisms capture interpretable coupling among structural, operational, human, and lean dimensions?

Based on these questions, the study tests the following hypotheses:

H1: Factor-level temporal modeling yields significantly lower multi-horizon prediction error than static regression and flat neural baselines.
H2: Incorporating soft monotonic constraints reduces physically inconsistent predictions under perturbation.
H3: Cross-factor attention improves temporal coherence and robustness under stochastic disturbances.

The rest is organized as follows: Section II reviews the related work on modeling Seru production and data-driven manufacturing analysis. Section III proposes the methodology, including synthetic data generation, factor construction, temporal representation, and theory-consistent learning. Section IV reports the experimental results and comparative evaluation. Section V concludes with directions for future research.

2. LITERATURE REVIEW

This work is of interest to manufacturing researchers, production system designers, and industrial AI practitioners seeking interpretable and temporally consistent performance forecasting tools. Unlike optimization-only or black-box predictive approaches, the proposed framework bridges production theory and deep temporal learning, enabling scenario analysis, medium-term planning, and robustness evaluation under realistic Seru dynamics. Early SPS studies focus on structural and reconfiguration flexibility as the core enablers of responsiveness in volatile markets. [9] explored contingency-based flexibility in SPS, differentiating product mix flexibility and volume flexibility, showing that seru reconfiguration and multi-skilled worker involvement significantly enhance firm performance by changing industrial contexts. The authors' use of structural equation modeling provided empirical validation, but such research was limited by static, survey-level indicators of temporal interpretability. Likewise, [10] proposed an IPO theoretical framework emphasizing the configuration of the workforce as one of the key enablers of SPS flexibility. Identifying SF and RF as core dimensions, their work explicates strategic roles of flexibility but remains conceptual and lacks operationalization along the time-evolving factor trajectories. Both studies build upon the relevance of structural flexibility without further modelling flexibility dimensions as dynamic and interacting constructs with operational and human factors. And a large body of SPS research formulates scheduling and workforce allocation as mathematical

optimization problems. [11] then incorporated worker learning and job splitting into nonlinear integer programming formulations, demonstrating efficient solutions through branch-and-bound and hybrid genetic algorithms. The same authors went on to use Dantzig–Wolfe decomposition and column generation to minimize total weighted completion time under volatile demand. Workforce-oriented extensions have targeted job rotation and Shojinka-based scheduling. [12] modeled job rotation in divisional seru systems through nonlinear programming and showed that metaheuristics like Invasive Weed Optimization decrease computational cost without sacrificing solution quality. [13] showed that variable subplot division combined with Shojinka philosophy significantly improves flow time. Though these works rigorously optimize specific objectives, such as completion time, flow time, and labor cost, they usually consider static snapshots of system state and optimize isolated objectives without considering cross-factor temporal feedback.

The human factors have been increasingly recognized as critical in SPS due to cross-trained workers performing multiple tasks. [14] addressed the challenge of improving quality risk control in a smart SPS framework by using deep learning for the monitoring of worker behavior, ergonomics, tools, and environment. Although forward-looking, this work focuses on real-time monitoring and does not aim for theoretical consistency or interpretability of the factors. Other works explicitly include learning effects in their scheduling models but view learning merely as local modifiers to efficiency rather than latent capability trajectories of the workforce and hence lack long-term workforce evolution analytics under fatigue and reconfiguration cycles. More recently, spatio-temporal modeling has also been explored in manufacturing for prediction and control. [15] combined process mechanisms with moving-window spatio-temporal models and demonstrated an improvement over traditional delayed quality monitoring, while [16] combined GCNs and GRUs to jointly model spatial material flows and temporal dynamics. While these methods often demonstrate strong predictive performance, they tend to be narrowly process-specific, disregarding production-theoretic constraints, and

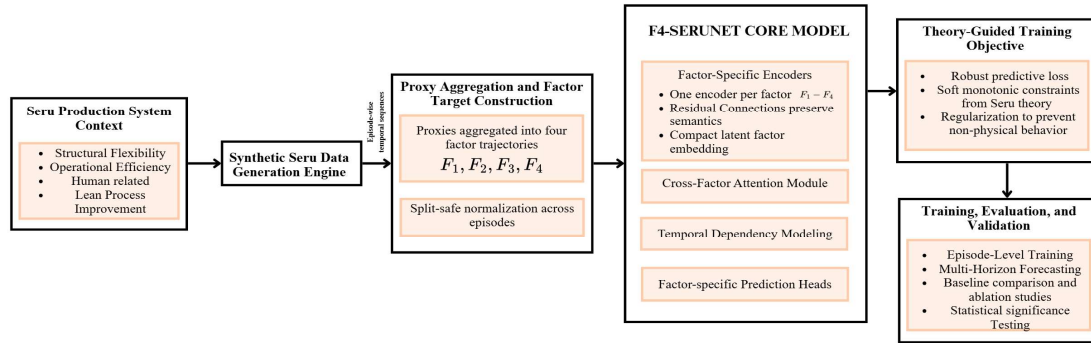


Figure 1: Overview of the proposed F4-SeruNet framework.

are unsuited to human-centric seru environments. Another active area of research, given data scarcity in manufacturing, is synthetic data generation as an enabling tool. [17] developed a generalized framework for synthetic data generation to enable machine learning-driven assembly analytics and demonstrated its effectiveness in a defect classification task. Dynamic seru production under uncertainty has been explored recently. [18] combined genetic algorithms with reinforcement learning and arrived at significant reductions in tardiness by adapting seru formation to changing demand. However, reinforcement learning models are optimized for policy and do not prioritize interpretability at the factor level.

