Cite this article
Huang W, Wei C, Shen K *et al.* (2025)
ITCNet: long sequence time-series forecasting of underwater excavation states for cutter suction dredgers.
Maritime Engineering **178(3)**: 61–79,
https://doi.org/10.1680/jmaen.25.00007

Research Article
Paper 2500007
Received 09/02/2025; Accepted 26/04/2025

Maritime Engineering

ITCNet: long sequence time-series forecasting of underwater excavation states for cutter suction dredgers

Wenjing Huang MSc
College of Mechanical and Electrical Engineering, Hohai University, Changzhou, PR China; Engineering Research Center of Dredging Technology of Ministry of Education, Hohai University, Nanjing, PR China

Changyun Wei PhD
Professor, College of Mechanical and Electrical Engineering, Hohai University, Changzhou, China; Engineering Research Center of Dredging Technology of Ministry of Education, Hohai University, Nanjing, PR China (corresponding author: c.wei@hhu.edu.cn)

Kaijun Shen MSc
College of Mechanical and Electrical Engineering, Hohai University, Changzhou, China; Engineering Research Center of Dredging Technology of Ministry of Education, Hohai University, Nanjing, PR China

Zenghui Liu PhD
Associate Professor, College of Mechanical and Electrical Engineering, Hohai University, Changzhou, China; Engineering Research Center of Dredging Technology of Ministry of Education, Hohai University, Nanjing, PR China

Ze Ji PhD
Professor, School of Engineering, Cardiff University, Cardiff, UK

Dawei Guan PhD
Professor, College of Harbour, Coastal and Offshore Engineering, Hohai University, Nanjing, China; Engineering Research Center of Dredging Technology of Ministry of Education, Hohai University, Nanjing, PR China

In cutter suction dredger (CSD) operations, direct visual assessment of underwater soil composition and terrain is unavailable to operators. This limitation necessitates reliance on indirect indicators, such as concentration meter readings, interpreted through the operator's accumulated experience. However, concentration meters are typically installed at the stern of CSDs rather than near the cutter head, introducing significant time delays in the recorded signals. Consequently, these signals fail to provide real-time feedback on the cutter head's excavation states. To address this challenge, this paper proposes a novel framework based on long sequence time-series forecasting for multi-step prediction of dredged slurry concentration. The framework aims to enable early assessment of underwater excavation states, thereby supporting timely decision making by operators. The study begins by identifying characteristic indicators related to slurry concentration through a detailed analysis of the excavation process. These indicators are then screened to construct a subset of relevant features. Furthermore, a velocity-integrated time compensation method is used to temporally align concentration data with other feature data. The proposed ITCNet model is evaluated against four baseline models using real-world construction data from CSD operations. Results demonstrate that the ITCNet delivers consistent multi-step predictions with minimal errors over a 15 s horizon, providing operators with ample time to respond to dynamic changes. Compared to baseline models, the ITCNet achieves superior accuracy across key metrics, empowering operators to make proactive and informed decisions.

Keywords: cutter suction dredgers/deep learning/long sequence forecasting/state prediction/United Nations sustainable development goals/UN SDG 9: Industry, innovation and infrastructure

Notation

A	cross-sectional area of the sediment
A_c	cutting area
b	cutting width
C_d	drag coefficient
C_i	the ith time segment of concentration series
C_p	correction coefficient used in centrifugal-to-drag ratio calculation
C_v	volume concentration of sediment in slurry
$C_i'(t)$	time-shifted concentration value for segment C_i
$\{C(t)\}$	slurry concentration time series
D_i	effective transport distance for segment C_i
D_p	discharge-pipe diameter
d	cutting thickness per unit time
d_d	actual soil cutting thickness in one swing
d_i	rank difference in Spearman analysis
$[d_{d\min}, d_{d\max}]$	allowable range of cutting thickness
F_1	centrifugal force
F_2	water flow force
F_L	sedimentation correction coefficient
G_{left}, G_{right}	Gini indices of child nodes
G_{parent}	Gini index of parent node

