Bayesian long short-term memory model for crane stability prediction and uncertainty quantification in safety monitoring systems*

Bohdan Solovei[1,*,†], Oleksandr Terentyev[2,†]

[1,2] *Kyiv National University of Construction and Architecture, 31, Air Force Avenue, Kyiv, 03037, Ukraine*

Abstract

This study proposes a Bayesian Long Short-Term Memory model for forecasting crane stability and quantifying its inherent uncertainty, aiming to enhance crane safety monitoring systems. The model's performance was evaluated through training and validation loss curves, with the training loss consistently decreasing from 0.538 to 0.02. The validation accuracy reached a maximum of 0.97 within 150 epochs. Additionally, the model's prediction uncertainty was quantified, revealing that it exhibited high confidence in the majority of predictions, while flagging a smaller subset of highly uncertain cases. A safety alert mechanism based on prediction uncertainty was developed, showing a clear relationship between the required confidence level and the number of safety alerts generated. As the confidence threshold increased from 80% to 95%, the system became more sensitive, with a notable spike in alerts at the 97.5% confidence level. This demonstrates the potential of tuning the crane safety system's sensitivity to meet specific operational needs.

Keywords

Bayesian Long Short-Term Memory model, alert notification model, stability index of a monorail crane, z-score, Gaussian distribution

1

1. Introduction

Mobile boom cranes play an indispensable role in modern civil engineering, particularly in dense urban construction sites and complex industrial settings. However, the operational flexibility that makes these machines so valuable also introduces significant safety challenges. To ensure the safe operation of any mobile boom crane, its stability must be guaranteed, eliminating any possibility of tipping over, even when subjected to the most unfavorable combinations of load and environmental factors [1]

The stability of a mobile boom crane is quantified by its stability coefficient. This coefficient is typically defined as the ratio of the stabilizing moments to the overturning moments generated by the load, the boom's self-weight, and external forces [2]. The

*ITTAP'2025: 5th International Workshop on Information Technologies: Theoretical and Applied Problems, October 22–24, 2025, Ternopil, Ukraine, Opole, Poland

[1*] Corresponding author.

[1†] These authors contributed equally.

solovei_ba-2023@knuba.edu.ua (B. Solovei); terentiev.oo@knuba.edu.ua (O. Terentyev)

0009-0008-0328-1123 (B. Solovei); 0000-0001-9499-6635 (O. Terentyev)

permissible values for crane stability coefficients are strictly regulated by the State Labor Service of Ukraine [3]. Therefore, accurate, real-time prediction and monitoring of the stability coefficient are the basis of any modern crane safety monitoring system.

In response to the need for proactive safety measures, recent advancements in artificial intelligence have led to the use of Recurrent Neural Networks (RNNs) and their more advanced variants, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models, for crane safety assessments. These deep learning architectures are uniquely suited to this task due to their inherent ability to model and learn from sequential, time-series data [4].

However, a fundamental limitation is inherent in these standard recurrent models: they are deterministic. While an LSTM or GRU can predict the most likely future stability coefficient, it produces only a single point-estimate, offering no insight into the model's confidence or the range of other plausible outcomes [5]. This is a significant shortcoming for safety-critical systems operating in unpredictable real-world environments. For example, a crane subjected to a sudden, unexpected impact may not be perfectly captured by the model's training data. In such scenarios, a deterministic prediction could be dangerously misleading.

This gap necessitates a probabilistic approach to risk assessment that treats the network's parameters as probability distributions rather than fixed values. A model capable of quantifying its own uncertainty would allow a safety system to make decisions based on a formal risk calculation, triggering an alert if the probability of failure exceeds a critical threshold.

2. Literature Review

Today, safety monitoring systems for building cranes increasingly rely on IoT streaming data combined with AI models capable of effectively processing this data and producing reliable, real-time results. The prediction of time-dependent parameters in engineering systems has been significantly advanced by the application of the Long Short-Term Memory (LSTM) model.

In study [6], an LSTM-based Recurrent Neural Network (RNN) model was developed to successfully capture the complex, non-linear dynamics of a crane system, aiming to mitigate undesired oscillations in overhead crane operations used in material handling. The proposed model architecture includes two LSTM layers with 64 neurons each, designed to learn patterns from time series data. The output of the second LSTM layer is fed into a fully connected layer with 128 neurons using ReLU activation. The final output layer, with a single neuron, returns a continuous value representing the predicted swing angle.

In study [7], an LSTM model was trained using real operational error data from container cranes to enhance error recognition and prediction in the multibillion-dollar container shipping industry. The model architecture consists of two LSTM layers and a final dense output layer. It was compiled using a binary cross-entropy loss function, the Adam optimizer, and accuracy as the evaluation metric. Training was conducted with a batch size of 50 over 40 epochs.

In study [8], an LSTM model was used in a hybrid solution alongside a Convolutional Neural Network (CNN) for a real-time crane automation system, designed to operate under challenging visual conditions. The system employs GoogLeNet, a deep CNN, to extract rich visual features from input images. These features are then passed to an RNN with LSTM units.

The LSTM acts as a controller, processing temporal sequences to guide the final detection of corner castings.