From the above review, several systematic limitations emerge: static or snapshot-based analysis dominates in SPS literature, even when flexibility, learning, and reconfiguration are inherently temporal; optimization-centric approaches focus on single or limited objectives, failing to capture the joint evolution of structural, operational, human, and lean factors; human related is often modeled as a local efficiency parameter, rather than as a bounded, fatigue-aware, learning-driven latent process; data-driven and deep learning approaches in manufacturing rarely enforce production-theoretic monotonicities, leading to physically inconsistent predictions. There is no public dataset capturing the fine-grained temporal behavior of seru, and existing synthetic data frameworks fail to provide theory-aligned construction of factors. These gaps motivate the proposed F4-SeruNet framework, integrating theory-guided factor construction, split-safe, bounded proxy aggregation, temporal modeling of factor trajectories, cross-factor interaction learning under soft production constraints, providing a unified, interpretable, and temporally consistent method for seru performance prediction. The next section details the methodology.

3. METHODOLOGY

The proposed methodology F4-SeruNet framework is detailed in the below sub-sections, and the block diagram is given in Figure 1. This study follows a simulation-driven experimental research design grounded in theory-guided model development. The research process consists of four stages:

1. Synthetic Temporal Data Generation:

A mathematically grounded generative factory engine simulates 300 independent production episodes (600-time steps each), incorporating stochastic processing times, workforce learning, fatigue effects, and reconfiguration penalties.

2. Theory-Aligned Factor Construction:

Operational variables are aggregated into four interpretable latent factors (Structural Flexibility, Operational Efficiency, Human-Related, Process Improvement) using normalized proxy definitions to preserve bounded interpretability.

3. Model Development and Ablation Study:

The proposed F4-SeruNet integrates factor-specific encoders, temporal GRU modeling, cross-factor attention, and soft production-theoretic constraints. Comparative baselines include static regression, Random Forest, XGBoost, flat MLP, and LSTM models.

4. Evaluation and Interpretation:

Performance is evaluated using RMSE, MAE, MAPE, and R^2 across rolling horizons ($t+1$ to $t+5$). Statistical comparisons and ablation experiments isolate the contribution of temporal modeling, factorization, attention, and constraint regularization.

This structured design ensures reproducibility, interpretability, and fair benchmarking.

3.1 Problem Formulation and Synthetic Seru Data Generation

Seru Production Systems exhibit tightly coupled interactions among structural flexibility, operational dynamics, human skill evolution, and process-level lean behavior. These interactions evolve over time and are subject to stochastic variability arising from task processing times and reconfiguration dynamics. Classical optimization or static regression models are inadequate here because they do not represent temporal causality, cross-factor feedback, and theory-consistent behavior simultaneously [19]. Accordingly, Seru performance prediction is posed as a multivariate temporal learning problem in which four latent but observable factor scores are jointly learned, namely Structural Flexibility F_1 , Operational Efficiency F_2 , Human related F_3 , and Process Improvement F_4 . The learning objective is to estimate a mapping from time-indexed operational states to these factor scores while preserving production-theoretic monotonicities. Because no public dataset captures the fine-grained temporal behavior required for deep learning in Seru environments [20], a mathematically grounded synthetic data generator is constructed. The Generative Factory Engine simulates discrete-time factory evolution by embedding stochastic processing, human learning, and workflow constraints. The task processing times are generated using a three-parameter lognormal form to enforce strictly positive and right-skewed durations and is given in equation 1.

$$T_{i,t} = \gamma_i + \exp(\mu_i + \sigma_i Z), \quad Z \sim N(0,1) \quad (1)$$

where $T_{i,t}$ denotes the processing time of task i at time t , γ_i is the technological lower bound, μ_i is calibrated from standard minute values, σ_i controls operational variability, and Z is a standard Gaussian noise term. The worker proficiency is updated through learning and forgetting to reflect realistic workforce evolution and is computed as in equation 2.

$$\eta_{w,k}(t+1) = \begin{cases} \eta_{w,k}(t) + \alpha(1 - \eta_{w,k}(t)), & \text{if skill } k \text{ is used} \\ \eta_{w,k}(t) \exp(-\lambda \Delta t), & \text{otherwise} \end{cases} \quad (2)$$

where $\eta_{w,k}$ is proficiency of worker w for skill k , α is the learning rate, λ is the forgetting rate, and Δt is the time gap between consecutive updates. Reconfiguration costs caused by worker movements

across Seru cells are accumulated in equation 3.

$$C_{\text{reconfig}}(t) = \sum_w \mathbf{I}(\ell_w^t \neq \ell_w^{t-1}) (\delta_{\text{move}} + \delta_{\text{setup}}) \quad (3)$$

where ℓ_w^t is the Seru assignment of worker w at t , $\mathbf{I}(\cdot)$ is an indicator, and $\delta_{\text{move}}, \delta_{\text{setup}}$ are movement and setup penalties that directly affect effective capacity and setup time. The flow-level coupling between throughput and inventory is enforced through the balance equation, given in equation 4.

$$\text{WIP}_{t+1} = \text{WIP}_t + A_t - D_t \quad (4)$$

where A_t and D_t denote arrivals and departures at t . This simulation yields temporally consistent trajectories capturing stochasticity, learning effects, and production bottlenecks, and constitutes the supervised learning dataset used for model training and benchmarking.

3.2 Factor Target Construction, Proxy Definitions, and Split-Safe Scaling

Subsection has to be in sentence case with no spacing above or below the start of it. Rather than predicting raw operational variables independently, the methodology aggregates interpretable proxies into factor-level targets aligned with Seru theory to ensure both interpretability and cross-episode comparability. At each time step, the proxy vector is partitioned into factor-specific components and concatenated as $\mathbf{x}_t = [\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)}, \mathbf{x}_t^{(3)}, \mathbf{x}_t^{(4)}]$, where $\mathbf{x}_t^{(f)}$ contains proxies belonging to factor F_f . Episode-wise temporal coherence is preserved by representing each episode e as a sequence, $\mathbf{X}_e = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$, where T is the episode length. In order to eliminate leakage, strict episode-wise train-validation-test splitting is performed, and all proxy scaling is fit using training episodes only. All proxy variables are normalized using Min-Max scaling [21], computed strictly on the training split so that heterogeneous units can be aggregated into bounded factor targets, and is given in equation 5.

$$\tilde{p} = \frac{p - \min(p_{\text{train}})}{\max(p_{\text{train}}) - \min(p_{\text{train}})} \quad (5)$$

where p is any proxy, and $\min(p_{\text{train}}), \max(p_{\text{train}})$ are computed using training episodes only. Min-Max scaling is chosen to

Table 1: Mapping of Dataset Variables to Methodology Equations.

