Quantum Machine Learning Potential in Cyber-Physical Systems: A Case Study in Manufacturing Lines

Mosab Adwan[1], Wiam Alali[1], Osama Hijawi[1], Anas Toma[2*], Mohammad Abuabiah[3], Inès Chihi[4]
[1] Master of AI Program, Faculty of Graduate Studies, An-Najah National University, Nablus, Palestine
[2] Department of Electrical and Computer Engineering, An-Najah National University, Nablus, Palestine
[3] Mechatronics Engineering Department, An-Najah National University, Nablus, Palestine
[4] Department of Engineering, University of Luxembourg, Luxembourg
* Corresponding Author Email: anas.toma@najah.edu,
Other Emails:[1] mosab.adwan@stu.najah.edu, wiam.a.alali@gmail.com, o.hijawi@stu.najah.edu,
[3]m.abuabiah@najah.edu, [4]ines.chihi@uni.lu

***Abstract*—Cyber-physical systems (CPS) involve complex interdependencies between physical dynamics and computational control. Understanding and capturing these interactions is crucial for analyzing, predicting, and managing the performance of these systems, as it enhances understanding and facilitates optimization of operations. This study explores how to identify the buffer occupancy level of a cyber-physical serial production line, where buffer occupancy fluctuates over time. Although machine learning (ML) can learn temporal CPS behaviors, it typically requires large amounts of data and may struggle with highly nonlinear scenarios. To overcome these limitations, Quantum Machine Learning (QML) was explored and evaluated in comparison to ML models. Results indicate that the use of a hybrid QML model yields more accurate and efficient predictions of buffer occupancy levels compared to ML models, with a reduction in the mean squared error of 44.23% These indications explore new opportunities for QML in predictive modeling for CPSs and promote broader adoption of quantum computing across various applications. They also demonstrate that adopting quantum technology in CPS enhances process control, improves production reliability, and achieves faster convergence with fewer parameters.**

***Index Terms*—Cyber Physical Systems, Classical Machine Learning, Quantum Machine Learning**

I. Introduction

A Cyber-Physical System (CPS) is a complex structure constructed of physical and computational (cyber) components that interact with each other and with humans [1]. This introduces heterogeneity, as the physical components include large and complex parts, while the cyber component provides communication and computing functions to monitor, manage, and coordinate the physical elements. These systems can be found in smart grids, autonomous robots, transportation, unmanned vehicles, embedded medical devices, and mechatronic systems. Recent developments in the Internet of Things (IoT) have greatly improved data access from various CPS components. This large volume of data presents a significant opportunity to develop innovative and data-driven models that can generate valuable insights into system performance and health. Despite the availability of data, many obstacles still hinder the widespread adoption of data-driven methods in industrial environments. A key issue is the complexity of data pre-processing and manual feature extraction, due to the dynamic nature of the interconnected components of the CPS. Due to this heterogeneity, feature extraction must be carefully adapted to each component's specific properties, operating conditions, and functional roles, which further complicates the process [2].

Although advanced computing techniques can help capture dependencies in CPS, the high dimensionality and nonlinearity in such systems make it challenging to do so accurately, especially in large-scale and complex systems. Therefore, it becomes necessary to adopt more emerging techniques and paradigms, such as Quantum Computing (QC) and quantum machine learning (QML). QC has become a valuable technology for addressing complex computational problems, especially in optimization and simulation modeling [3], where it offers potentially greater computational power compared to traditional methods [4].

It is built on the basic principles of quantum mechanics, such as superposition and entanglement. These principles enable capabilities that might be difficult for classical methods. For example, superposition allows exponential data encoding, so a large amount of data can be represented with a small number of qubits [5]. Entanglement between qubits can be used to represent and analyze complex data. It also potentially helps in

discovering hidden and complex patterns in data [6]. Theoretical and early experimental research shows that QML may offer many advantages over classical approaches, including faster execution, higher learning efficiency, and better generalization from limited data. QML models might achieve similar or superior performance compared to classical models with fewer samples. This is especially important in fields where data is scarce and expensive to collect, enabling faster training and lower energy consumption. However, this does not mean that quantum models will always outperform classical ones. These benefits greatly depend on the specific problem and conditions. This study employed hybrid quantum-classical models and evaluated them using a framework developed based on operational data from machine and buffer behaviors.

