Predicting Error Types and Timing in Quay Crane Operations with eXtreme Gradient Boosting

Robert Klar*†, Vangelis Angelakis*
*Department of Science and Technology, Linköping University, Campus Norrköping, 60 174, Sweden
†Swedish National Road and Transport Research Institute (VTI), SE-581 95 Linköping, Sweden
E-mail: robert.klar@liu.se, vangelis.angelakis@liu.se

***Abstract*—Efficient port operations depend on the disruption-free operation of quay cranes (QCs), which transfer containers between vessels and internal trucks. As global container throughput rises, QCs face increased pressure, resulting in accelerated wear and tear. This can lead to QC downtime, which could interrupt the entire chain of port operations. Therefore, timely identification and prediction of critical errors is essential to enable timely maintenance to lower the risk of downtime. This study utilizes two years of QC monitoring data, enriched with weather conditions and terminal operational context, alongside twenty critical error events identified by the terminal operator. The goal is to predict the occurrence and timing of these critical errors through a three-stage machine learning model. The first stage predicts the type of the next critical event based on historical error patterns, warnings, and contextual data. The second stage estimates a time window in which the event will occur. The third stage refines timing predictions when more than one hour remains. The first two stages are formulated as multiclass classification problems, and the third as a regression task. All stages utilize eXtreme Gradient Boosting (XGBoost). SHapley Additive exPlanations (SHAP) are used to identify influential features. Results show that the model predicts the next critical error type with 83% accuracy and its immediacy with 71% accuracy. However, approximating the timing of events anticipated to occur beyond one hour remains challenging. These findings support proactive maintenance planning and operational adjustments, helping port operators mitigate disruptions and enhance QC reliability.**

***Index Terms*—eXtreme Gradient Boosting (XGBoost), Machine Learning, Predictive Maintenance, Quay Cranes, Resilient Port Operations.**

I. Introduction

Disruption-free operation of quay cranes (QCs) is essential for efficient port operations. QCs act as an interface for ship-to-shore operations, transferring containers between vessels and the ports' internal trucks [1]. However, while global container throughput is consistently rising, the number of QCs often remains limited due to spatial expansion constraints [2] and the high costs of acquiring and maintaining QCs [3].

An overview of global container throughput over the past 25 years is presented in Figure 1. The data reveals a consistent upward trend, with global throughput rising from 759 million twenty-foot equivalent units (TEUs) in 2020 to 978 million TEUs in 2025—a 28.85% increase [1]. Notably, this growth persisted despite the disruptions caused by the COVID-19 pandemic. Over the entire 25-year period, global container throughput has surged by 334.67%, underscoring the steady growth of maritime transport [4].

This persistent growth places increasing pressure on port infrastructure, driving the need to maximize equipment utility and operational efficiency [5]. Among all port assets, QCs are particularly vulnerable to wear and tear. Their critical role in bridging maritime and landside operations, combined with high workload, operational complexity, limited availability, and safety concerns, especially when multiple QCs operate in parallel, makes them the most error-prone and frequent bottleneck in terminal operations [4], [6].

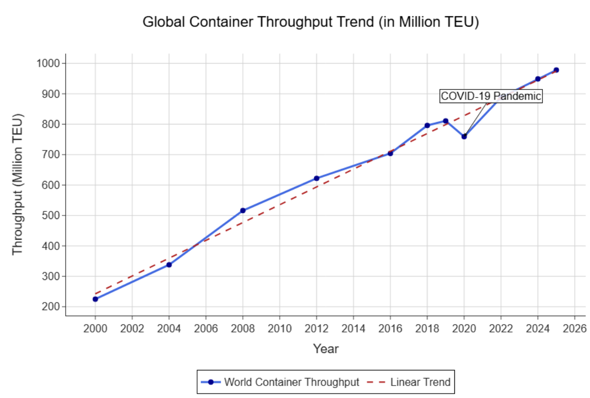

Fig. 1: Visualization of world container growth [1].

Consequently, the performance of QCs is recognized as a key performance indicator (KPI) in evaluating overall port efficiency. Notteboom et al. [5] identify six core KPIs, as summarized in Table I. The table illustrates how QC downtime affects the entire operational chain: it directly impacts KPIs P2, P3, and P4, while indirectly influencing P1, P5, and P6. Among these, crane performance (P3), typically measured by the average number of QC movements per hour, is a common bottleneck. For maritime shipping companies, this metric is particularly critical, as it directly correlates with vessel turnaround time and, consequently, port-related costs [5].

Building on the classification proposed by Li et al. [7], recent research approaches QC disruptions from three key angles: (1) identifying specific fault types, (2) uncovering their

This work has been supported by Trafikverket Sweden as part of the Triple F (MODIG-TEK) project under Grant 2019.2.2.16. We would further like to thank Yilport Gävle for providing us with operational data and insights regarding their quay crane maintenance routines.

TABLE I: Impact of QC Downtime on port performance [5].