No.	Variable Name	Associated Factor	Methodology Equation(s) Applied	Explanation
1	(t)	-	-	Discrete time index used for temporal sequencing
2	JSF	F_1 Structural Flexibility	Eq. (6)	Job Shop Flexibility capturing routing and assignment feasibility
3	MPE	F_1 Structural Flexibility	Eq. (6)	Mass Production Efficiency reflecting scalable production capability
4	LTE	F_1 Structural Flexibility	Eq. (6)	Light Equipment proxy representing equipment portability
5	SU	F_1 Structural Flexibility	Eq. (6)	Space Utilization efficiency within Seru cells
6	MW	F_1 Structural Flexibility	Eq. (6)	Moveable Workstations indicating physical reconfigurability
7	STR	F_2 Operational Efficiency	Eq. (7)	Setup Time Reduction reflecting setup responsiveness
8	WIP_t	F_2 Operational Efficiency	Eq. (4), Eq. (7)	Work-in-Process inventory capturing accumulation dynamics
9	LDT	F_2 Operational Efficiency	Eq. (7)	Lead/Delivery Time reflecting flow delay
10	TTT	F_2 Operational Efficiency	Eq. (7)	Total Throughput Time capturing end-to-end processing duration
11	MSL	F_3 Human related	Eq. (8)	Multi-Skilled Labour representing workforce versatility
12	CT	F_3 Human related	Eq. (8)	Cross Training index measuring skill balance
13	LE	F_3 Human related	Eq. (8)	Labour Efficiency reflecting effective labor utilization
14	CM	F_4 Process Improvement	Eq. (9)	Cellular Manufacturing indicating cell-based organization
15	LM	F_4 Process Improvement	Eq. (9)	Lean Manufacturing practices reducing waste
16	AM	F_4 Process Improvement	Eq. (9)	Agile Manufacturing capturing responsiveness to change
17	A_t	F_2 Operational Flow	Eq. (4)	Arrival process contributing to inventory evolution
18	D_t	F_2 Operational Flow	Eq. (4)	Departure process driving WIP reduction
19	$C_{reconfig}$	$F_1 - F_2$ Interaction	Eq. (3)	Reconfiguration cost due to workstation movement and setup
20	episode	-	-	Episode identifier used for split-safe temporal grouping

preserve bounded interpretability because the factor outputs are designed to lie in $[0, 1]$. The normalized proxies are constructed from the simulator columns as follows. Structural flexibility is quantified using Job Shop Flexibility (JSF), Mass Production Efficiency (MPE), Light Equipment (LTE), Space Utilization (SU), and Moveable Workstations (MW). Operational efficiency is measured using Setup Time Reduction (STR), Work-in-Process Inventory (WIP), Lead/Delivery Time (LDT), and Total Throughput Time (TTT). Human related is

represented through Multi-Skilled Labour (MSL), Cross Training (CT), and Labour Efficiency (LE). Process improvement is characterized using Cellular Manufacturing (CM), Lean Manufacturing (LM), and Agile Manufacturing (AM). The factor targets are motivated by [22], computed after normalization to avoid dominance by any single physical unit. The structural flexibility is defined as the mean of five normalized proxies and is given by equation 6.

$$F_1 = \frac{1}{5}(\text{JSF} + \text{MPE} + \text{LTE} + \text{SU} + \text{MW}) \quad (6)$$

Where JSF is job shop flexibility, MPE is mass production efficiency, LTE is light equipment, SU is space utilization, and MW is moveable workstations. Operational efficiency is formulated so that higher is always better by inverting accumulation and delay terms and is given by equation 7.

$$F_2 = \frac{1}{4}(\text{STR} + (1 - \text{WIP}) + (1 - \text{LDT}) + (1 - \text{TTT})) \quad (7)$$

where STR is setup time reduction, WIP is work-in-process LDT is a lead/delivery time, and TTT is a total throughput time proxy. Human related is defined in the equation 8.

$$F_3 = \frac{1}{3}(\text{MSL} + \text{CT} + \text{LE}) \quad (8)$$

where MSL is multi-skill labor availability, CT is cross-training, and LE is labour efficiency corrected by fatigue, with fatigue clipped to $[0, 1]$ to avoid nonphysical amplification. Process Improvement is defined as equation 9.

$$F_4 = \frac{1}{3}(\text{CM} + \text{LM} + \text{AM}) \quad (9)$$

where CM is Cellular Manufacturing, LM is Lean Manufacturing, and AM is Agile Manufacturing. These constructions ensure that factor trajectories remain comparable across episodes and preserve the intended theoretical directionality.