II. Literature Review

Deep learning (DL) has emerged as a powerful technique for handling the inherent complexities of CPS. It has been utilized in various applications, including control, optimization, fault diagnosis, and anomaly detection. Gaba et al. [7] reviewed and analyzed various attack detection techniques available for CPS using DL, discussing the challenges such as the need for large amounts of data for training. In contrast, CPS data is often limited and expensive to collect. The integration of QC and QML with CPS was studied in revolutionizing Industry 6.0 by Pineda et al. [8]. They provide a systematic literature review that aims to identify variables and applications associated with QC-AI integration. While Devadas and Sowmya [9] offered an overview of QML and hybrid quantum-classical frameworks, gathering the most recent developments in QML's theoretical basis, practical applications, and future directions. Meanwhile, Villalba-Diez et al. [4] promoted QC by introducing a quantum framework that assesses Industry 4.0 cyber-physical systems more efficiently than typical simulations used to represent interconnected systems. While Gao and Wu [10] explored innovative quantum-related strategies for addressing power system challenges to show how power systems can benefit from QC. Rajawat et al. [11] proposed a Quantum DL to find attacking patterns in CPS for the purpose of bridging the gap between implementing security measures and moving forward with the digitization of all industries.

In the security aspect, numerous studies have explored the use of quantum technology in identifying cyberattacks on CPSs. Saber et al. [12] proposed a framework to detect Fault Masking Attacks to disguise faults on the protected line. On the other hand, new research has emerged on post-quantum CPS in preparation for the expected ability of quantum computers to break the highly developed cryptographic algorithms that protect numerous systems [13]. Furthermore, QC has a promising future in the field of fault diagnosis. Chen et al. [14] proposed a novel fault diagnosis approach using quantum Q-learning, where his method outperformed traditional methods. Cultice et al. [15] explored the application of quantum-hybrid support vector machines and robust preprocessing techniques in anomaly detection, aiming to improve the identification of cyber-attacks in physical sensor data. Kuo et al. [16] proposed a Quantum-inspired Tabu Search algorithm with Entanglement to estimate the bare minimum of sensors needed and in what position they should be placed.

III. Problem Statement and Research Gap

Despite the increasing research on QC and QML, a critical gap remains: little work integrates these methods with the unique data characteristics of CPS. Most current efforts mainly focus on securing the cyber components of the CPS against cyber attacks [17], often overlooking the physical layer where real-world dynamics and failures can happen, or concentrating on domain-specific applications such as smart grids, leaving other areas such as healthcare, transportation, industry, and autonomous systems underrepresented [12]. Furthermore, most comparisons of classical and QML algorithms have been conducted on general benchmark datasets with characteristics that differ from those of CPS, including temporal dependencies, multi-modal sensing, and real-time limitations. This creates a massive gap in our comprehension of whether quantum or hybrid techniques can outperform classical ML in CPS contexts. Addressing this gap is essential not only for academic reasons but also for its practical significance, as CPS supports critical infrastructure and requires higher standards of reliability, security, and intelligence. Recognizing the potential of quantum and hybrid approaches in this domain may open up new opportunities and mitigate hazards that existing methods cannot fully address.

IV. Methodology

A. Data Collection

A CPS-based simulation representing a serial production line (SPL) was employed to address common limitations of real industrial datasets, including missing data, inconsistent logging, and low temporal resolution. The designed simulation integrates discrete manufacturing events, network communication, and continuous physical dynamics, allowing precise control of process behavior and providing high-quality, fully synchronized data. This enables repeatable experiments and comprehensive evaluation of data-driven monitoring and predictive maintenance strategies. The SPL model consists of three machines, M1, M2, and M3, in series with finite buffers, B1 and B2, between them to absorb

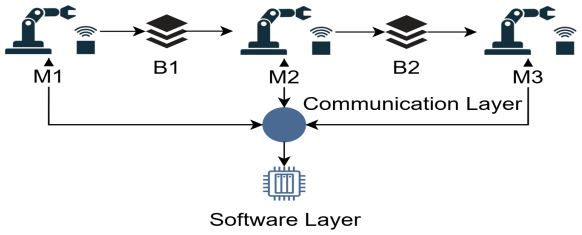

Fig. 1. The SPL Setup with Three Machines and Two Buffers Equipped with Multi-Sensors

flow imbalances. Every machine is equipped with sensors for temperature, current, and vibration, and all machines communicate with a centralized controller through a network layer that introduces latency and probabilistic packet loss as shown in Figure 1.

The SPL topology was selected because it illustrates the fundamental dependencies and flow limitations found in manufacturing. Previous simulation and analytical studies on unreliable production lines with finite buffers inspired the design of the SPL model and its dynamic behavior under machine downtime and recovery [18]. Both operational dynamics and machine reliability behavior were shown by using dynamic heating and cooling profiles that emulate the Mean Time To Failure (MTTF) and Mean Time To Repair (MTTR). As processing continues, thermal energy builds up, signifying wear and operational load. For this implementation, a discrete-time representation of the standard first-order thermal equation was used to approximate each machine's thermal behavior, where during operation the temperature increase $\frac{dT}{dt} = r_h$ and during recovery it decreases as $\frac{dT}{dt} = -r_c$. Here, the machine's temperature is denoted by T, r_h is the effective heating rate while operating (which defines MTTF), and r_c is the effective cooling rate during recovery (defining MTTR). These equations were discretized into one-second intervals for numerical simulation. Random perturbations were applied to both heating and cooling rates at each step to account for natural variability in thermal responses, consistent with general approaches in reliability modeling [19].