As shown in studies [6-8], those approaches employed an LSTM model based on RNN architecture (Figure 1) which for a sequence of input vectors $X=\{x_1,x_2,\ldots,x_t\}$ iteratively computes a hidden state h_t that depends on the current input x_t and the previous hidden state h_{t-1} [7] The computation of the hidden state h_t at each time step t is defined by:

$$h_t = \tan h\left(W_{xh}x_t + W_{hh}h_{t-1} + b_h\right) \quad (1)$$

The output y_t is generated from the hidden state via another transformation:

$$y_t = \tan h\left(W_{hy}h_t + b_y\right) \quad (2)$$

where W_{xh}, W_{hh}, W_{hy} represent the input-to-hidden, hidden-to-hidden, and hidden-to-output weight matrices, respectively; b_h, b_y are biases terms; the tanh(x) function returns the hyperbolic tangent of the input.

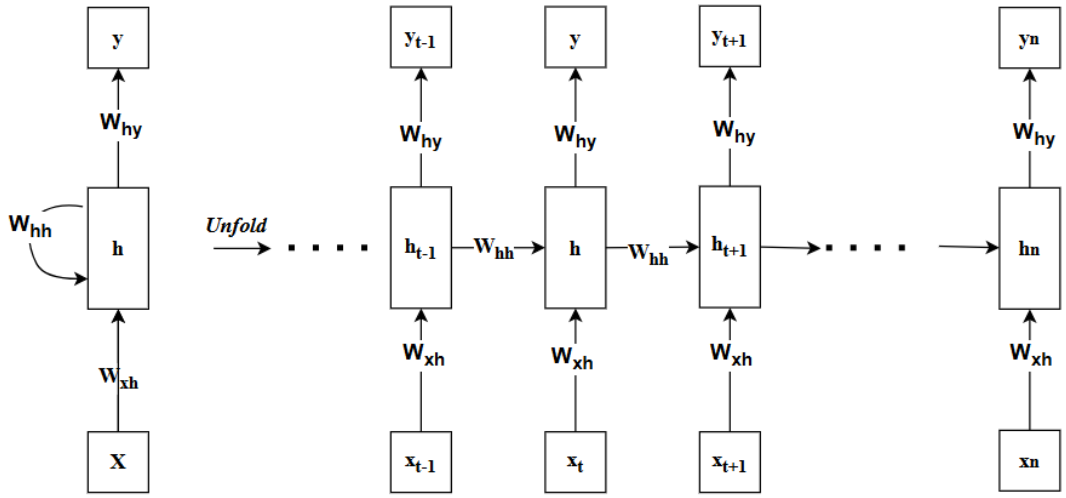

Figure 1: RNN model with a single hidden layer

The standard approach for training LSTMs on long sequences is Truncated Backpropagation Through Time technique (TBTT), which involves breaking the sequence into smaller chunks for gradient computation (Figure 2) [9].

LSTM based on RNN incorporates a "gating mechanism" - specialized neural network components that regulate the flow of information through the cell (Figure 3). Its architecture employs three distinct gates: an input gate to control which new information is stored, a forget gate to discard irrelevant memories from the cell state, and an output gate to determine what information influences the next hidden state. This intricate structure allows LSTMs to effectively capture and maintain information over extended time horizons, making it a standard for complex sequence modeling tasks [10].