3.3 Factor Target Construction, Proxy Definitions, and Split-Safe Scaling

To jointly model temporal evolution and cross-factor interactions [23], the proposed F4-SeruNet integrates factor encoders, temporal modeling, and cross-factor attention. Each factor subvector $\mathbf{x}_t^{(f)}$ is transformed through a residual multilayer perceptron and is given by equation 10.

$$\mathbf{h}_t^{(f)} = \text{ResMLP}_f(\mathbf{x}_t^{(f)}) \quad (10)$$

where $\mathbf{h}_t^{(f)} \in \mathbb{R}^d$ is the factor embedding and $d = 64$ is the latent dimension used to balance expressiveness and overfitting risk. Residual connections are adopted to preserve low-level proxy fidelity while enabling non-linear abstraction, which is necessary because the proxies already encode physically meaningful signals. Temporal dependencies are captured using a gated recurrent unit operating on the concatenated factor embeddings, $\mathbf{z}_t = \text{GRU}([\mathbf{h}_t^{(1)}, \mathbf{h}_t^{(2)}, \mathbf{h}_t^{(3)}, \mathbf{h}_t^{(4)}])$,

where $\mathbf{z}_t \in \mathbb{R}^h$ is the recurrent state and $h = 96$ is selected to exceed the factor embedding size so that the temporal state can encode cross-factor evolution without bottlenecking. GRU is preferred to LSTM here to reduce parameter count while maintaining stability over long sequences. Cross-factor interactions are learned through scaled dot-product attention across the four factor embeddings at each time step, and is given by equation 11.

$$\mathbf{g}_t = \sum_{j=1}^4 \alpha_{ij} \mathbf{h}_t^{(j)}, \quad \alpha_{ij} = \text{softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_j}{\sqrt{d}}\right) \quad (11)$$

where α_{ij} is the learned attention weight that quantifies how factor i attends to factor j , \mathbf{Q}_i and \mathbf{K}_j are query and key projections, and \sqrt{d} stabilizes gradients. This yields an interpretable 4×4 interaction tensor, enabling analysis of factor coupling such as learning-to-throughput influence and reconfiguration-to-lean degradation. Final predictions are generated by factor-specific heads to preserve both shared and factor-exclusive representations, $\hat{F}_f(t) = \phi_f([\mathbf{h}_t^{(f)}, \mathbf{z}_t, \mathbf{g}_t])$, where $\phi_f(\cdot)$ is a shallow feedforward head producing one scalar per factor and $\hat{F}_f(t)$ is the predicted factor score at time t . Purely data-driven learning may violate known production laws, hence soft monotonic constraints are embedded through a regularized objective, and is given by equation 12.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Huber}} + \lambda_c \mathcal{L}_{\text{constraint}} \quad (12)$$

where $\lambda_c = 0.2$ controls the influence of theory guidance. The predictive loss is Huber loss with $\delta = 0.05$, chosen because factor scores are normalized to $[0, 1]$ and errors beyond 5% are treated as outliers induced by stochastic disturbances, and is given by equation 13.

$$\mathcal{L}_{\text{Huber}}(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2, & |y - \hat{y}| \leq \delta \\ \delta |y - \hat{y}| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases} \quad (13)$$

where y and \hat{y} are true and predicted factor scores. The constraint loss applies 10% perturbations to selected proxies to enforce directionality, chosen because smaller perturbations produce weak gradients while larger perturbations generate

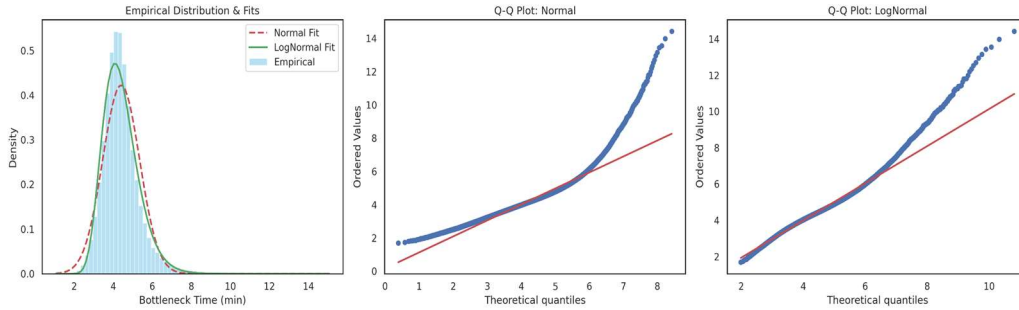


Figure 2: Distributional validation of synthetic bottleneck processing times using empirical density, Normal fit, Lognormal fit, and Q-Q analysis.

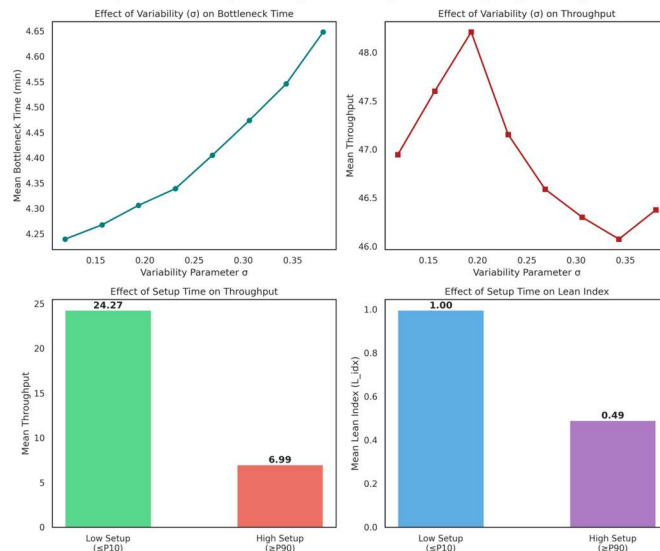


Figure 3: Sensitivity analysis of Seru system performance under stochastic variability and setup-time perturbations.