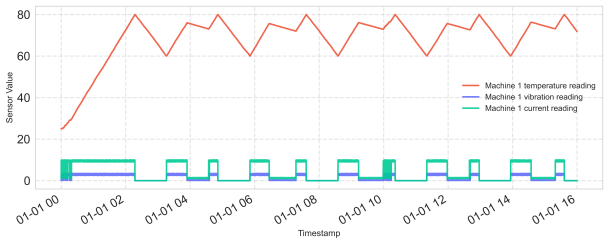

Fig. 2. Temporal sensors readings for M1 during an operational shift

The machine fails when its temperature exceeds $80\,^{\circ}\mathrm{C}$, triggers a cooling and recovery phase till the temperature falls below $60\,^{\circ}\mathrm{C}$ Figure 2.. Thermal system studies conceptually support the modeled heating and cooling dynamics [20]. Mapping temperature to reliability metrics such as MTTF and MTTR is consistent with general reliability modeling approaches in CPS and cloud cooling systems. Moreover, each machine has a mean processing time for each part it processes, and each buffer has a maximum capacity of 50 parts.

All of the data streams from this simulation are resampled into uniform time increments and processed to construct a dataset suitable for regression tasks. The target variable corresponds to buffer occupancy at a given moment, while the features record the most recent processing history, machine statuses, and temperature.

B. Data Preprocessing Modeling

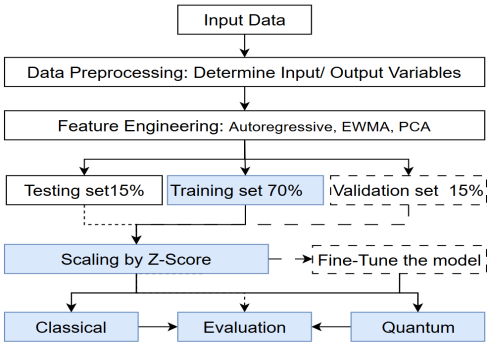

Fig. 3. Data Preprocessing Workflow

The preprocessing phase began by transforming the raw CPS into a clean time-series dataset with aggregated features every 5 seconds, resulting in a dataset of 11520 rows,26 features and two targets. Features related to the number of processed or accumulated parts were removed as they represent the system's historical state and not its causal dynamics and might introduce information leakage. This resulted in a final number of features of 19.

As shown in Figure 3, the dataset was split chronologically into a training–validation set (85%) and a test set (15%). All numeric features were scaled using z-score normalization, computed solely from the statistics of the training set to prevent data leakage and ensure the model generalizes properly. The standardization formula $z = \frac{x-\mu_{\text{train}}}{\sigma_{\text{train}}}$ uses the mean (μ_{train}) and standard deviation (σ_{train}) calculated from the training portion only. After that, several complementary feature engineering pipelines were explored to enhance the modeling capabilities and were used in various architectures of

neural network (NN) models. The number of each feature each pipeline produce is reported in Table I. The BASE pipeline uses all numeric features without any modification. For improved computational efficiency, a BASE with a Principal Component Analysis (PCA) pipeline is used, which applies train-only Singular Value Decomposition (SVD) and retains the fewest principal components necessary to explain the target variance. Moreover, the Y_LAGS pipeline adds lagged versions of each target variable with one lag $y_{t-1}\ldots$ to exploit the time-series nature of the data. In addition, the EWMA pipeline includes causal exponentially weighted moving averages of each target and their residuals ($y_t - \mathrm{EWMA}_t$) [21].

TABLE I
NUMBER OF INPUT FEATURES PER PIPELINE

Pipeline	Raw Features	Added Features	PCA Components	Final Feature Dimension
BASE	18	0	–	18
Y-LAGS	18	2	–	20
EWMA / EWD	18	12	–	30
PCA	18	0	4 (95% var)	4

C. Classical ML Models

The Recurrent Neural Network (RNN) was first used in this study, as it handles sequence prediction problems and is designed with a feedback loop of the recurrent cells that inherently addresses the temporal order and dependencies of its given sequences of data [22]. However, RNNs may struggle with the vanishing gradient problem, which affects and limits their ability to capture information from distant elements in data sequences. The Long Short-Term Memory (LSTM) addresses this challenge through its gating mechanisms by embedding a system of addition, forget, input, and output gates to organize the information flow. On the other hand, the LSTM may suffer from high computation overhead. Therefore, a new architecture called Gated Recurrent Unit (GRU) was developed to perform equally well as LSTM on sequence modeling problems with reasonable computation overhead [23]. Another model, originally introduced for natural language processing and known as the Transformer, was utilized and demonstrated promising results in pattern prediction for time series applications [24]. This model can grasp long-range dependencies and interactions between several agents in a scene by attending to relevant spatial and temporal information.