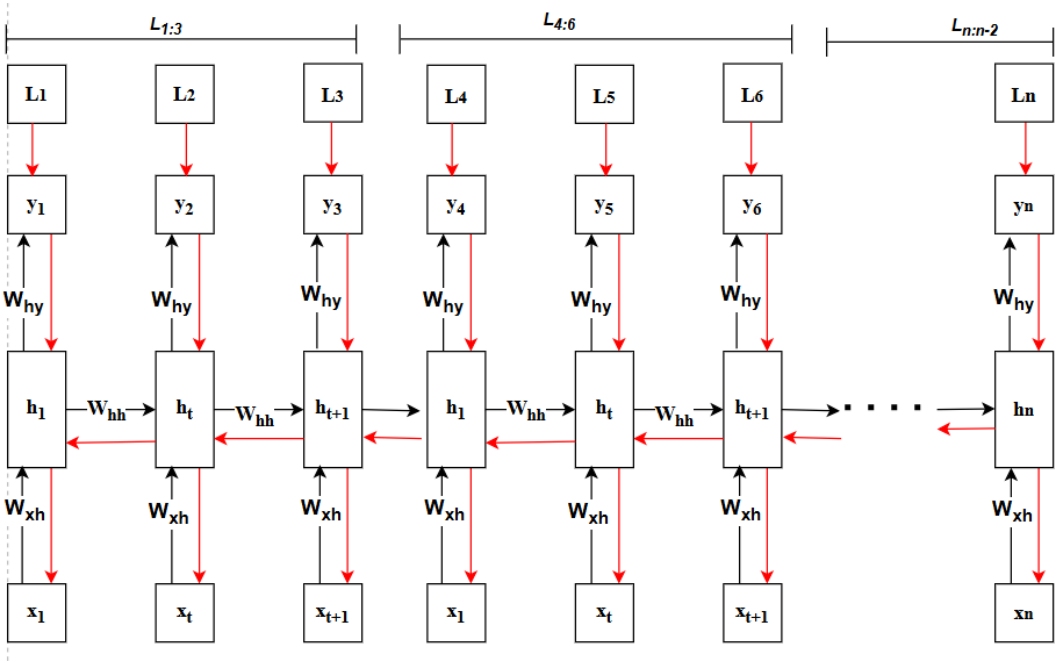

Figure 2: TBTT technique for length of sequence T=m, with chunk value k=3

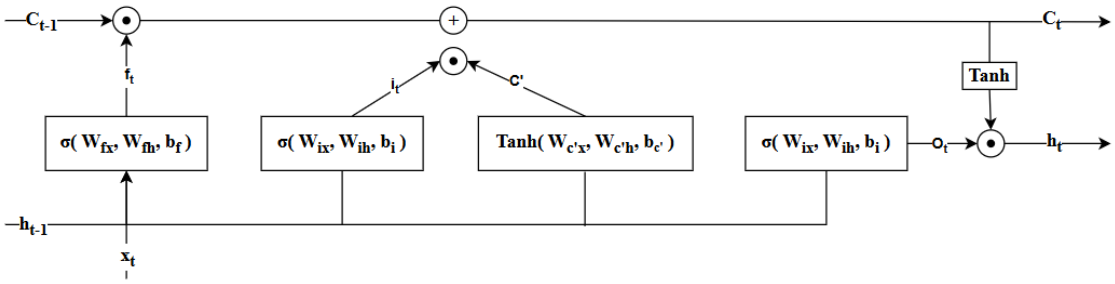

Figure 3: An architecture of a single cell of the LSTM model

LSTM forget gate ft by applying a sigmoid function σ identifies which parts of the previous cell state should be retained. The output is in the range [0, 1]:

$$f_t = \sigma\left(W_{fx} x_t + W_{fh} h_{t-1} + b_f\right), \tag{3}$$

where W_{fx},W_{fh} are the weight matrices for the forget gate; x_t is the current input at time step t; h_{t-1} is the previous hidden state, b_f is the bias vector of the forget gate.

The input gate i_t also applies a sigmoid function to determine how much of the new input should be written to the cell state:

$$i_t = \sigma\left(W_{ix} x_t + W_{ih} h_{t-1} + b_i\right), \tag{4}$$

where W_{ix},W_{ih} are the weight matrices for the input gate; b_i is the bias vector for the input gate.

A new candidate cell state $c_t^{'}$ is generated using the tanh function, which proposes potential new values to add to the cell state:

$$c_t^{'} = \tan h\left(W_{c'x} x_t + W_{c'h} h_{t-1} + b_{c'} \right), \quad (5)$$

where $W_{c'x}$, $W_{c'h}$ are weight matrices for the candidate cell state; $b_{c'}$ is the bias vector of the candidate state.

The updated cell state C_t combines the previous state (modulated by the forget gate) and the new candidate (modulated by the input gate):

$$C_t = \left(C_{t-1} \odot f_t \right) \oplus \left(i_t \odot c_t^{'} \right), \quad (6)$$

where $\odot$, $\oplus$ are elementwise multiplication and sum respectively.

The output gate O_t controls how much of the cell state is output:

$$O_t = \sigma\left(W_{ox} x_t + W_{oh} h_{t-1} + b_o \right), \quad (7)$$

where W_{ox}, W_{oh} are output gate weights, and b_o is the bias vector.

The final hidden state h_t at time step t is computed as:

$$h_t = O_t \odot \tan h\left(C_t \right), \quad (8)$$

Based on equations (3–8), it is evident that while an LSTM can predict the most likely future value, it produces only a single point estimate. This offers no information about the model's confidence, which makes the solutions in studies [2–4] less reliable for safety-alerting systems.

In the current study, we propose a hybrid Bayesian LSTM model for building crane safety monitoring systems. A Bayesian LSTM model has the advantage of being able to quantify its own uncertainty. This will allow the safety monitoring system to make decisions based on formal risk calculations, such as triggering an alert if the probability of the stability coefficient falling below a critical threshold exceeds a specified tolerance.

3. Research Objective

The goal of this research is to develop a Bayesian Long Short-Term Memory (LSTM) model to forecast crane stability and quantify its own uncertainty. Together, those components will allow a crane's safety monitoring system to trigger an alert if the probability of failure exceeds a critical threshold.

To achieve the goal this study will resolve the tasks:

- Design an architecture of Bayesian LSTM model for predicting crane stability.
- Create a mathematical model for the stability index of a monorail crane.
- Define an alert notification model for crane stability monitoring.
- Train and evaluate Bayesian Long Short-Term Memory (LSTM) model.
- Evaluate the alert notifications model.

4. Materials and methods

4.1. Bayesian LSTM model for predicting crane stability formalization

Bayesian LTSM extends the standard LSTM architecture by treating all model parameters as random variables, allowing the network to capture and quantify its predictive uncertainty. In our research, we assumed a standard Gaussian prior to treat the weights W as random

variables governed by probability distributions, the sampled weight matrices for all gates and biases (Figure 3) are:

$$W_x = \begin{bmatrix} W_{fx} \\ W_{ix} \\ W_{c'x} \\ W_{ox} \end{bmatrix} = \mu_x + \sigma_x \cdot \epsilon_x = \mu_x + softplus(\rho_x) \cdot \epsilon_x \quad (9)$$

$$W_h = \begin{bmatrix} W_{fh} \\ W_{ih} \\ W_{c'h} \\ W_{oh} \end{bmatrix} = \mu_h + \sigma_h \cdot \epsilon_h = \mu_h + softplus(\rho_h) \cdot \epsilon_h \quad (10)$$

$$b = \mu_b + softplus(\rho_b) \cdot \epsilon_b \quad (11)$$

where ϵ_x, ϵ_h, $\epsilon_b \backsim \mathcal{N}(0,I)$ is Gaussian noise for sampling, μ_x, $\mu_h \in R^{4H*D}$ and $\mu_b \in R^{4H}$ are mean of the weight distribution, ρ_x, $\rho_h \in R^{4H*D}$ and $\rho_b \in R^{4H}$ are learnable parameters which are used to parametrize the standard deviations $\sigma_x, \sigma_h, \sigma_b$ of the weight distribution in Bayesian neural network based on softplus function:

$$\sigma = \text{softplus}(\rho) = \log(1+e^{\rho}), \quad (12)$$

For $\rho \to +\infty$, softplus(ρ) $\to \rho$, so uncertainty is large; for $\rho \to -\infty$, softplus(ρ) $\to 0^+$, so uncertainty becomes negligible, but always positive.