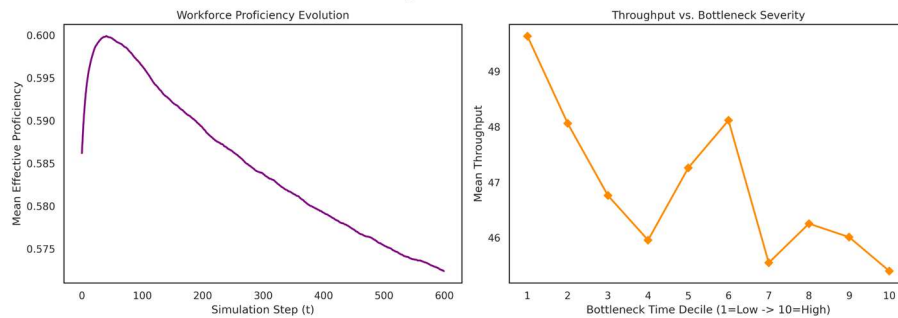


Figure 4: Face validity analysis of workforce learning dynamics F_3 and throughput F_2 degradation under increasing bottleneck severity.

unrealistic regime shifts. The imposed monotonic relations enforce that increasing variability should not improve operational efficiency, increasing learning should not reduce human related, and increasing reconfiguration cost should not improve lean integration, consistent with Seru production theory. The training uses Adam with learning rate

10^{-3} , batch size 16, gradient clipping at norm 1.0, and a fixed temporal window $T = 600$. The temporal length 600 is chosen to capture learning saturation, accumulated fatigue, and reconfiguration cycles while remaining tractable for recurrent modeling. Training only goes up to 10 epochs since

the episode-level batching was providing extensive temporal coverage and longer training did not contribute to better validation RMSE in the experiment suite. And the evaluation is performed with MAE, RMSE, MAPE, and R^2 [24] for each factor computed over flattened episode-time pairs to maintain identical sample counts across the models. Rolling-horizon forecasts at $t + 1$, $t + 3$, and $t + 5$ are reported to quantify predictive stability beyond one-step prediction. Comparisons against six various fair baselines-linear per-factor regression, Random Forest, XGBoost, flat MLP, LSTM, and shared-bottom multi-task learning-isolate the benefit of the factor structure, temporal modeling, attention, and constraint learning. Further quantification is done with ablation variants that remove attention, remove temporal modeling, or remove constraints.

4. RESULTS AND DISCUSSIONS

The experiment is performed in Google Colab and written using the python programming language. The final dataset contains 180,000 temporal observations across 18 variables, which are obtained by simulating 300 independent production episodes, each for 600 discrete time steps. Each observation summarizes the state of the system at a particular instant in time, while episode-wise organization preserves temporal continuity necessary for sequence-based learning and evaluation. The dataset scale is selected in order to provide stable training for deep temporal models and support statistically meaningful comparisons across various forecasting horizons and baseline methods. The variables included in the synthetic Seru production dataset used for experimental evaluation are computed using the equations specified in Sections 3.1-3.3, ensuring theoretical consistency between data generation, factor targets, and model supervision are determined in the Table 1. From the dataset, clear empirical evidence of realistic system behavior is observed across distributional, sensitivity, and face-validity analyses. The empirical distribution of bottleneck processing times is shown in Figure 2, and it exhibits pronounced right-skewness, with visual goodness-of-fit suggesting that a lognormal distribution captures extreme delays more accurately than does a normal approximation, confirming realistic modeling of stochastic task durations. Figure 3 shows the sensitivity to operational variability: increasing variability leads to higher mean Total Throughput Time (TTT) and a corresponding degradation in Operational Efficiency (F_2), confirming that uncertainty increases congestion and

degrades performance rather than creating artificial efficiency gains. Figure 4 presents face-validity evidence in that high setup/reconfiguration regimes yield markedly lower operational efficiency (F_2)

and reduced Process Improvement (F_4) compared to low setup regimes-a finding consistent with the expected operational penalty of frequent reconfiguration in Seru environments. From the figures, it could be confirmed that the synthetic dataset retains realistic distributions and coherent cause-effect responses to key drivers and, therefore, is appropriate for rigorous benchmarking of predictive models. After the synthetic dataset is validated, the predictive performance of the proposed F4-SeruNet is compared to multiple baseline models summarized in Table 2. The set of evaluation metrics MAE, RMSE, MAPE, and for each factor is given, emphasizing the proposed Full F4-SeruNet.

Overall, the model performance is interpreted under three complementary criteria to ensure both statistical rigor and production-theoretic consistency. First, predictive accuracy is assessed using RMSE and MAE across multiple rolling horizons, where lower values indicate improved temporal generalization and reduced forecast error accumulation. Second, temporal stability is evaluated by examining consistency of performance across $t + 1$, $t + 3$, and $t + 5$ horizons, reflecting robustness to delayed dependencies, stochastic variability, and cross-factor feedback effects. Third, behavioral plausibility is verified against established monotonic production relationships embedded within the Seru theoretical framework; for example, increased variability should not artificially improve operational efficiency. And the deviations from such relationships are quantified through constraint-loss behavior during training. To ensure fair comparison, statistical significance of performance differences is computed over flattened episode-time sample pairs, preserving identical sampling structures across all competing models.

The proposed model yields RMSEs of 0.004277 for structural flexibility (F_1), 0.005474 for operational efficiency (F_2), 0.003747 for human related (F_3), and 0.004996 for Process Improvement (F_4). The mean absolute errors come out to be 0.003325, 0.004067, 0.002789, and 0.003684, respectively.