D. Quantum Machine Learning

QC is the study of information processing using quantum hardware. In recent years, technology has witnessed rapid and impressive advances. Quantum technologies are among the promising fields expected to progress steadily, motivating many researchers to empirically explore this field through simulation, optimization, and machine learning.

The preprocessing pipeline Y-LAGS was applied to create sequences with window sizes of 32 and 64 for subsequent model training and evaluation. Two hybrid quantum–classical architectures were selected: HybridQNNLSTM and HybridQNNRNN, where the classical recurrent models (LSTM and RNN) were used to encode the temporal dependencies in the data. Then, a fully connected layer was used to map the output of the last time step to an n-dimensional feature vector compatible with the n-qubit quantum block, where four qubits are used. Following the quantum processing, a final linear layer was used to produce the final multi-output forecasts. The Quantum Neural Network consists of (1) an embedding layer, (2) a variational circuit (*ansatz*), and (3) a measurement layer. The embedding layer can be viewed as a feature map that encodes the classical data into quantum states in a high-dimensional Hilbert space [25]. An angle encoding was used for embedding, where each real-valued feature is mapped to a rotation gate applied to a designated qubit. These rotation angles are data-dependent and not trainable.

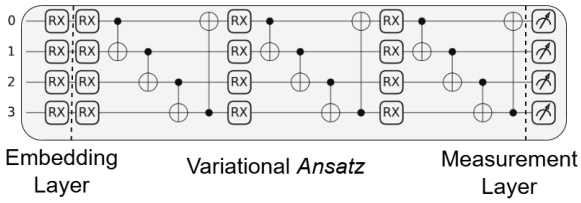

Fig. 4. QNN layer using four qubits.

PyTorch and Pennylane were used to create the hybrid classical models as shown in Figure 4. The PennyLane library supports differentiable quantum programming, enabling the development of hybrid models and quantum machine learning applications. Its templates and device interfaces for quantum hardware and simulators make it easy to embed a variational quantum circuit inside a PyTorch module and train the entire hybrid model with classical deep-learning tools. The *BasicEntanglerLayers* was used as it applies parameterized single-qubit ($R_X(\theta_i)$ on qubit i) rotations on all qubits, followed by a ring of two-qubit CNOTs where each qubit is connected to its neighboring qubit, and the last qubit wraps around to connect with the first qubit. At the same time, the trainable parameters (θ) are similar to weights in classical neural networks and are optimized jointly end-to-end with the classical layers. The measurement layer decodes the quantum states into classical information by computing the expectation values of an observable, where the Pauli-Z gate is applied to each qubit. The

output of the quantum block is a compact feature vector that the final linear layer maps to the desired multi-output forecasts.

E. Hyperparameter Optimization and Evaluation

Two strategies were employed for hyperparameter tuning. The first one comprises classical models, which are optimized through a grid search hyperparameter search with strict temporal validation. The second one explores quantum (hybrid) models using a grid search to find the best combination of hyperparameters. These hybrid models utilize a classical encoder to parameterize a small variational quantum circuit, with circuit expectations feeding into a classical regression head. The search spaces included the hidden state size of the classical recurrent model, the number of layers in the classical recurrent model, the learning rate, the batch size, and the depth of the variational *ansatz*. The *ansatz* depth refers to the number of repeated variational blocks, the parameterized rotations and entanglement blocks, within the QNN. The Adam optimizer was used for all models to minimize the Mean Squared Error (MSE) with early stopping based on validation loss, thereby preventing overfitting and reducing training time. Then, the best-performing model was evaluated using test data and compared using the MSE, Mean Absolute Error (MAE), and the coefficient of determination (R^2) score. Model selection and parameter tuning were performed only on the training–validation set using expanding-window time-series cross-validation. In each of the three TimeSeriesSplit folds, the validation set comprised the next 15% of the training–validation data, with the training set including all earlier observations. The final model was retrained on the full training–validation set with the best hyperparameters and evaluated once on the separate test set.

V. Experimental Evaluation and Simulation

A. Setup

The system was simulated using a continuous-time discrete-event model with a single run and a distinct fixed random seed (12) controlling all stochastic events, including part arrivals, processing times, thermal dynamics, and network effects. This single-run approach was chosen to ensure exact replicability. It avoids unrealistic results caused by averaging discrete buffer occupancies across numerous runs, which might result in fractional numbers that do not accurately represent the system's behavior. The deterministic design ensures that consecutive executions of the same seed provide similar outcomes. The simulation run over 16 operational hours (57,600 seconds), features sensors updates every second, and logs observations every 5 seconds. Table II provides the observed machine failures, mean time between failures (MTBF), and additional simulation parameters.