Performance Indicator	Description	Impact by Crane Downtimes
Average Anchorage Time (P1)	Time ships wait at anchorage before berthing.	Longer berth occupancy reduces slot availability and increases anchorage delays.
Ship Turnaround Time (P2)	Total time a ship spends in port.	Delayed handling extends port stay and disrupts schedules.
Crane Performance (P3)	Average container moves per crane per hour.	Downtime reduces operational availability, severely diminishing productivity.
Yard Dwell Time (P4)	Time containers remain in the yard.	Slower unloading causes intra-terminal congestion and longer storage times.
Truck Turnaround Time (P5)	Time trucks spend in the terminal.	Container delays lead to longer truck queues.
Gate Waiting Time (P6)	Time trucks wait at terminal gates.	Unloading delays lead to disrupted truck schedules.

underlying causes, and (3) developing rescheduling strategies for QC tasks following fault events [8]. While much of the existing literature focuses on scheduling strategies, relatively few studies leverage machine learning (ML) to understand or predict the root causes of QC faults.

ML has been applied in various QC-related contexts, for instance, to forecast crane productivity using artificial neural networks [9], or to enhance port safety by predicting container terminal accidents based on historical operational data [10]. However, its use in diagnosing and predicting events that might cause QC disruptions remains limited. Existing ML-based approaches towards exploring QC downtime prediction tend to concentrate on isolated components, such as the electric control system [11], [12] or the hoist system [13], [14], and often lack integration with broader terminal operations or contextual data such as weather conditions. Consequently, Filom et al. [15] highlight the need for more comprehensive ML-driven research into QC downtime.

Given the growing complexity of QC systems and the shortcomings of current predictive methods, there is an urgent need for more integrated, data-driven solutions capable of anticipating particular error types and their timing. This paper addresses this need by proposing a machine learning framework that uses real-world QC monitoring data enriched with operational and environmental context to predict critical error events and their expected occurrence times. The next section outlines the specific research gaps addressed and summarizes the key contributions of this study.

II. Research gap and contributions

QCs are essential to efficient port operations, enabling the transfer of containers between sea and land and thereby facilitating multimodal transport. However, their limited availability and intensive usage make downtime particularly disruptive to terminal performance [16]. This has motivated research efforts aimed at understanding, predicting, and mitigating critical events that lead to QC downtime.

Most existing literature on QC downtime focuses on scheduling-based solutions rooted in operations research (OR) [3], [8], [16]. These approaches primarily address post-downtime rescheduling to maintain high operational efficiency. In contrast, there is limited research leveraging machine learning (ML) to proactively understand and predict the causes of QC faults before they occur. The few ML-based studies in this area tend to focus on anomaly detection [14], fault identification in QC components or subsystems [12], [17], or post-hoc analysis of explanatory features contributing to QC downtime [18].

TABLE II: Comparison of relevant papers on predicting and preventing QC downtime.

Ref.	Summary
OR-based Approaches	
[8]	Proposes integrated scheduling of QCs, yard cranes, and automated guided vehicles under QC fault scenarios. Uses mixed integer programming and a two-stage NSGA-II algorithm. Demonstrates reduced fault impact and improved operational continuity.
[19]	Addresses joint berth allocation and QC assignment with preventive maintenance. Employs integer linear programming and an ϵ-constraint heuristic. Shows that Pareto-optimal solutions outperform traditional maintenance strategies in terms of efficiency and fault resilience.
[3]	Develops a nonlinear integer programming model for integrated berth, QC, and truck scheduling with QC maintenance. Uses a hybrid SWO-GA algorithm for large-scale instances. Results show significant improvements in retrofit scheduling and fault recovery.
ML-Based Approaches	
[14]	Applies unsupervised learning to detect anomalies in QC motor vibrations using irregular IoT data. Combines load-based clustering with one-class SVMs. Effectively flags discordant patterns without labeled data, enabling early fault detection.
[20]	Uses deep learning (LSTM) for error detection and prediction in QC operations. Applies SMOTE to address class imbalance. Achieves high precision and accuracy, though recall is limited due to short data span and sparse fault labels.
[18]	Predicts QC breakdowns using explainable AI with operational and environmental features, employing global and local SHAP assessments and nested cross-validation across classifiers. Identifies key contributing factors and achieves up to 83% accuracy, supporting proactive fault mitigation.

An overview of these recent works and their methodological distinctions is presented in Table II. It shows that most prior ML-based approaches use binary classification to identify specific faults [14], [20] in QC subsystems or to predict the occurrence of general downtimes [18]. However, previous research also highlights that QCs are highly complex computerized systems composed of multiple subsystems, such as the electric control system [11], [12] and the hoist system [13], [14]. This complexity requires not only the prediction of general downtime or monitoring of a specific subsystem, but also the sequential identification of critical errors to enable component-specific predictive maintenance. Furthermore, predicting the timing of such errors is essential for data-driven decisions regarding the extent of preventive maintenance actions, or whether a crane stop is necessary if insufficient time remains.