To train the Bayesian LSTM model on long time-series data and limit gradient computation to a fixed time window, we employ Truncated Backpropagation Through Time (TBPTT) technique. The full input sequence of length T is split into m chunks of length k: [1,k], [k+1,2k], ..., [T−k+1,T]. The forward pass equations (Eq. 3–8) for each chunk i are defined as:

$$h_t = LSTM(x_t, h_{t-1}, c_{t-1}), \; t \in [T[i], T[i+k-1]] \quad (13)$$

The training objective is to maximize the Evidence Lower Bound (ELBO). To achieve this using standard gradient-based optimizers, is equivalently to minimize the Negative ELBO, which is defined as loss function $\mathcal{L}_{loss}$. For each chunk (i) over the interval $[t_i, t_{i+k-1}]$ the loss is defined as:

$$L_{loss}^{(i)} = -E_{q(W)}\left[\sum_{t=t_i}^{t_i+k-1} logp(y_t | x_t, h_{t-1}, W)\right] + \frac{1}{m} KL(q(W) || p(W)), \quad (14)$$

where $p(y_t|x_t,h_{t-1},W)$ is the likelihood of observing the true target y_t given the model's predictive distribution generated from the input x_t and weights W, q(W) is the learned variational posterior distribution over the weights, p(W) is the fixed prior distribution over the weights, KL(q(W)||p(W)) is a Kullback-Leibler divergence between the posterior and the prior, X and Y are the sequences of input and corresponding target.

The first term measures how well the model's predictions fit the data within chunk. The KL term acts as a regularizer, penalizing q(W) for deviating significantly from the prior p(W), thus controlling model complexity.

The backpropagated gradient $\nabla_W L_{loss}^{(i)}$ over the time window from t to t+k-1 is defined via the chain rule as:

$$\nabla_{\rho_w} L_{loss}^{(i)} = \sum_{j=t}^{t+k-1} \frac{L_{loss}^{(i)}}{\partial W_j} \cdot \frac{\partial W_j}{\partial \rho_W} \tag{15}$$

$$\nabla_{\mu_w} L_{loss}^{(i)} = \sum_{j=t}^{t+k-1} \frac{L_{loss}^{(i)}}{\partial W_j} \cdot \frac{\partial W_j}{\partial \mu_W} \tag{16}$$

The parameters μ_W, ρ_W are updated using gradient ascent:

$$\mu_W^{(i)} = \mu_W^{(i)} - \alpha \nabla_{\mu_w} L_{loss}^{(i)} \tag{17}$$

$$\rho_W^{(i)} = \rho_W^{(i)} - \alpha \nabla_{\rho_w} L_{loss}^{(i)} \tag{18}$$

where α is the learning rate, and the gradients are accumulated across time steps in the chunk.

4.2. Bayesian LSTM model for predicting crane stability architecture

We propose model architecture consists of Bayesian LSTM layer with 32 hidden units, followed by a dense layer with ReLU activation. The final layer is a single-neuron linear layer with no activation function, which acts as a linear head.

The Bayesian properties of the model are achieved through the application of Monte Carlo Dropout. Dropout layers with a rate of 0.3 are applied after the recurrent layer and the dense layer. During inference, dropout is kept active, and we perform multiple stochastic forward passes for each input data point. The mean of the resulting distribution of predictions is taken as the final predicted value (μ), and its standard deviation is used as a measure of model uncertainty (σ).

4.3. The mathematical model of stability index for the monorail crane

The load stability index of a monorail crane in the first operating position, when the boom is aligned along the rail track (Figure 4) and the crane is under load, is defined under the condition in which the restoring and overturning moments are balanced is defined as [11]:

$$\beta = \frac{G_{cr}(k-b) + G_{pr}(c+k) + G_t k}{G_{gr}(a-k)}, \tag{19}$$

where G_{cr} is a crane own weight (kN); G_{gr} is a weight of the load, (kN); G_{pr} is a weight of the counterweight, (kN); G_t is a weight of the trolley with mechanisms; a is a distance from the axis of rotation to the center of gravity of the load, (m); b is a distance from the crane's axis of rotation to its center of gravity, (m); c is a distance from the crane's axis of rotation to the counterweight's center of gravity, (m), k = 0.4·a is distance from the crane's axis of rotation to the running wheel, (m).

Equation (20) assumes static and idealized conditions, without considering dynamic effects, structural deformations, or external forces such as wind. To improve accuracy under real-world operating conditions in our research we apply the finite element method (FEM). In this model, the crane beam is discretized into four two-node beam elements. Each node has two degrees of freedom: vertical displacement and angular rotation. According to FEM theory [12], the load vector for a beam element is defined as:

$$F = \begin{bmatrix} F_1 \\ M_1 \\ F_2 \\ M_2 \end{bmatrix}, \tag{20}$$

where $F_1=G_{cr}\cdot k$ is a vertical force at node 1, $F_2=G_{gr}\cdot a$ is a vertical force at node 2, $M_1= G_{cr}\cdot k\cdot s$ is a moment at node 1, $M_1= G_{gr}\cdot a$ is a moment at node 2, s is a distance from the boom hinge to the center of gravity of the load.