TABLE II
Simulation and Network Parameters

Parameter	Value
Number of runs	1
Part arrival mean (sec)	20
Process time mean (sec)	20
Process time std (sec)	20
Network latency mean (sec)	0.5
Network latency std (sec)	0.2
Packet loss rate	0.02
Overheat temperature	80
Cooling resume temperature	60
Base temperature	25
Average heating (seconds)	7,779.59
Average cooling (seconds)	16,051.05
Heating rate (base)	0.00707
Cooling rate (base)	0.00343
M1 failures	6
M2 failures	5
M3 failures	5
M1 MTBF (sec)	9,588.71
M2 MTBF (sec)	9,679.87
M3 MTBF (sec)	10,801.04
Mean network delay (sec)	0.499
Std network delay (sec)	0.199
Min network delay (sec)	0
Max network delay (sec)	1.300

The experiments were implemented in cloud-based environments (Kaggle for the simulation and the classical models, while Google Colab is used for the hybrid models). This separation of execution environments does not influence the validity or reproducibility of the results, as the outputs of the simulation were logged deterministically, and the seed was fixed for the data-driven models

B. Results

Predicting the buffer occupancy levels between sequential machines was the main objective of this experiment, and it can be viewed as a critical indicator of production line stability and overall performance. The overall performance was calculated by computing the macro-average across both buffers, as shown in Table III. These results reveal a high variation in the performance across different algorithms and benchmarks. The LSTM (EWMA) model achieved the best macro results, with a low MSE value of 12.14 and an MAE of 2.00, along with a high R^2 score of 0.9736. This demonstrates that a relatively simple LSTM, together with suitable data preparation, can effectively capture buffer patterns. Both RNN and Transformer models demonstrated good predictive capabilities, achieving R^2 scores of 0.9490 and 0.8899, respectively.

TABLE III
TEST PERFORMANCE AND MODEL DESIGN CHARACTERISTICS (BEST RESULTS)

Pipeline–Model	Model Design							Macro Performance Metrics			B1 Metrics (M1→M2)			B2 Metrics (M2→M3)		
	Layers	Hidden	Window	LR	Time (s)	Params (K)	Convergence	MSE	MAE	R^2	MSE	MAE	R^2	MSE	MAE	R^2
EWMA–LSTM	1	160	64	0.001	301.04	621.48	38	12.14	2.00	0.9736	7.35	2.03	0.9788	16.93	1.97	0.9685
EWMA–RNN	2	160	64	0.001	385.40	31.042	24	22.77	3.11	0.9490	16.73	2.98	0.9516	28.81	3.24	0.9464
Y_LAGS–RNN	1	160	48	0.001	632.77	29.442	42	38.92	4.19	0.9026	48.59	4.42	0.8596	29.3	3.96	0.9456
Y_LAGS–Transformer	1	160	64	0.001	567.30	210.242	16	42.14	5.10	0.8899	61.63	6.77	0.8219	22.65	3.44	0.9579

TABLE IV
TEST PERFORMANCE AND MODEL DESIGN CHARACTERISTICS (BEST RESULTS)

Hybrid–Model	Model Design									Macro Performance Metrics			B1 Metrics (M1→M2)			B2 Metrics (M2→M3)		
	Layers	Hidden	Batch size	q_depth	LR	NN params.	QNN params.	Time (s)	Convergence	MSE	MAE	R^2	MSE	MAE	R^2	MSE	MAE	R^2
LSTM-QNN	1	8	32	3	0.001	1006	12	333.63	37	17.61	1.68	0.9644	17.78	3.05	0.9572	17.43	2.31	0.9697
RNN-QNN	1	8	32	2	0.001	286	8	454.32	59	6.77	2.23	0.9863	9.16	2.75	0.9779	4.63	1.71	0.9924

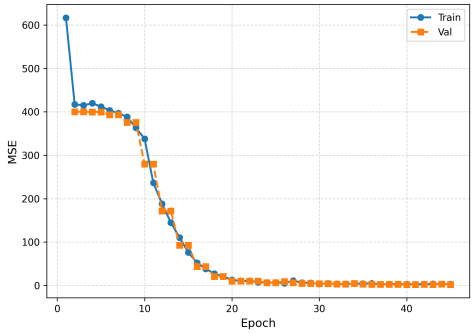

Fig. 5. Training and validation MSE convergence of the LSTM (EWMA) Model.

While the RNN (EWMA) model exhibited a more balanced efficiency across both buffers, with a slight decline compared to the other models, it presented an edge over the other techniques. The LSTM (EWMA) model demonstrated a strong alignment between predicted and actual values across the two buffers, as shown in Figure 7, where predictions closely track ground truth, with a zoomed view shown in Figure 8. While the Transformer converged early at 16 epoch with great predictive power.