Consequently, our contributions can be summarized as follows:

1) **Real-world data integration:** We utilize two years of QC monitoring data from three QCs, enriched with terminal operations and weather context, constructing a

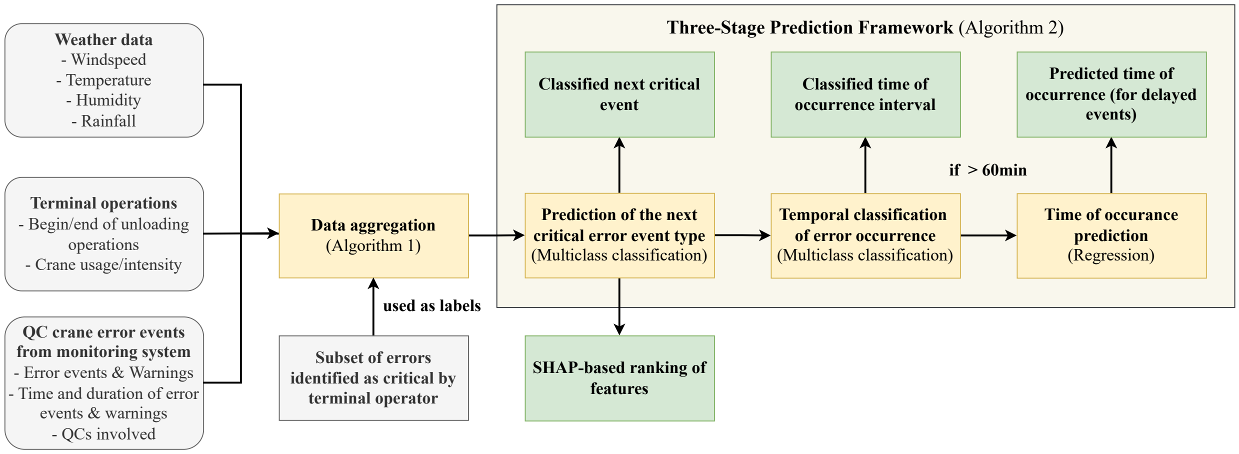

Fig. 2: The overall methodological framework for predicting QC error type and timing.

real-world, data-driven foundation for predictive maintenance decision-making.

2) **Sequential error prediction:** We propose a supervised machine learning framework centered around Extreme Gradient Boosting to predict the next occurring error type, enabling component-specific proactive maintenance planning.
3) **Timely occurrence prediction:** Based on the predicted error type, we estimate its time of occurrence to assess whether proactive maintenance is feasible or if a crane stop is required due to insufficient lead time.

III. Methodology

This section presents the methodological framework used to predict the type and timing of critical QC error events. First, it outlines the problem context (III-A). Then, it describes the related data and feature engineering process (III-B) and introduces the chosen classifier, eXtreme Gradient Boosting (III-C). Finally, it provides a detailed description of the proposed three-stage prediction framework (III-D) and outlines performance evaluation (III-E). Figure 2 depicts a concise overview of our overall methodological framework.

A. Problem Context and Terminal Layout

To contextualize the prediction task, Figure 3 illustrates the layout of the container terminal system from which the data is retrieved, forming the basis of our analysis.

The terminal consists of a quay equipped with three rail-mounted QCs used for loading and unloading containers from vessels. Each crane can be repositioned along the berth to optimize operational efficiency. Depending on the vessel's cargo and the availability of cranes, one, two, or all three QCs may be scheduled for a given operation. These operations can range in duration from a few hours to several days. A comprehensive overview of the container terminal operations depicted in Figure 3 is provided in [21].

B. Data aggregation and feature engineering

The QC monitoring system data, denoted as D_{QC}, covers two full years (2023 and 2024) of operational logs from

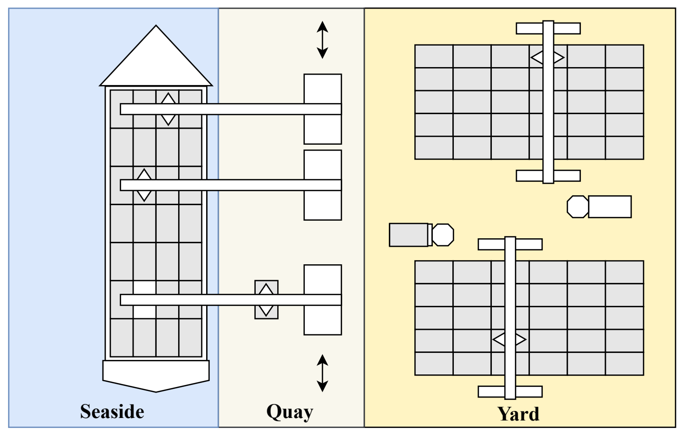

Fig. 3: Schematic of the terminal layout used in this study, showing three QCs servicing a vessel and the adjacent yard area.

the three QCs illustrated in Figure 3. To ensure that only operationally relevant events are included, we exclude error events recorded during maintenance periods or periodic safety tests. Specifically, we focus on error events that occurred during active ship servicing operations.

Based on a prioritisation workshop with the terminal operator Yilport Gävle, we further restrict the prediction targets to the 20 most operationally disruptive error types. However, when constructing lag features and analyzing event sequences, we include all prior events, regardless of type, to capture their potential influence on subsequent critical errors.