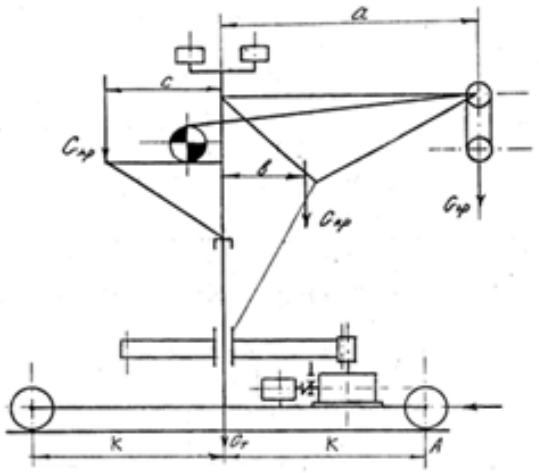

Figure 4: A monorail crane in the first operating position, when the boom is aligned along the rail track

The column vector of unknown nodal displacements and rotations u_f is defined as:

$$u_f = K^{-1} F, \tag{21}$$

where K is the global stiffness matrix (23), assembled from the element of the stiffness matrices k^e (eq. 24):

$$K = \sum_{e=1}^{4} k^e, \tag{22}$$

$$k^e = \frac{EI}{l_e^3} \begin{bmatrix} 12 & 6l_e & -12 & 6l_e \\ 6l_e & 4l_e^2 & -6l_e & 2l_e^2 \\ -12 & -6l_e & 12 & -6l_e \\ 6l_e & 2l_e^2 & -6l_e & 4l_e^2 \end{bmatrix}, \tag{23}$$

where E - Young's modulus, Pa; I is a moment of inertia of the cross-section, m^4;l_e – length of the crane beam, m.

The load stability index β of the crane (Figure 5), accounting for nodal displacements and rotations u_f, is defined as the ratio of stabilizing moments M_{stab} to overturning moments M_{ovt}:

$$\beta = \frac{Mstab}{M\,ovt}, \tag{24}$$

where $M_{stab}=G_{cr}(k-u_1)+G_{gr}(a-u_2)$ is a total stabilizing moment, accounting for beam deformation, $M_{ovt}=G_{gr}(a-k-u_3)$ is overturning moment, adjusted for displacement, u_1,u_2,u_3 – are displacements at relevant nodes (determined from FEM solution u_f)

The stability index S is modeled as a binary threshold function:

$$S=\begin{cases}1, & \frac{Mstab}{M\,ovt}\geq 1.4\\ 0, & \frac{Mstab}{M\,ovt}<1.4\end{cases}. \tag{25}$$

Condition (25) indicates whether the system is stable (1) or unstable (0) based on a safety factor of 1.4, as per engineering standards.

4.4. Alert notifications model for crane stability monitoring

Given that the Bayesian LSTM model predicts the stability index as a Gaussian (normal) distribution, we define:

$$\text{Stability Index} \backsim N(\mu,\sigma^2),$$

where μ is a mean expected stability; σ is the standard deviation (model's uncertainty about stability).

Let the risk of failure be the probability that the predicted stability index falls below a critical threshold τ. The z-score for the threshold τ is computed as [13]:

$$z=\frac{\tau-\mu}{\sigma+\epsilon}, \tag{26}$$

where τ is the critical threshold of stability, $\epsilon=10^{-8}$ is a constant to ensure numerical stability in cases of near-zero variance.

The cumulative probability of failure P_f is calculated using the cumulative distribution function (CDF) for the standard normal distribution Φ(·):

$$P_f=\Phi(z) \tag{27}$$

An alert notification is triggered if the probability of failure exceeds the maximum acceptable risk tolerance level α:

$$A=\begin{cases}1, & if\ \ P_f>\alpha\\ 0\end{cases} \tag{28}$$

The sequence of operations in the alert notification model is illustrated in Figure 5.

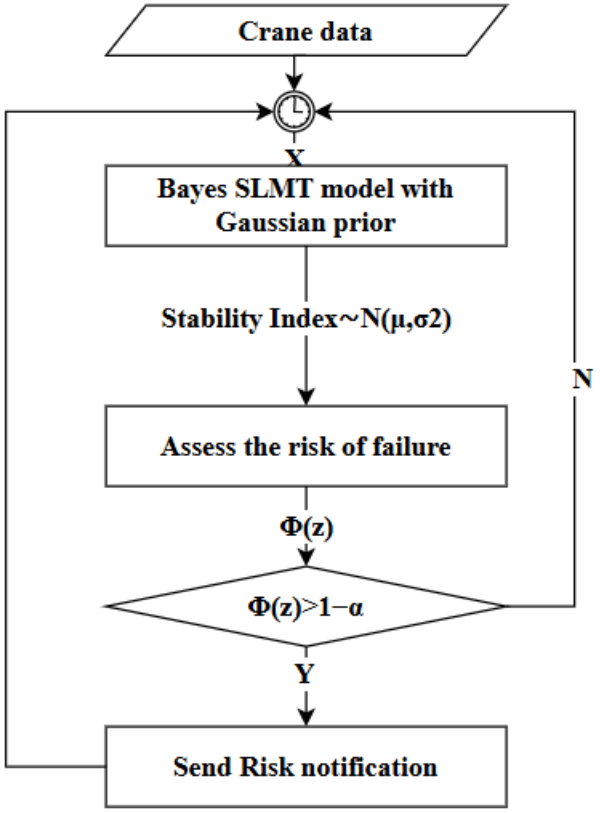

Figure 5: The sequence of operations in the Alert Notification model

It includes the steps:

- At each time step, a new data vector X is collected, representing the current crane state.
- The input X is passed through the pretrained Bayesian LSTM model. The model outputs a Gaussian distribution for the predicted stability index: $N(\mu, \sigma^2)$, where μ is the most likely stability index (the mean), and σ^2 is the variance, which quantifies the model's uncertainty about its prediction.
- The system evaluates the risk that the true stability index is below the critical threshold τ using Equations (27-28).
- If the calculated risk exceeds the defined acceptable limit α, an alert (A=1) is issued. This notification is sent to the automated monitoring system for potential intervention.

5. Experiment preparation

5.1.1. Data Collection and Annotation

The application of the proposed stability index calculation, which is based on FEM theory and defined by the formulas in this study, requires a set of specific input parameters. These

parameters are sourced from various IoT devices installed on the monorail crane. The key simulated devices and their corresponding data schemas are detailed in Table1 -Table2. Each device streams data using the MQTT protocol, which is aggregated by a gateway and published to the central data platform (Figure 7).

Table 2
Data Schema for Position Encoder Devices

Field name	Field description	Data type
deviceId	Unique identifier of the device (e.g., "encoder_01")	String
timestamp	ISO 8601 timestamp equal to time when data is captured	Timestamp
beam_length_m	The total length of the crane's main beam	Float
distance_a_m	Distance from the beam's reference point to the load	Float
distance_k_m	Distance from the crane’s axis of rotation to the running wheel	Float
status	Operational status of the sensor (e.g., "OK", "ERROR")	String

Table 3
Data Schema for Load Cell Devices

Field name	Field description	Data type
deviceId	Unique identifier of the device (e.g., "loadcell_01")	String
timestamp	ISO 8601 timestamp equal to time when data is captured	Timestamp
load_mass_kg	The mass of the payload as measured by the load cell	Float
status	Operational status of the sensor (e.g., "OK", "ERROR")	String

To simulate a comprehensive range of realistic operational conditions, a synthetic dataset comprising 1,000 records with data for parameters (listed in Table 2-3) were sampled from uniform distributions, with the specific ranges chosen to reflect typical monorail crane specifications, as detailed in Table 4.

Table 4
Data ranges to sample from Uniform distribution

Parameter	Distribution	Range
beam_length_m	Uniform	[5.0, 15.0]
distance_a_m	Uniform	[1.0, 14.0]
distance_k_m	Uniform	[4.0, 5.0]
load_mass_kg	Uniform	[100.0, 5000.0]

The generated data was used to emulate a real-time data ingestion pipeline. Each record was serialized into a JSON message format and published to a dedicated Apache Kafka topic

corresponding to its device type (topic-encoders, topic-load-cells). A Kafka consumer application was developed to subscribe to these topics and process the incoming data streams.

The data processing workflow executed by the consumer involved several key stages. First, each incoming message was validated against its corresponding schema (as defined in Tables 2-3) to ensure data integrity. Messages from the different streams were then joined by their timestamp to create a unified record representing the state of the crane at a single point in time.

Following this aggregation, for each unified record, the nodal displacements and rotations (u_1,u_2,u_3,) were calculated according to Equation (22). Subsequently, the total stabilizing moment and overturning moment were computed using Equation (25), and stability index via Equation (26). This fully enriched dataset, containing both the raw sensor values and the derived physical metrics, was stored into structured table. This table served as the complete, supervised dataset for the training, validation, and final evaluation of the Bayesian LSTM model.

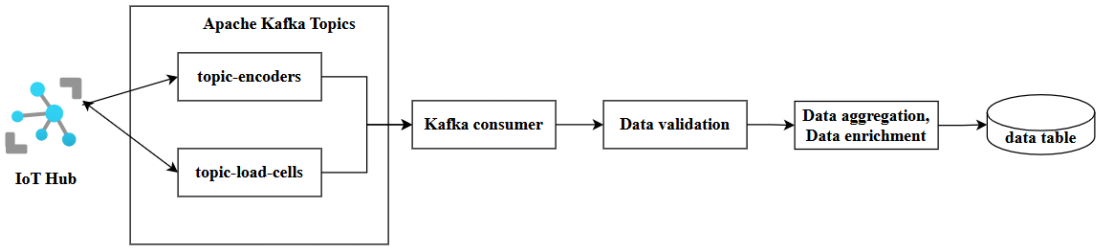

Figure 6: Streaming data collection process for crane safety monitoring system

Figure 7 visualizes the spatial distribution of the stability index (β) of a monorail crane as a function of the stabilizing moments M_{stab} to overturning moments M_{ovt} in the collected dataset. It shows that regions where the stability index β is less than 1.4 correspond to cases where the overturning moment exceeds the stabilizing moment, which is consistent with theoretical mechanical principles governing crane stability.