Figure 5 shows the training curve of the LSTM (EWMA) model, where its obvious the model struggles in the training phase for the first 10 epochs. Then the models converges at epoch 38. Results in Table IV report test metrics across the two buffers for the hybrid models. The Hybrid LSTM-QNN (Y_LAGS) with $R^2 = 0.9644$ almost matches the performance of the best classical model (LSTM with EWMA pipeline) with $R^2 = 0.9736$ as shown in Table III.

The Hybrid RNN-QNN (Y_LAGS) sets a new best score, improving the macro MSE from 22.77 to 6.77, MAE from 3.11 to 2.23 , and R^2 from 0.9490 to 0.9863. For each buffer, the hybrid models are consistent. Against RNN (EWMA), Hybrid RNN-QNN reduces error on B1 for MSE from 16.73 to 9.16, while MAE is reduced from 2.98 to 2.75, and raises the R^2 from 0.9516 to 0.9779. For B2, the same happens as MSE is reduced from 28.81 to 4.63, while MAE is reduced from 3.24 to 1.71, and R^2 improves from 0.9464 to 0.9924.

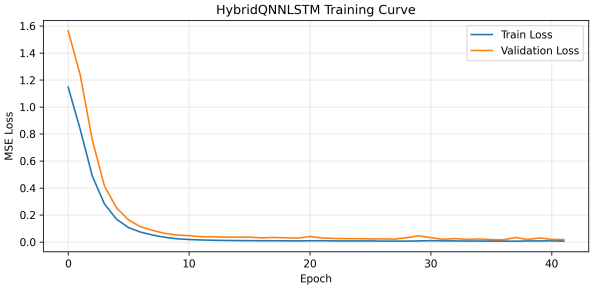

Fig. 6. Training and validation MSE convergence for the Hybrid LSTM-QNN model.

Overall, both hybrid models narrow the prediction gap between B1 and B2 while achieving better overall performance with Hybrid RNN-QNN. Figure 6 shows the smooth training curve for the hybrid LSTM-QNN model where it converged at 37 epoch. For a fair comparison both for classical and hybrid model the training time, number of parameters and convergence epochs were reported in Tables III & IV. The training for all models started with 80 epochs with an early stopping with patience of 5 epochs.

Besides the improvement in performance , the hybrid models has a low number of parameters and a fast convergence. Figure 9 shows that the model closely tracks the ground truth for both buffers. The difference between the predicted and actual values is almost negligible as shown in Figures 8 &10.

The hybrid model reduces under- and overshoots, particularly on B2, where the classical curve shows more lag, reflecting the numerical metrics referred to earlier.

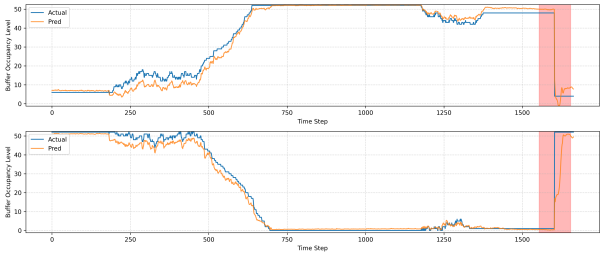

Fig. 7. Actual vs. predicted buffer occupancy for classical LSTM (EWMA) Model.

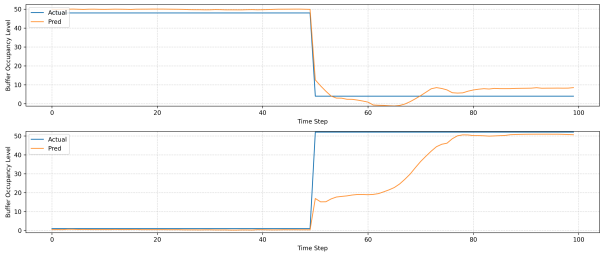

Fig. 8. A Zoom in for the actual vs. predicted buffer occupancy for LSTM (EWMA) Model.

These results were obtained under an ideal quantum simulator (default.qubit). The quantum circuit was intentionally restricted to a small number of qubits (4 quits) and shallow depth (2-3 layers), which is aligned with current NISQ devices and helps reduce the effect of noise when run on a real quantum computer.

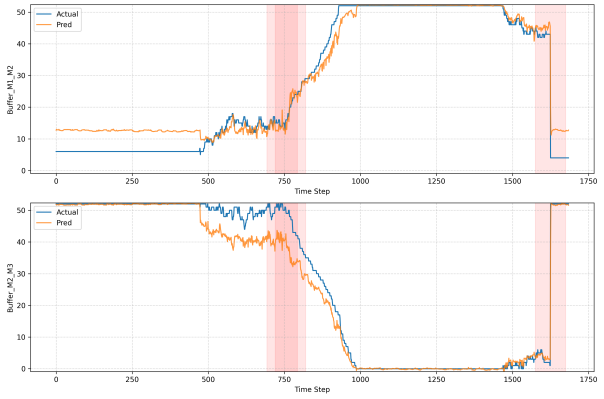

Fig. 9. Actual vs. predicted buffer occupancy for the hybrid LSTM-QNN models.