This filtered dataset forms the foundation of our analysis. For each error event in D_{QC}, contextual terminal operations data from D_{terminal} are synchronized based on timestamps to capture crane movements and terminal operations characteristics during unloading operations. Additionally, weather data D_{weather} from the Swedish Meteorological and Hydrological Institute (SMHI), collected at 15-minute intervals, are aligned with each error event by assigning the temporally closest weather observations.

The full data aggregation and feature engineering process is depicted in Algorithm 1. An overview of the final engineered features, including their role as label or features, are depicted in Table III. Columns are grouped by feature category, including lagged event history, operational context, and environmental conditions.

Algorithm 1 Data Alignment and Feature Engineering for Predicting Citical QC Events

Require: D_{QC}: QC event logs with timestamps, crane IDs, and event types; D_{weather}: Time-stamped weather data; D_{terminal}: Operational context (e.g., vessel ID, QC moves/hour); E_{top}: Set of critical events to be predicted
Ensure: Feature matrix $X \in \mathbb{R}^{n \times m}$ and target vector $y_{\text{event}} \in E_{\text{top}}^n$, where each row corresponds to a critical event

Step 1: Event Filtering and Target Definition
Select events occurring during servization from D_{QC}
Identify critical events $e_i \in E_{\text{top}}$ to serve as prediction targets
For each e_i, define $y_{\text{event}}[i]$ as the next occurring critical event
Retain all prior events (critical and non-critical) for lag feature construction
Step 2: Temporal Alignment
for each event e_j in D_{QC} **do**
  Retrieve crane and operation context o_j from D_{terminal}
  Align e_j with nearest weather observation w_j from D_{weather}
  Construct enriched event record $r_j = \{e_j, o_j, w_j\}$
end for
Step 3: Lag Feature Construction
for each critical event $e_i \in E_{\text{top}}$ **do**
  Identify crane c_i associated with e_i
  Retrieve previous 5 events $\{e_{i-1}, \ldots, e_{i-5}\}$ for crane c_i (any event type)
  Compute lag features:
  - Inter-event times: $\Delta t_{i-j} = t_{i-j} - t_{i-j-1}$ for $j = 1$ to 5
  - Binary flag indicating whether each previous event is critical: $f_{i-j} = 1$ if e_{i-j} is in E_{top}, otherwise $f_{i-j} = 0$
  - Usage aggregates: mean, standard deviation, max, and recency-weighted usage

  Construct lag feature vector L_i
end for
Step 4: Dataset Assembly
for each critical event $e_i \in E_{\text{top}}$ **do**
  Combine enriched event record r_i and lag features L_i into feature vector x_i
end for
Encode categorical variables and standardize numeric ones
Assemble feature matrix $X = [x_1, x_2, \ldots, x_n]$
return (X, y_{event})

TABLE III: Structure of the feature matrix X for subsequent critical QC event prediction.

Category	Type	Description
Target Variable		
Event label	Categorical	Encoded next critical event
Lagged Event Features (Past 5 Events)		
Event lag 1–5	Categorical	Event type of the i-th previous event (all events)
Critical flag (lag 1–5)	Binary	1 if i-th previous event was critical
Usage lag 1–5	Numeric	Crane usage metric at i-th previous event
Duration lag 1-5	Numeric	Event duration of i-th previous event
Inter-event time lag 1–5	Numeric	Time elapsed since previous events
Aggregated Lag Features		
Usage mean/std/max	Numeric	Statistical aggregates of usage over last 5 events
Recency-weighted usage	Numeric	Exponentially decayed average with $\alpha = 0.6$
Operational Context		
Crane ID	Categorical	Unique crane identifier (OneHot encoded)
Ship ID	Categorical	Vessel being serviced during event
Weather and Environmental Conditions		
Wind speed	Numeric	Wind speed at event time (m/s)
Humidity	Numeric	Min/Mean/Max wind relative humidity (%)
Temperature	Numeric	Min/Mean/Max temperature (°C)
Rainfall	Numeric	Min/Mean/Max rainfall (mm/h)

C. eXtreme Gradient Boosting (XGBoost)

Gradient boosting builds an ensemble of decision trees in a stage-wise manner, where each successive tree aims to correct the residual errors of the preceding ensemble [22]. Among gradient boosting methods, eXtreme Gradient Boosting (XGBoost) has emerged as one of the most efficient and accurate implementations, achieving state-of-the-art results across numerous classification and regression benchmarks, including top rankings in Kaggle competitions [23].

XGBoost distinguishes itself from traditional gradient boosting machines through several innovations, including built-in regularization, optimized handling of sparse data, and support for parallel and distributed computation, all of which contribute to its scalability and robustness [22], [24]. In this study, we thus adopt XGBoost due to its strong predictive performance, flexibility, and proven generalizability across diverse datasets.

D. Prediction framework

To realistically capture both the type and timing of the next critical error event, we formulate two multi-class classification problems. The first task is to predict the next critical event, denoted as $\hat{e}_{t+1}$, by classifying it among the 20 most disruptive error types identified by the terminal operator. This prediction is based on prior lag events, historical QC usage, past weather conditions, and current terminal and environmental context.