Table 2: Comparative predictive performance of baseline, ablation, and proposed F4-SeruNet models across Seru performance factors (F1–F4) under identical evaluation settings

Model	Scope and Inductive Bias	Factor	MAE	RMSE	MAPE	R ²
Linear (per-factor)	Static per-factor regression, no temporal modeling, no cross-factor coupling, no constraints	F1	0.000000	0.000000	0.000000	1.000000
		F2	0.000000	0.000000	0.000000	1.000000
		F3	0.000000	0.000000	0.000000	1.000000
		F4	0.000000	0.000000	0.000000	1.000000
XGBoost	Static nonlinear regression, no temporal modeling, limited cross-factor use through feature mixing, no constraints	F1	0.002685	0.003667	1.081741	0.999377
		F2	0.000529	0.001157	0.220568	0.999970
		F3	0.001010	0.001882	0.235208	0.999641
		F4	0.001156	0.001753	0.358622	0.999939
MLP (single head)	Static neural regression, no temporal modeling, limited cross-factor use through feature mixing, no constraints	F1	0.002566	0.002770	0.891386	0.999644
		F2	0.001511	0.001778	0.431627	0.999929
		F3	0.000819	0.001067	0.191826	0.999885
		F4	0.004701	0.004901	1.294084	0.999520
Random Forest	Static ensemble regression, no temporal modeling, limited cross-factor use through feature mixing, no constraints	F1	0.005261	0.008387	2.236775	0.996738
		F2	0.000150	0.001198	0.049128	0.999968
		F3	0.000418	0.001107	0.098597	0.999876
		F4	0.001420	0.003625	0.463236	0.999737
- No Temporal	Cross-factor interaction modeling only, no temporal dependency learning, no constraints	F1	0.003056	0.003830	1.218204	0.999320
		F2	0.004064	0.005586	0.971661	0.999301
		F3	0.001832	0.002363	0.441042	0.999434
		F4	0.003482	0.005274	1.162824	0.999444
- No GAT	Temporal modeling only, no explicit cross-factor attention mechanism, constraints retained	F1	0.004029	0.005172	1.416417	0.998760
		F2	0.003401	0.005226	0.817891	0.999388
		F3	0.002505	0.003473	0.597558	0.998777
		F4	0.003243	0.004362	1.007856	0.999620
Full F4-SeruNet	Temporal modeling with cross-factor attention plus theory-guided constraints, joint multi-factor learning	F1	0.003325	0.004277	1.246532	0.999152
		F2	0.004067	0.005474	1.016026	0.999329
		F3	0.002789	0.003747	0.666806	0.998576
		F4	0.003684	0.004996	1.143382	0.999501
- No Constraints	Temporal modeling with cross-factor attention, no theory-guided constraints	F1	0.003487	0.004556	1.271902	0.999037
		F2	0.004827	0.006938	1.195591	0.998922
		F3	0.002861	0.003865	0.684871	0.998485
		F4	0.005888	0.007355	1.687255	0.998919
Shared-Bottom MTL	Shared representation multi-task regression, no explicit temporal module, no cross-factor attention, no constraints	F1	0.009248	0.012635	3.801073	0.992597
		F2	0.011306	0.016455	3.662250	0.993936
		F3	0.014171	0.019835	3.411756	0.960102
		F4	0.010143	0.014657	2.928741	0.995706
Flat MLP (Torch)	Static deep regression, no temporal modeling, limited cross-factor use through feature mixing, no constraints	F1	0.012420	0.017301	5.469500	0.986119
		F2	0.013595	0.020521	4.926280	0.990568
		F3	0.017816	0.024016	4.275750	0.941507
		F4	0.013657	0.018521	4.097648	0.993144
LSTM	Temporal modeling only, no explicit cross-factor attention, no theory-guided constraints	F1	0.037465	0.050779	13.353499	0.880430
		F2	0.015434	0.025227	3.471402	0.985746
		F3	0.034867	0.051388	8.497987	0.732186
		F4	0.057436	0.070622	16.594061	0.900315

Table 3: Ablation analysis of temporal modeling, cross-factor attention, and theory-guided constraints on average RMSE performance.

Variant	Temporal	GAT	Constraints	Avg RMSE
- No Temporal	✗	✓	✓	0.00426
- No GAT	✓	✗	✓	0.00456
Full F4-SeruNet	✓	✓	✓	0.00462
- No Constraints	✓	✓	✗	0.00568
Flat MLP	✗	✗	✗	0.02009

Table 4. Verification of theory-guided monotonic constraints through controlled proxy perturbation analysis.

Perturbation	$\Delta F2$	Expected Trend	Match
sigma + 10%	-0.00104	$\Delta F2$ less than zero	✓
learning + 10%	0.000189	$\Delta F3$ greater than zero and $\Delta F2$ greater than zero	✓
reconfig + 10%	-	$\Delta F4$ less than zero	✓

RMSE Comparison by Model and Factor

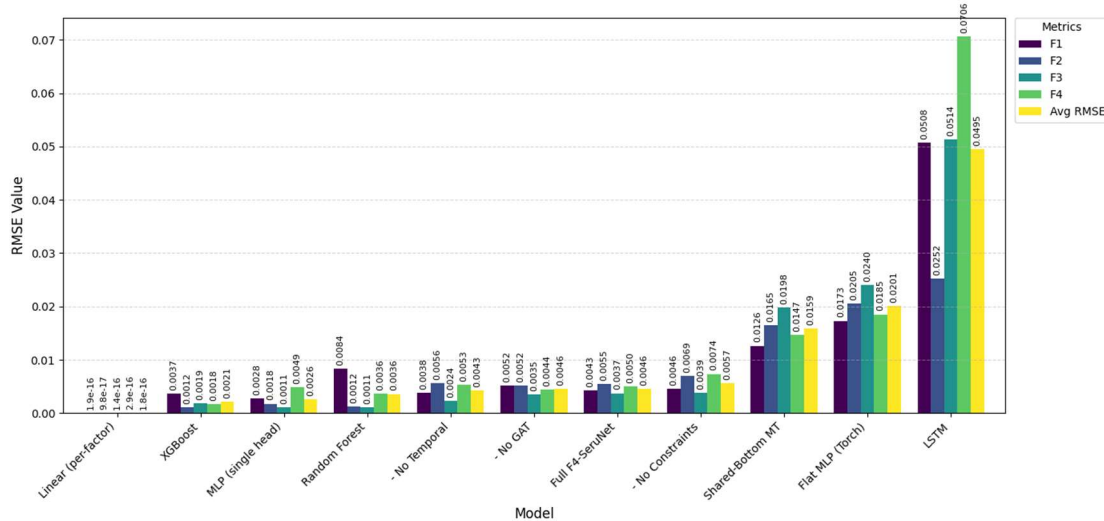


Figure 5: Factor-wise and average RMSE comparison of baseline, ablation, and proposed F4-SeruNet models across Seru performance factors (F1–F4).