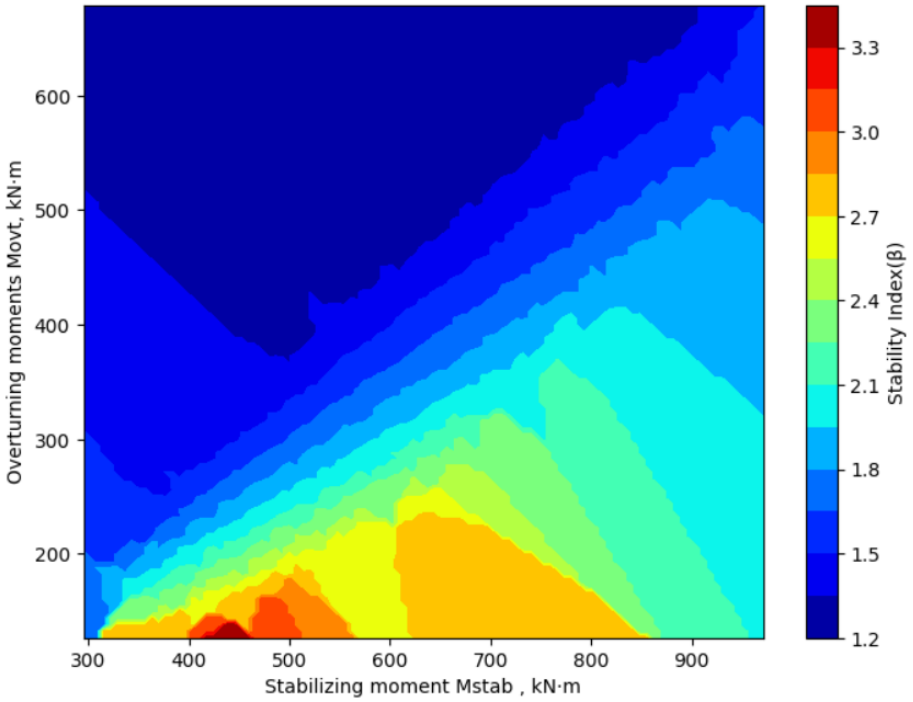

Figure 7: Distribution areas of the stability coefficient of a monorail crane in the dataset

6. Results and discussion

Figure 8 (a) illustrates the convergence of the training loss for the Bayesian LSTM model, which decreases consistently from an initial value of 0.538 to a final value of approximately 0.02. This confirms the model was continuously minimizing its error on the training data. In contrast, the validation loss, after initially decreasing from 0.271 to a minimum of approximately 0.066, began to consistently rise, eventually finishing at 0.09. This divergence, where the validation loss rises while the training loss falls, is a definitive indicator that the model has started to overfit.

This evidence justifies our proposal to implement an early stopping mechanism, which halts the model's training when the validation loss fails to improve for a set number of epochs (e.g., 15–20). This practice ensures the model is saved at its peak generalization performance.

The accuracy plot in Figure 8 (b) shows that the validation accuracy improves over the epochs reaching its maximum value 0.97 within 150 epochs.

Figure 8 (c), which displays prediction uncertainty, shows that while the model is confident in the vast majority of its predictions (260 predictions with σ less than 0.02), it successfully identifies a smaller subset of cases where it is highly uncertain (140 predictions with σ more than 0.02).

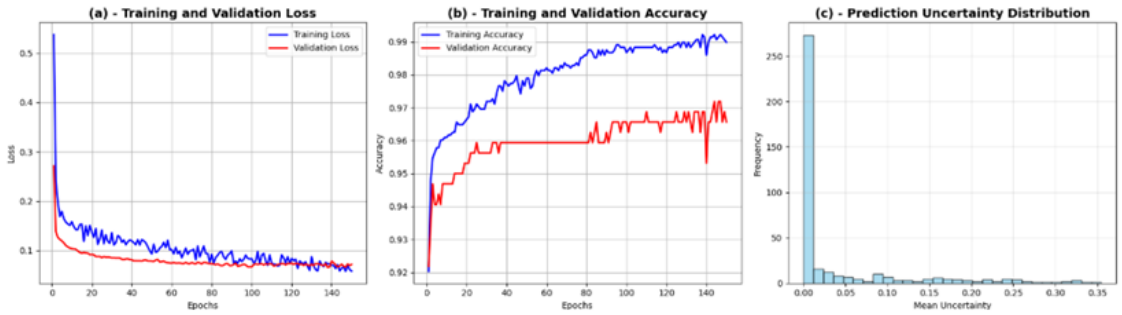

Figure 8: (a) - convergence of the training and validation loss for the Bayesian LSTM model; (b) - accuracy plot for the Bayesian LSTM model; (c) - Bayesian LSTM model prediction uncertainty

Figure 9 shows the relationship between the confidence level and the number of safety alerts generated. As the required confidence level for safe operation increases from 80% to 95%, the system becomes more sensitive, triggering a steady increase in alerts from 60 to 101, respectively. This suggests that for each incremental increase in required confidence, a consistent number of additional edge cases are flagged.

A spike in alerts occurs when moving from 95% to 97.5% confidence threshold, with the number of alerts jumping from 101 to 160—an increase of nearly 60%. This demonstrates that a higher confidence requirement makes the system more sensitive to potential operational risks, with the 97.5% threshold being particularly effective at identifying a large volume of uncertain conditions.