Once the next event $\hat{e}_{t+1}$ is classified as one of the critical events in E_{top}, it serves as input for the second task: predicting its time of occurrence. This is formulated as a multi-class classification problem with four time intervals: 0–15, 15–30, 30–60, and 60+ minutes. If the predicted time interval exceeds 60 minutes, we assume that proactive maintenance is feasible. In such cases, we apply an XGBoost regressor to estimate the exact time of occurrence, enabling data-driven decisions regarding the extent of preventive maintenance actions prior to the event. The full procedure is presented in Algorithm 2.

Algorithm 2 Three-Step Prediction Framework

Require: Historical crane monitoring data D with events, timestamps, and features
Ensure: Predicted next event type $\hat{e}_{t+1}$ and time $\hat{t}_{t+1}$

Preprocess D (see Algorithm 1):
Train classifier C_{event} for next-event type
Predict $\hat{e}_{t+1} = C_{event}(x_t)$
Train classifier C_{time} for time-to-next-event bins:
  {0–15, 15–30, 30–60, 60+ minutes}
Predict $\hat{c}_{t+1} = C_{time}(x_t, \hat{e}_{t+1})$
if $\hat{c}_{t+1} = 60+$ minutes **then**
  Train regressor R_{60+} on long-horizon samples
  Predict continuous time $\hat{t}_{t+1} = R_{60+}(x_t, \hat{e}_{t+1})$
else
  Set $\hat{t}_{t+1}$ to bin midpoint
end if
return $(\hat{e}_{t+1}, \hat{t}_{t+1})$

E. Performance evaluation and feature explainability

To evaluate model performance, the dataset was split into training and testing subsets using stratified sampling to preserve the distribution of error types. An 80/20 split was applied, and a fixed random seed ensured reproducibility. The classification tasks were assessed using confusion matrices and accuracy scores, which measure the proportion of correct predictions out of the total. For the regression task, the coefficient of determination (R2) was used to evaluate how well the model explains the variance in the target variable. To identify the most influential features driving the prediction of the next error event type, we employed SHapley Additive exPlanations (SHAP) [25], a game-theoretic approach that assigns an importance value to each feature for a given prediction, enhancing model interpretability.

IV. Results

The results of the first classification task, predicting the next critical error event among 20 candidate classes, are presented in Figure 4. The confusion matrix shows that XGBoost correctly classifies the majority of events, particularly those with sufficient sample sizes. However, events with fewer training samples tend to be misclassified, which is expected in imbalanced multi-class settings. Despite this, the model achieves an overall accuracy of 0.83, which is relatively high for a 20-class classification problem, indicating that the classifier effectively learns from the available data.

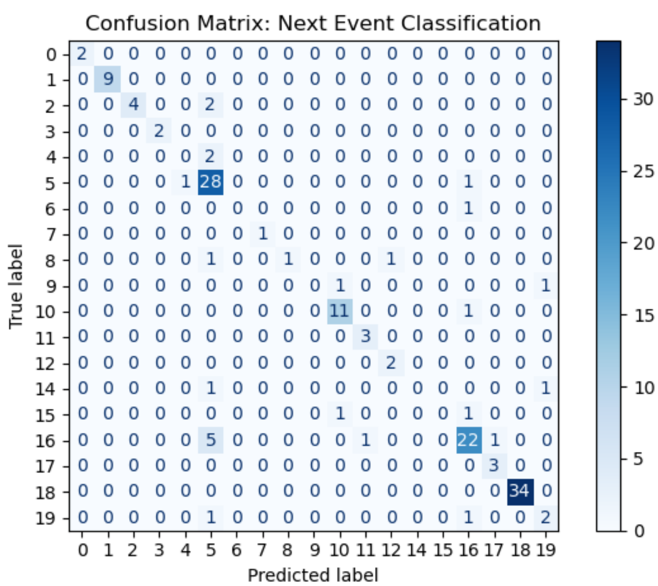

Fig. 4: Confusion matrix for subsequent error event classification. Accuracy = 0.83.

To better understand the model's decision-making process, Figure 5 presents the top features driving the classification, as identified using SHapley Additive exPlanations (SHAP). Among all candidate features, the most influential is the previous error event, particularly the one occurring directly before the current prediction (lag 1 feature). Temporal features such as the time difference between recent errors and the accumulated error count over the past 6 and 24 hours also contribute significantly to the model's decisions. Additionally, weather conditions, especially temperature, wind speed, and humidity, exhibit a notable impact on the occurrence of subsequent error events. Lastly, operational metrics such as the number of QC moves are also associated with the predicted error type.