VI. Conclusion

The results highlight the potential of QML in time-series forecasting, particularly in the industrial sector. They demonstrate that a hybrid quantum-machine learning model achieved performance comparable to ML and even slightly outperformed it in some cases, resulting in a high R^2 of 98.6%. Although QML slightly outperformed classical ML, the results highlight the influence

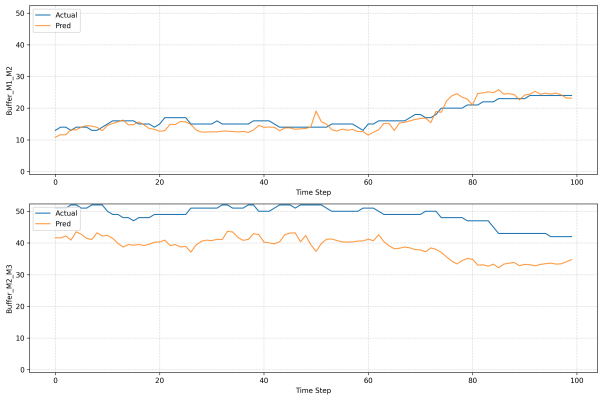

Fig. 10. A Zoom in for the actual vs. predicted buffer occupancy for hybrid LSTM-QNN Model.

of quantum technology in better representing and optimizing industrial production lines, as well as reducing downtime in smart manufacturing environments. Future work could explore the effects of different quantum hardware noise models and develop customized, improved variational ansatzes to reduce noise and enhance the robustness of QML models. Nonetheless, these results demonstrate that integrating quantum and classical layers can capture complicated nonlinear relations in time series data, achieve high prediction accuracy and efficiency, and provide higher expressiveness with fewer parameters compared to classical models. Furthermore, future work will extend the proposed framework to real-world datasets to validate the capabilities of these hybrid models in practical settings and account for noise, sensor variability, and rare events.

References

[1] V. Gunes, S. Peter, T. Givargis, F. Vahid *et al.*, "A survey on concepts, applications, and challenges in cyber-physical systems." *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 12, pp. 4242–4268, 2014, https://doi.org/10.3837/tiis.2014.12.001.

[2] P. Sajjadi, F. Dinmohammadi, and M. Shafiee, "Fault detection of cyber-physical systems using a transfer learning method based on pre-trained transformers," *Sensors*, vol. 25, no. 13, p. 4164, 2025, https://doi.org/10.3390/s25134164.

[3] H. Araujo, X. Wang, M. Mousavi, and S. Ali, "Using quantum annealing to generate test cases for cyber-physical systems," *arXiv preprint arXiv:2504.21684*, 2025, https://doi.org/10.48550/arXiv.2504.21684.

[4] J. Villalba-Diez, A. González-Marcos, and J. Ordieres-Meré, "Quantum cyber-physical systems," *Scientific Reports*, vol. 12, no. 1, p. 7964, 2022, https://doi.org/10.1038/s41598-022-11691-x.

[5] S. M.-L. Pfaendler, K. Konson, and F. Greinert, "Advancements in quantum computing—viewpoint: Building adoption and competency in industry," *Datenbank-spektrum*, vol. 24, no. 1, pp. 5–20, 2024, https://doi.org/10.1007/s13222-024-00467-4.

[6] C. Aishwarya, M. Venkatesan, and P. Prabhavathy, "A scoping survey of quantum machine learning and deep learning for real-world applications," *Procedia Computer Science*, vol. 258, pp. 633–646, 2025, https://doi.org/10.1016/j.procs.2025.04.297.

[7] S. Gaba, I. Budhiraja, V. Kumar, S. Martha, J. Khurmi, A. Singh, K. K. Singh, S. S. Askar, and M. Abouhawwash, "A systematic analysis of enhancing cyber security using deep learning for

cyber physical systems," *IEEE Access*, vol. 12, pp. 6017–6035, 2024, https://doi.org/10.1109/ACCESS.2023.3349022.

[8] V. G. Pineda, A. Valencia-Arias, F. E. L. Giraldo, and E. A. Zapata-Ochoa, "Integrating artificial intelligence and quantum computing: A systematic literature review of features and applications," *International Journal of Cognitive Computing in Engineering*, 2025, https://doi.org/10.1016/j.ijcce.2025.08.002.