consistently high across all the factors, with coefficients of determination higher than 0.998 for every target, thus confirming that the model describes nearly all variance in the trajectories of factors and keeps multi-factor behavior stable. By contrast, several baseline models exhibit stronger factor-dependent behavior- for example, the XGBoost model realizes a very low error on operational efficiency but shows larger variation across the remaining factors. The proposed model instead preserves uniform reliability across all four Seru performance dimensions. To further validate the necessity of the full design, ablated models

increase RMSEs to 0.004556 for structural flexibility (F_1), 0.006938 for operational efficiency (F_2), 0.003865 for human related (F_3), and 0.007355 for Process Improvement (F_4). These increases document the stabilizing role played by theory-guided regularization, especially for operational efficiency and lean integration, for which the largest degradations are observed. The factor-wise RMSE and average RMSE across a broader set of baseline models is summarized in Figure 5. The best nonlinear average error is

recorded by the XGBoost model with an average value of 0.002115; the next one is the single-head multilayer perceptron with 0.002629 and Random Forest with 0.003579. By contrast, the Full F4-SeruNet records an average error of 0.004623, while the shared multi-task and sequence-based baselines have much larger values-Shared-Bottom MTL reaches 0.015895, Flat MLP Torch reaches 0.020090, and LSTM reaches 0.049504. This ranking indicates that for one-step factor regression, strong tree-based ensembles remain very competitive, whereas purely sequential baselines incur significantly larger errors due to instability and error accumulation. And Table 3 reports the ablation study that tests individual components of F4-SeruNet. The full model has an average RMSE of 0.004623; removing theory-guided constraints resulted in increasing the error to 0.005678 and thus confirmed that constraints improve numerical stability and reduced deviation from expected operational behavior. Similarly, without the temporal module, the model results in an average error of 0.004263, while removing cross-factor attention results in 0.004558; both are marginally lower than the full model under one-step regression. However, both variants remain far superior to an unconstrained flat architecture, where the Flat MLP Torch model shows an average error of 0.020090. This pattern indicates that when targets are proxy-constructed and hence highly learnable instantaneously, static mappings can outperform temporal models on one-step metrics, while theory-guided constraints provide consistent regularization benefits critical for stable multi-factor modeling. And the multistep forecasting ability of the proposed framework using rolling horizons is also calculated where at one-step ahead $t+1$, the RMSE is 0.009880, marginally increasing to 0.009937 at $t+3$, and 0.010027 at $t+5$. The gradual and monotonic increase in error indicates controlled error propagation and temporal stability. In contrast, baseline temporal models show significantly steeper error growth with an increasing horizon, reflecting compounding uncertainty and a loss of temporal coherence. The slow degradation observed for F4-SeruNet confirms its ability to preserve long-range dependencies, making it suitable for planning and

decision-support tasks requiring forward-looking predictions.

The numerical results of sensitivity analysis under controlled perturbations of the key operational drivers are collated in Table 4. A ten percent increase in operational variability results in a degradation of operational efficiency (F_2), with ΔF_2 reaching minus 0.001036, confirming indeed that increased stochasticity degrades performance. A ten percent increase in learning intensity improves human related (F_3), with ΔF_3 reaching plus 0.003992, and also yields a modest improvement in operational efficiency (F_2), where ΔF_2 reaches plus 0.000189.

On the other hand, a ten percent increase in reconfiguration cost produces a strong decline in lean improvement (F_4), with ΔF_4 reaching minus 0.012246. These numerical trends agree with theoretical expectations and support the fact that the learned relationships respond coherently to perturbations rather than exhibiting spurious sensitivity. This behavior of F4-SeruNet can altogether be explained by the alignment between its architectural inductive biases and the underlying production dynamics encoded in the synthetic Seru environment. It does not optimize for instantaneous one-step accuracy primarily-the niche of strong static regressors like tree ensembles-but learns temporally consistent, cross-factor representation respectful of causality, monotonicity, and interaction effects across structural flexibility, operational efficiency, human related, and lean integration. Therein also lies a reason why F4-SeruNet maintains balanced error profiles across factors, controlled error growth under rolling-horizon forecasts, and coherent sensitivity responses under perturbations-even when its one-step RMSE is marginally higher than that of the static baselines. At the same time, using synthetic data has obvious drawbacks, i.e., even though the generator is theory-grounded and shows realistic distributions, sensitivities, and face validity, it clearly cannot capture unmodeled disruptions, behavioral heterogeneity, or rare systemic shocks present in the real factories. Thus, absolute performance values should be interpreted as a relative benchmark rather than direct deployment guarantees. Nevertheless, results show a clear

practical relevance of Seru planning and decision support. The ability of F4-SeruNet to jointly forecast multiple interacting performance factors over several horizons will allow planners to gauge trade-offs between flexibility, efficiency, workforce development, and lean stability before costly reconfigurations. Its robustness to variability and theory-consistent responses to learning and reconfiguration changes make it particularly fit for scenario analysis, policy testing, and medium-term planning in dynamic Seru production systems where interpretability and temporal coherence are as critical as the pointwise predictive accuracy.