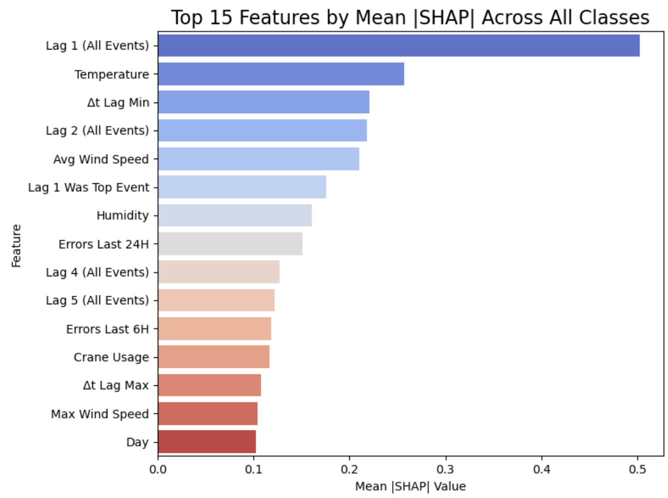

Fig. 5: Overview of the top 15 most influential features based on SHAP values across all classes in Figure 4.

The second classification task focuses on predicting the time interval until the next critical event. The results, shown in Figure 6, indicate that the model performs well in identifying events that occur either immediately (0–15 minutes) or after a longer delay (60+ minutes). These two classes appear to be more distinguishable based on the input features. However, intermediate time intervals (15–30 and 30–60 minutes) are more challenging to classify accurately, likely due to the lower number of training samples. The overall accuracy across the four time classes is 0.71, suggesting that the model captures temporal patterns reasonably well, although further refinement may be needed for mid-range predictions.

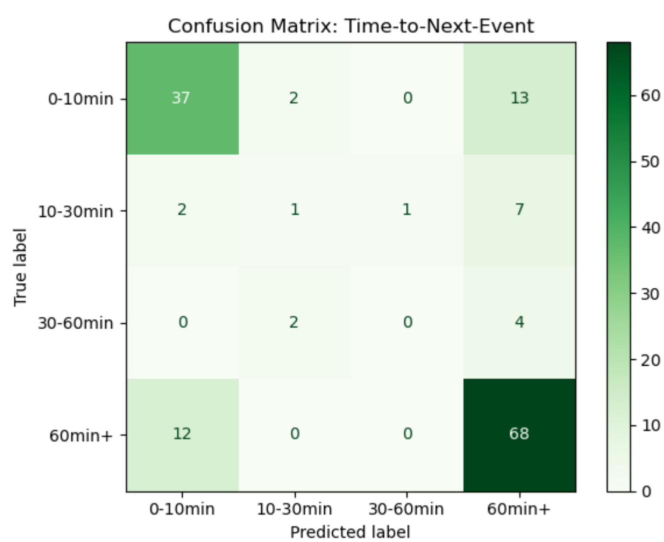

Fig. 6: Confusion matrix for time interval classification of the predicted critical event. Accuracy = 0.71.

To enable more precise temporal predictions, an XGBoost regressor was trained using events classified to occur after 60 minutes in stage two. Figure 7 shows correctly classified samples in blue and classifier false positives in red. This unified visualization illustrates both the regression's accuracy and the amplification of classification errors, showing that, although the model generally performs well, substantial deviations are evident, especially for samples misclassified in stage two. This is reflected in the low R2 value of -0.02. Figure 8 reveals that most prediction errors fall within a few hours and that most misclassifications are caused by stage two false positives. Consequently, the model exhibits a slight tendency to detect upcoming events with a delay.

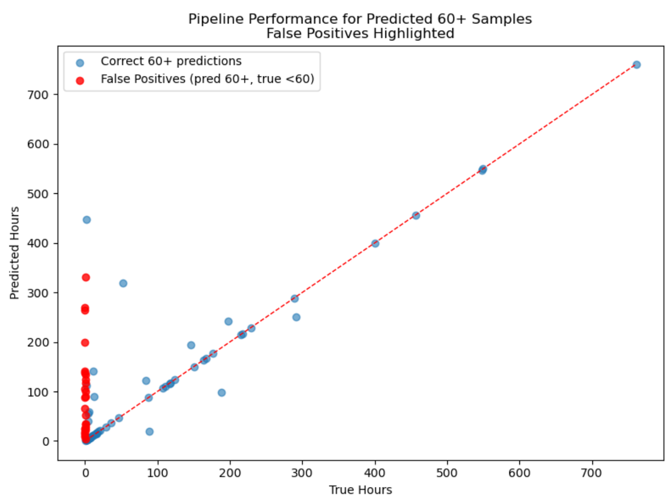

Fig. 7: Prediction of time-to-event for delayed errors (≥60 minutes).

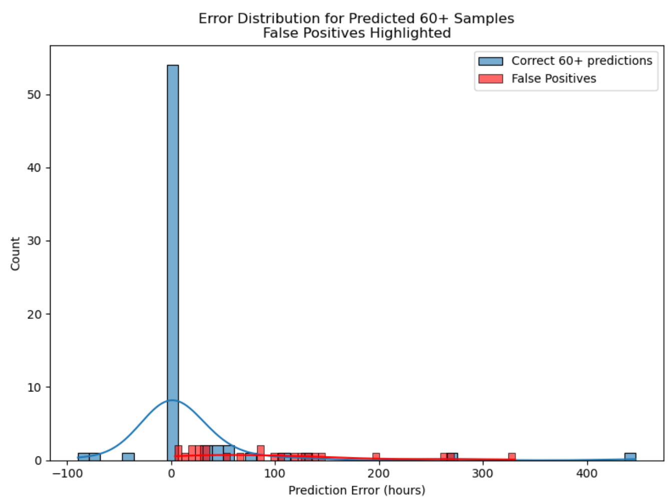

Fig. 8: Histogram of prediction errors for delayed errors (≥60 minutes).