[9] R. M. Devadas and T. Sowmya, "Quantum machine learning: A comprehensive review of integrating ai with quantum computing for computational advancements," *MethodsX*, p. 103318, 2025, https://doi.org/10.1016/j.mex.2025.103318.

[10] F. Gao and G. Wu, "Application of quantum computing in power systems," *Energies*, vol. 16, no. 5, 2023, https://doi.org/10.3390/en16052240.

[11] A. S. Rajawat, S. Goyal, P. Bedi, N. B. Constantin, M. S. Raboaca, and C. Verma, "Cyber-physical system for industrial automation using quantum deep learning," in *2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART)*. IEEE, 2022, pp. 897–903, https://doi.org/10.1109/SMART55829.2022.10047730.

[12] A. M. Saber, A. Youssef, D. Svetinovic, H. Zeineldin, and E. F. El-Saadany, "Unmasking covert intrusions: Detection of fault-masking cyberattacks on differential protection systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2024, https://doi.org/10.1109/TSMC.2024.3456810.

[13] M. Barbeau and J. Garcia-Alfaro, "Cyber-physical defense in the quantum era," *Scientific Reports*, vol. 12, no. 1, p. 1905, 2022, https://doi.org/10.1038/s41598-022-05690-1.

[14] Y. Chen, K. Deng, X. Du, Z. Chang, and T. Wan, "A quantum q-learning fault diagnosis method for intelligent manufacturing equipment," *Machines*, vol. 13, no. 7, p. 629, 2025, https://doi.org/10.3390/machines13070629.

[15] T. Cultice, M. S. H. Onim, A. Giani, and H. Thapliyal, "Anomaly detection for real-world cyber-physical security using quantum hybrid support vector machines," in *2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2024, pp. 619–624, https://doi.org/10.1109/ISVLSI61997.2024.00117.

[16] S.-Y. Kuo, Y.-H. Chou, and C.-Y. Chen, "Quantum-inspired algorithm for cyber-physical visual surveillance deployment systems," *Computer Networks*, vol. 117, pp. 5–18, 2017, https://doi.org/10.1016/j.comnet.2016.11.013.

[17] V. K. Bathalapalli, S. P. Mohanty, C. Pan, and E. Kougianos, "Qpuf 2.0: Exploring quantum physical unclonable functions for security-by-design of energy cyber-physical systems," in *2025 IEEE 26th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2025, pp. 281–286, https://doi.org/10.1109/WoWMoM65615.2025.00055.

[18] T. McNamara, S. Shaaban, and S. Hudson, "Simulation of unbalanced buffer allocation in unreliable unpaced production lines," *International Journal of Production Research*, vol. 51, no. 6, pp. 1922–1936, 2013, https://doi.org/10.1080/00207543.2012.720726.

[19] J. Faulin, A. A. Juan, S. Martorell, and J.-E. Ramírez-Márquez, *Simulation methods for reliability and availability of complex systems*. Springer, 2010, https://doi.org/10.1007/978-1-84882-213-9.

[20] D. F. Gomes, R. P. Tavares, and B. M. Braga, "Mathematical model for the temperature profiles of steel pipes quenched by water cooling rings," *Journal of Materials Research and Technology*, vol. 8, no. 1, pp. 1197–1202, 2019, https://doi.org/10.1016/j.jmrt.2018.06.022.

[21] A. A. H. Ahmadini, I. Khan, H. AlQadi, and S. Hussain, "Enhancing risk-adjusted ewma control chart utilizing artificial neural networks," *Journal of Statistical Theory and Applications*, vol. 24, pp. 42–54, 2025, https://doi.org/10.1007/s44199-024-00094-8.

[22] H. Hewamalage, K. B. Lim, U. Mumtaz, A. R. Ramachandran, P. Bhardwaj, S. Lee, M. T. Qin, and R. Wang, "Recurrent neural networks for time series forecasting: Current status and future directions," *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, 2021, https://doi.org/10.1016/j.ijforecast.2020.06.008.

[23] G. Velarde, P. Brañez, A. Bueno, R. Heredia, and M. Lopez-Ledezma, "An open source and reproducible implementation of lstm and gru networks for time series forecasting," *Engineering Proceedings*, vol. 18, no. 1, p. 30, 2022, https://doi.org/10.3390/engproc2022018030.

[24] V. Bharilya and N. Kumar, "Machine learning for autonomous vehicle's trajectory prediction: A comprehensive survey, challenges, and future research directions," *Vehicular Communications*, vol. 46, p. 100733, 2024, https://doi.org/10.1016/j.vehcom.2024.100733.

[25] Y.-Y. Hong, C. J. E. Arce, and T.-W. Huang, "A robust hybrid classical and quantum model for short-term wind speed forecasting," *IEEE Access*, vol. 11, pp. 90 811–90 824, 2023, https://doi.org/10.1109/ACCESS.2023.3308053.