5. CONCLUSION

The work F4-SeruNet proposed is a theory-guided temporal deep learning framework that was proposed to predict multiple interacting performance factors in Seru Production Systems: structural flexibility, operational efficiency, human related, and lean process improvement. The experimental results, synthesized by a mathematically grounded synthetic generator, showed that F4SeruNet achieves consistently high goodness-of-fit across all factors, with balanced error profiles, stable rolling-horizon forecasts, and coherent responses to controlled perturbations, thus underscoring the ability to capture temporal causality and cross-factor interactions beyond static regression models. Whereas strong nonlinear baselines achieved superior one-step errors, the proposed model outperformed them in terms of temporal consistency and robustness-essential qualities for planning and decision support in dynamic Seru environments. Future work will focus on the validation of the framework using real industrial data, extending the simulator to include disruption and demand uncertainty, integrating optimization or reinforcement learning for adaptive decision-making, and incorporating uncertainty-aware forecasting into risk-sensitive Seru planning.

REFERENCES:

- [1] M. Urگو, G. Lanza and D. Gyulai, "Future-Proof Production Scheduling and Control," *CIRP Annals*, 2025.
- [2] O. Torkul, İ. H. Selvi, M. Şişci and D. D. Diren, "A New Model for Assembly Task Recognition: A Case Study of Seru Production System," *IEEE Access*, 2024.
- [3] B. Li and Y. Wu, "Integrated optimization of worker assignment, batch splitting and scheduling for a hybrid assembly line-seru production system," *Computers & Industrial Engineering*, vol. 194, p. 110399, 2024.
- [4] Y. Yin, K. E. Stecke, M. Swink and I. Kaku, "Lessons from seru production on manufacturing competitively in a high cost environment," *Journal of operations management*, vol. 49, p. 67–76, 2017.
- [5] Y. Yu and J. Tang, "Review of seru production," *Frontiers of engineering management*, vol. 6, p. 183–192, 2019.
- [6] S. Zeng, Y. Wu and Y. Yu, "Multi-skilled worker assignment in seru production system for the trade-off between production efficiency and workload fairness," *Kybernetes*, vol. 52, p. 3495–3518, 2023.
- [7] C. Liu, K. E. Stecke, J. Lian and Y. Yin, "An implementation framework for seru production," *International transactions in operational research*, vol. 21, p. 1–19, 2014.
- [8] P. Tiwari, D. Kim, A. Hajian and A. Ghasemi, "A prescriptive analytics approach for evaluating two production systems: Simulation optimization algorithm," *Decision Analytics Journal*, vol. 12, p. 100513, 2024.
- [9] C. Liu, Z. Li, J. Tang, X. Wang and M.-J. Yao, "How SERU production system improves manufacturing flexibility and firm performance: an empirical study in China," *Annals of Operations Research*, vol. 316, p. 529–554, 2022.
- [10] Y. Ren and J. Tang, "Flexibility of seru production system: an input-process-output system view," 2022.
- [11] Z. Zhang, X. Song, H. Huang, Y. Yin and B. Lev, "Scheduling problem in seru production system considering DeJong's learning effect and job splitting," *Annals of Operations Research*, vol. 312, p. 1119–1141, 2022.
- [12] A. Ayough, F. S. Nouri, B. Khorshidvand and F. Farhadi, "Modeling workers rotation in divisional seru production systems," *Computers & Industrial Engineering*, p. 111141, 2025.
- [13] B. G. Yılmaz, Ö. F. Yılmaz and E. Cevikcan, "Lot streaming in workforce scheduling problem for seru production system under Shojinka philosophy," *Computers & Industrial Engineering*, vol. 185, p. 109680, 2023.
- [14] O. Torkul, İ. H. Selvi and M. Şişci, "Smart seru production system for Industry 4.0: a conceptual model based on deep learning for

- real-time monitoring and controlling,” *International Journal of Computer Integrated Manufacturing*, vol. 37, p. 385–407, 2024.
- [15] S. Xie, Y. Hua, S. Lu and X. Li, “A novel spatio-temporal adaptive prediction modeling strategy for industrial production process,” *IEEE Transactions on Instrumentation and Measurement*, vol. 72, p. 1–11, 2023.
- [16] J. Guo, M. Han, G. Zhan and S. Liu, “A spatio-temporal deep learning network for the short-term energy consumption prediction of multiple nodes in manufacturing systems,” *Processes*, vol. 10, p. 476, 2022.
- [17] V. Buggineni, C. Chen and J. Camelio, “Enhancing manufacturing operations with synthetic data: a systematic framework for data generation, accuracy, and utility,” *Frontiers in Manufacturing Technology*, vol. 4, p. 1320166, 2024.
- [18] G. Fu, C. Han, Y. Yu, W. Sun and I. Kaku, “A phased intelligent algorithm for dynamic seru production considering seru formation changes,” *Applied Intelligence*, vol. 53, p. 1959–1980, 2023.
- [19] J. B. Carlin and M. Moreno-Betancur, “On the uses and abuses of regression models: a call for reform of statistical practice and teaching,” *Statistics in Medicine*, vol. 44, p. e10244, 2025.
- [20] H. Zhang, J. Guo and B. Du, “Multiple batches scheduling for reconfigurable seru production system considering fatigue effect,” 2025.
- [21] K. M. Sujon, R. B. Hassan, Z. T. Towshi, M. A. Othman, M. A. Samad and K. Choi, “When to use standardization and normalization: empirical evidence from machine learning models and XAI,” *IEEE access*, 2024.
- [22] M. Kaushik and D. Choudhary, “Essential factors for seru implementation using ANP and AHP method,” *International Journal of Process Management and Benchmarking*, vol. 15, p. 524–550, 2023.
- [23] M. Xu, W. Zhou, X. Shen, J. Qiu and D. Li, “Temporal-spatial cross attention network for recognizing imagined characters,” *Scientific Reports*, vol. 14, p. 15432, 2024.
- [24] J. A. Segovia, J. F. Toaquiza, J. R. Llanos and D. R. Rivas, “Meteorological variables forecasting system using machine learning and open-source software,” *Electronics*, vol. 12, p. 1007, 2023.