V. Discussion

Ports are pivotal nodes in global supply chain and transportation networks. Despite increasing port digitalization, much of the available data remains underutilized, highlighting the need for more ML-based, data-driven decision-making approaches [15]. One of the most essential pieces of port equipment is the QC, whose uninterrupted operation is vital for the entire port operation chain [4]. However, QCs are still underexploited in terms of data analytics, even as computerization continues to advance.

This study introduces a three-stage predictive framework that leverages two years of QC monitoring data, enriched with operational and weather data for additional context, to forecast subsequent critical error types and their timing. These tasks are formulated as two multi-class classification problems and one regression problem for predicting the exact time of more distant errors. The proposed model achieves an accuracy of 83% for predicting the next error type and 71% for estimating the time interval until its occurrence. However, predicting the exact time of distant errors remains challenging. These results mark a promising step toward reducing QC downtime and enhancing the resilience of port operations. Nonetheless, several limitations and challenges remain.

First, the model assumes that the three QCs operate independently, considering only error events and warnings from the same crane. In practice, however, the downtime of one crane can indirectly affect others due to increased operational loads, which may reduce available maintenance windows. Additionally, QCs operating on the same vessel must maintain safe working distances, and interlock systems can restrict the movement of nearby cranes, potentially influencing error patterns.

Second, in accordance with the terminal operator, we restricted the warnings and error events captured by the QC monitoring systems to those that occurred during ship servicing. This ensures that warnings and error events caused by maintenance actions or testing of critical functions, including emergency event triggers, are not captured. However, in practice, some events may be caused by post-stress in QC components due to heavy loads or severe weather conditions outside of usage. Currently, these events are not captured in the data and are therefore not included in our analysis.

Third, the model assumes that weather conditions and crane usage metrics (e.g., moves per hour) are consistently observed and available at the time of prediction. These features are included as inputs to the model. However, in real-world scenarios, such data may not always be fully known or accurately forecasted, which could lead to slightly optimistic performance estimates.

Currently, SHAP values are averaged across all 20 error event classes to identify the most influential features overall. While this approach provides useful insights, conducting SHAP-based explainability analyses separately for each error class could offer a more granular understanding of feature importance. Additional visualizations, such as beeswarm plots, could illustrate how individual features contribute to predictions in terms of magnitude and direction. Furthermore, examining local explanations, such as feature interactions, partial dependence plots, and instance-level SHAP values, could provide deeper insights into model behavior.

Given that the data in the current aggregation and feature engineering process is tabular and consists of both numeric and categorical input features, making it highly heterogeneous, we used XGBoost. XGBoost also facilitates deeper model and feature insights because feature contribution can be decomposed and visualized. However, future research could benefit from incorporating more temporal dynamics. Models such as recurrent neural networks (RNNs) or long short-term memory networks (LSTMs) may better capture sequential patterns, especially if time-dependent relationships play a critical role in error prediction.

VI. Conclusion

This study introduces a three-stage machine learning framework for predicting subsequent critical error types and their timing in QC operations. Using two years of QC monitoring data enriched with operational and weather information, the model addresses two multi-class classification tasks and one regression problem for error events predicted to occur with a delay ($\geq$60 minutes). It achieves 83% accuracy in predicting the next error type and 71% accuracy in estimating the time interval until its occurrence, while exact predictions for delayed errors remain challenging. SHAP-based explainability is applied to identify key features influencing predictions, highlighting the importance of recent error history, weather conditions, and operational load.

The findings offer actionable insights for terminal operators aiming to reduce QC downtime and improve operational resilience. By anticipating error types and their timing, maintenance planning can be optimized and proactive interventions scheduled. The integration of contextual data, such as weather and operational metrics, into predictive models underscores the value of data-driven decision-making. Furthermore, explainable AI techniques like SHAP enhance transparency, supporting trust and adoption in real-world port environments.

This study assumes independent operation of QCs and excludes maintenance-related events and those occurring outside ship servicing, which may overlook interdependencies and stress-induced errors. Additionally, the reliance on consistently available weather and operational data may lead to optimistic predictions. Future research should incorporate crane interactions, consider events and conditions beyond active crane usage, and apply time-series modeling techniques such as LSTMs to better capture sequential patterns. Finally, event-specific SHAP analyses could yield deeper insights into the drivers of individual error types.

References

[1] K.-H. Kim, *Planning and Operation of Container Terminals*. Elsevier, 2024.

[2] R. Klar and V. Angelakis, “Linking ports’ digital twins to those of cities,” in *2023 IEEE Smart World Congress (SWC)*. IEEE, 2023, pp. 1–8.

[3] T. Wang, G. Gao, K. Wang, and J. Shi, “Integrated operation models with quay crane maintenance in a container terminal,” *Ocean & Coastal Management*, vol. 251, p. 107101, May 2024.