$G(x)$	weight of Gaussian kernel at position x
g	gravitational acceleration
J_c	parameter characterising slurry flow condition
K	convolution filter size
k	correction coefficient for minimum J_c calculation
k_f	kernel radius
L	input sequence length
l	pipeline length between cutter head and concentration sensor
M	mass of a sediment particle
$\mathrm{MI}(X;Y)$	mutual information between X and Y
m	total number of time steps
N	total number of observations
N_{enc}	number of encoder layers
$N_{\mathrm{left}}, N_{\mathrm{right}}$	sample counts in child nodes
N_{total}	total samples in parent node
n	number of sampling points in segment C_i
n_c	cutter rotation speed
$\mathrm{PE}(i)$	positional embedding at i
$p(x)$	marginal probability density of X
$p(y)$	marginal probability density of Y
$p(x,y)$	joint probability density function
Q	total slurry discharge per unit time
R	total number of time segments
R^2	coefficient of determination
R_c	rotation radius of the cutter head
S_m	degree of fragmentation
S_r	correction factor for pipeline geometry
$SE_p(i)$	timestamp embedding from component p
t	time index
t_j, t_{j+1}	adjacent time points for interpolation
t_{lag}	total lag time for slurry transport
u	number of top query vectors
V	volume of sediment excavated by the cutter per unit time
V_c	cutting speed
V_i	volume of sediment entering pipeline per unit time
V_m	volume of sediment forming slurry
v_m	slurry flow rate
v_{me}	slurry flow velocity at rated speed
v_{m0}	critical slurry flow velocity
v_s	swing speed
$\overline{v}_i$	average flow velocity over segment C_i
$v(t_{i,j})$	instantaneous flow velocity at sample j of C_i
X	candidate feature
$X_{\mathrm{feed}_{de}}$	decoder input vector
$X_{\mathrm{feed}_{en}}$	encoder input vector
$X_t^{\mathrm{feed}}[i]$	final input vector at position i in the sequence after projection and embedding
X_t	original time series value at index t
X_{t+x}	original value at position $t+x$ in input sequence
X_{token}	decoder start token
X_0	placeholder token for future sequence
x	distance from kernel centre (time steps)
x_t	input sequence values at t
Y	target variable (slurry concentration)
Y_t	smoothed value after Gaussian filtering
y_t	decoder output at time t
y_i	actual value at time i
$\hat{y}_i$	predicted value at time i
$\overline{y}$	mean of actual values
$\hat{y}_t$	output at time step t in temporal convolutional network decoder
Z_c	number of blade rotations
$\hat{z}^{i-1}$	output from the $(i-1)$th residual block before convolution
$\hat{z}^i$	output from the ith residual block after convolution
α	linear interpolation weight
αu_t^i	projected feature vector at position i
γ	ratio of slurry volume entering pipeline
ΔG_j	Gini index reduction at node j
Δt	sampling interval
η	number of paired observations
Θ	minimum integer satisfying cumulative distance condition
σ	standard deviation of Gaussian kernel
ρ	spearman correlation coefficient
ρ_s	density of sediment particles
ρ_w	density of water
$\sum_{k=1}^{\Theta} D_k$	cumulative transport distance
σ_y	standard deviation of actual values
ω_n	angular velocity of the cutter head
ω_{ne}	cutter's rated speed
$\omega_{n_{\max}}$	cutter's maximum speed

1. Introduction

In coastal construction, cutter suction dredgers (CSDs) are widely used in port construction, mining, and channel regulation due to their high operational efficiency and cost-effectiveness. CSDs operate by employing a cutter head to break underwater rocks, mixing the excavated material with water to form a mud–water slurry. This slurry is then transported to the discharge area through pipelines using hydraulic power. This straightforward dredging process enables the excavation of various soil types while maintaining high productivity (Tang *et al*., 2008; Tang and Wang, 2008). The structural layout of a typical CSD, including key operational components, is illustrated in Figure 1. Currently, CSD operations are predominantly manual. Operators monitor multiple sensor signals in real time and rely on their expertise to adjust the cutter head's depth or cutting speed during swing movements. These adjustments are essential to prevent issues such as insufficient excavation concentration or excessively high slurry density (Helmons *et al*., 2016). The quality of slurry transportation through pipelines is critical, as it directly impacts construction efficiency. Slurry density in pipelines serves as a key indicator of dredging output. If the concentration is too high, part of the mixture will settle at the bottom, forming sand and stone piles that hinder transportation. In severe cases, pipeline blockages may

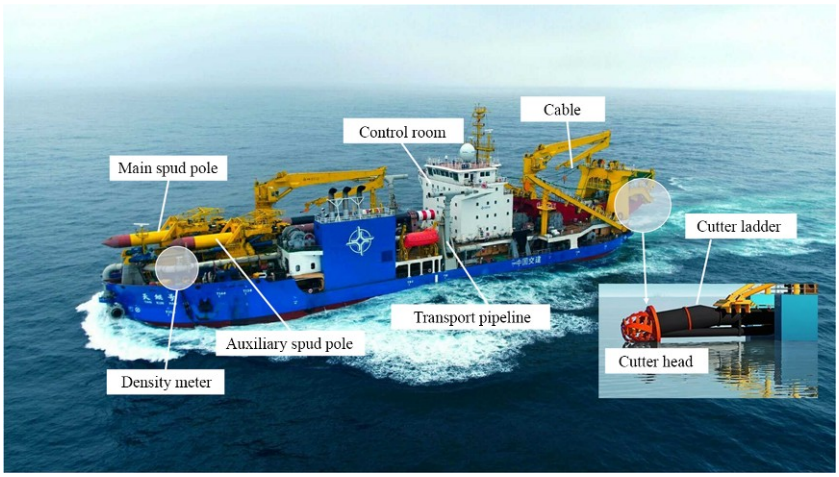

Figure 1. The main components of a cutter suction dredger

occur, leading to bursts and other faults. Conversely, if the slurry concentration is too low, construction efficiency is adversely affected. Therefore, maintaining optimal slurry concentration during excavation is essential.