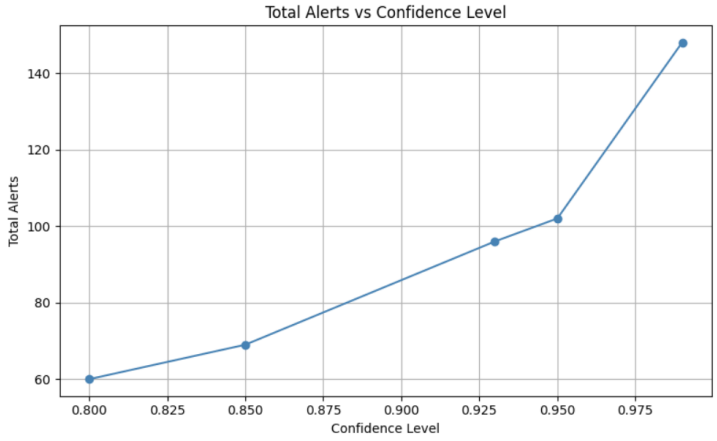

Figure 9: Relationship between the confidence level and the number of safety alerts generated

7. Conclusion

This paper proposes a Bayesian Long Short-Term Memory (LSTM) model to forecast crane stability and quantify its own uncertainty. Together, those components llow a crane's safety monitoring system to trigger an alert if the probability of failure exceeds a critical threshold.

Our results demonstrate that the Bayesian LSTM model can effectively learn the complex, non-linear dynamics of crane operation. The model's ability to quantify its own prediction

uncertainty was clearly validated with the accuracy reached 0.97 and training and validation losses are 0.02 and 0.07 correspondingly. The showed high confidence (uncertainty is less than 0.01) in predictable scenarios while successfully identifying and flagging a critical subset of operations with high uncertainty, which correspond to ambiguous or potentially dangerous situations.

The proposed z-score alerting mechanism proved to be a practical and effective method for translating model uncertainty into actionable safety alerts. The analysis showed a clear, non-linear relationship between the required confidence level and system sensitivity. A confidence threshold of 97.5% was found to be particularly effective, triggering a sharp increase in alerts that captured a large volume of uncertain conditions missed by lower thresholds. This confirms that the crane safety monitoring system can be tuned to a desired level of caution, enabling a formal, risk-based approach to safety management.

Future work focus is to explore an alternative probabilistic architectures, such as Bayesian GRUs, which may offer different performance trade-offs. Investigating adaptive thresholding for the z-score rule, where the alert sensitivity could be dynamically adjusted based on environmental factors like wind speed or ground conditions.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] F. H Guo, Y Zhou, Z Pan, Z Zhang, Y Yu, Y Li, “Automated selection and localization of mobile cranes in construction planning”. Buildings, vol.12 no. 5, (2022). doi: 10.3390/buildings12050580.

[2] B.J. Yoon, K.S. Lee, J.H. Lee, "Study on overturn proof monitoring system of mobile crane". Applied Sciences, vol. 11, no. 15, 6819, (2021). doi: 10.3390/app11156819.

[3] Ie, Gorbatyuk, O. Bulavka, & V. Voliyanuk, V, “Analysis of studies of stationary tower cranes under wind loads”. Girnichi, budivelni, dorozhni ta meliorativni mashini [Mining, construction, road and reclamation machines]. vol. 102, pp. 17-23, (2023). doi: 10.32347/gbdmm, 201.

[4] A. Abubakar, A. I. Halilu, & E. Y. Ejam, “A review of deep learning models for time series forecasting”. Journal of Computer Science and Its Application, vol. 28, no.2, pp. 1-15, (2021). doi: 10.1109/ACCESS.2024.3422528.

[5] W. Li, & K. E. Law, “Deep learning models for time series forecasting: A review”. IEEE Access, vol. 12, 92306-92327, (2024). doi: 10.1109/ACCESS.2024.3422528.

[6] Cui, X., Chipusu, K., Ashraf, M. A., Riaz, M., Xiahou, J., & Huang, J, “Symmetry-Enhanced LSTM-Based Recurrent Neural Network for Oscillation Minimization of Overhead Crane Systems during Material Transportation”. Symmetry, vol. 16, no. 7, 920, (2024). doi: 10.3390/sym16070920.

[7] A. Awasthi, A., Krpalkova, L., & Walsh, J. “Deep Learning-Based Boolean, Time Series, Error Detection, and Predictive Analysis in Container Crane Operations”, Algorithms, vol. 17, no.8, 333, (2024). doi:10.3390/a17080333.

[8] M. Safaei, M. Hejazian, S. Pedrammehr, S. Pakzad, M. Ettefagh, M. Fotouhi, “Damage Detection of Gantry Crane with a Moving Mass Using Artificial Neural Network”. Building, vol. 14, no.2, 458, (2024). doi:10.3390/buildings14020458.
[9] J. Pasukonis, T. Lillicrap, D. Hafner, “Evaluating long-term memory in 3d mazes”. arXiv preprint arXiv:2210.13383, (2022).
[10] P.L. Seabe, C. Moutsinga, “Pindza,Forecasting cryptocurrency prices using LSTM, GRU, and bi-directional LSTM: a deep learning approach”. Fractal and Fractional, vol. 7, no. 2, (2023). doi: 10.3390/fractalfract7020203.
[11] Волянюк, В. О., Горбатюк, Є. В. Розрахунок механізмів вантажопідіймальних машин: навч. посіб. Київ: КНУБА, 2021. 136 с. ISBN 978-966-627-233-4.
[12] S. Lipiec, O. Zvirko, I. Dzioba, & O. Venhryniuk. “Application of the numerical simulation method for the strength analysis of long-term portal crane components”. Advances in Science and Technology. Research Journal, vol. 19, no.4, (2025). doi :10.12913/22998624/200055
[13] Y. Zhou, D. Zhang, H. Huang & Y. Xue. “Effect of normal transformation methods on performance of multivariate normal distribution”. ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering, vol. 8, no.1, 04021074, (2022). doi: 10.1061/AJRUA6.0001198