[4] R. Klar, A. Fredriksson, and V. Angelakis, “Digital Twins for Ports: Derived From Smart City and Supply Chain Twinning Experience,” *IEEE Access*, vol. 11, pp. 71 777–71 799, 2023.

[5] T. Notteboom, A. Pallis, and J.-P. Rodrigue, *Port Economics, Management and Policy*, 1st ed. London: Routledge, Dec. 2021.

[6] H. J. Carlo, I. F. A. Vis, and K. J. Roodbergen, “Seaside operations in container terminals: literature overview, trends, and research directions,” *Flexible Services and Manufacturing Journal*, vol. 27, no. 2-3, pp. 224–262, Sep. 2015.

[7] Y. Li, F. Chu, F. Zheng, and I. Kacem, “Integrated Berth Allocation and Quay Crane Assignment With Uncertain Maintenance Activities,” in *2019 International Conference on Industrial Engineering and Systems Management (IESM)*. Shanghai, China: IEEE, Sep. 2019, pp. 01–06.

[8] T. Li, Q. Dong, and X. Sun, “Integrated Scheduling of Handling Equipment in Automated Container Terminal Considering Quay Crane Faults,” *Systems*, vol. 12, no. 11, p. 450, Oct. 2024.

[9] K. Chargui, T. Zouadi, A. El Fallahi, M. Reghioui, and T. Aouam, “A quay crane productivity predictive model for building accurate quay crane schedules,” *Supply Chain Forum: An International Journal*, vol. 22, no. 2, pp. 136–156, Apr. 2021.

[10] Ü. Atak and Y. Arslanoğlu, “Machine learning methods for predicting marine port accidents: a case study in container terminal,” *Ships and Offshore Structures*, vol. 17, no. 11, pp. 2480–2487, Nov. 2022.

[11] S. P. A. Gothandapani, M. N. Ab Rahman, and H. Hishamuddin, “Quay crane performance improvement and lifecycle extension: Retrofit determination–a case study,” *Jurnal Kejuruteraan*, vol. 36, no. 4, pp. 1483–1493, 2024.

[12] B. A. Putra, L. M. Putranto, and R. Irnawan, “Early Failure Detection of Quayside Container Crane Using the IoT Based Measurement Data,” in *2024 International Conference on Technology and Policy in Energy and Electric Power (ICTPEP)*. Bali, Indonesia: IEEE, Sep. 2024, pp. 297–302.

[13] M. R. Jalal, A. S. A. Kader, M. F. A. Hamid, and H. S. Kang, “A stochastic Petri Net-based approach for operational performance estimation of quay cranes,” *Quality and Reliability Engineering International*, vol. 39, no. 5, pp. 1660–1680, 2023.

[14] A. Mukherjee, M. Sasidharan, M. Herrera, and A. K. Parlikad, “Unsupervised constrained discord detection in iot-based online crane monitoring,” *Advanced engineering informatics*, vol. 60, p. 102444, 2024.

[15] S. Filom, A. M. Amiri, and S. Razavi, “Applications of machine learning methods in port operations – A systematic literature review,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 161, p. 102722, May 2022.

[16] Y. Li, S. Li, Q. Zhang, B. Xiao, and Y. Sun, “Application of Big Data Technology in Ship-to-Shore Quay Cranes at Smart Port,” *Infrastructures*, vol. 7, no. 5, p. 73, May 2022.

[17] A. Crespo Del Castillo, M. Sasidharan, C. Nentwich, J. Merino, and A. Kumar Parlikad, “Data-Driven Asset Health Index – an application to evaluate Quay Cranes in container ports,” *Maritime Policy & Management*, vol. 51, no. 8, pp. 1805–1823, Nov. 2024.

[18] R. Klar, A. Andersson, and V. Angelakis, “Understanding and Predicting Quay Crane Breakdowns Using Explainable AI,” 2026, under review.

[19] Y. Li, F. Chu, F. Zheng, and M. Liu, “A Bi-Objective Optimization for Integrated Berth Allocation and Quay Crane Assignment With Preventive Maintenance Activities,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 2938–2955, Apr. 2022.

[20] A. Awasthi, L. Krpalkova, and J. Walsh, “Deep learning-based boolean, time series, error detection, and predictive analysis in container crane operations,” *Algorithms*, vol. 17, no. 8, p. 333, 2024.

[21] R. Klar, A. Andersson, A. Fredriksson, and V. Angelakis, “Container Relocation and Retrieval Tradeoffs Minimizing Schedule Deviations and Relocations,” *IEEE Open Journal of Intelligent Transportation Systems*, vol. 5, pp. 360–379, 2024.

[22] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[23] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, “A comparative analysis of gradient boosting algorithms,” *Artificial Intelligence Review*, vol. 54, no. 3, pp. 1937–1967, 2021.

[24] P. Zhang, Y. Jia, and Y. Shang, “Research and application of xgboost in imbalanced data,” *International Journal of Distributed Sensor Networks*, vol. 18, no. 6, p. 15501329221106935, 2022.

[25] S. M. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” *Advances in neural information processing systems*, vol. 30, pp. 4765–4774, 2017.