However, most density meters are installed at the stern of the ship rather than near the cutter head, causing a delay in concentration data relative to other sensor signals (Bai *et al*., 2019). This time lag prevents operators from accurately assessing excavation states in real time, increasing the risk of errors, severe pipeline blockages, and construction delays (Yue *et al*., 2015). Accurate prediction of the excavation states in CSD operations is thus crucial for optimising operations, reducing construction time, and lowering costs (Wei *et al*., 2022).

In addressing time-series prediction problems, research methodologies have evolved significantly, driven by the increasing volume and dimensionality of data. Traditional statistical methods, such as moving average (MA), autoregressive (AR) models, and autoregressive integrated moving average (ARIMA), are valued for their simplicity and strong mathematical foundations (Box and Pierce, 1970; Gu *et al*., 2010; Teräsvirta, 1994). These methods excel in time-series analysis and forecasting by providing clear model behaviour and parameter interpretations while requiring minimal computational resources. However, they are most suitable for stationary time series with linear relationships. They struggle to capture the complex non-linear patterns in real-world dredging data and face challenges with non-stationary data and long-term dependencies, necessitating extensive preprocessing and parameter tuning.

Recent advances have spurred the development of model-based and data-driven time-series forecasting algorithms (Wang *et al*., 2020). Significant progress includes machine learning techniques like support vector machines (SVM), decision trees, and random forests (RF), which enhance prediction accuracy (Biau and Scornet, 2016; De Ville, 2013; Noble, 2006). In addition, deep learning techniques, such as recurrent neural networks (Chung *et al*., 2015; Hewamalage *et al*., 2021), convolutional neural networks (CNNs) (O'Shea and Nash, 2015), and transformer models (Han *et al*., 2021) have further improved forecasting performance by effectively modelling long-term dependencies and complex patterns. Transformers, in particular, have gained attention for their ability to handle time-series data. The attention mechanism inherent in transformers effectively captures long-range dependencies, enabling their application to various forecasting tasks (Liu *et al*., 2019). However, traditional transformer models face challenges with long time-series data, including quadratic time complexity, high memory usage, and slow inference speeds due to step-by-step decoding. Efficient variants such as LogTrans (Li *et al*., 2019), Informer (Zhou *et al*., 2021), and FEDformer (Zhou *et al*., 2022) have been proposed to address these issues. While these advancements have significantly improved forecasting accuracy in many domains, challenges persist in specialised applications and real-world deployments.

In CSD operations, predicting the excavation state has been a key area of research to enhance operational efficiency and ensure safety. Researchers have explored intelligent prediction tools for optimising the excavation process. Key achievements include the application of LightGBM for early blockage warnings using dimensionality reduction techniques (Chen *et al*., 2024) and machine learning models such as RF, SVM, and ANN for predicting slurry density while reducing dependence on expensive sensors (Zarifianshafiei and Amrit, 2023). Several studies have also addressed the time lag between slurry concentration and other sensor signals. Han *et al*. (2022) developed a short-term slurry

concentration prediction model using data mining techniques and ensemble strategies to mitigate lag effects. Wang *et al*. (2023) combined CNN and long short-term memory (LSTM) for cutting torque prediction to guide operational decisions, while Wang *et al*. (2022) proposed a dual-drive approach integrating mechanistic and data-driven models for slurry concentration prediction. Wei *et al*. (2023) introduced the preprocessing-prediction-learning control method, combining CNN models and TD3 reinforcement learning to optimise dredge operation parameters in real time.

Despite these advancements, existing models primarily focus on short-term or single-step predictions, limiting their utility for early and long-term sensing of CSD excavation states. Furthermore, traditional machine learning methods still dominate dredging research, while deep learning approaches remain underutilised. Current long sequence time-series forecasting (LSTF) algorithms often prioritise forecasting trends but neglect the simultaneous capture of both local and long-term dependencies inherent in dredging processes. These limitations, compounded by challenges posed by varying environmental conditions across geographical locations, highlight the need for efficient and accurate methods tailored to CSD operations.

This paper presents ITCNet, an encoder-based framework addressing two critical challenges in CSD operations: delayed slurry concentration feedback and long-term excavation state prediction. The proposed architecture combines a sparse self-attention encoder inspired by Informer (Zhou *et al*., 2021) with a temporal convolutional decoder following Lea *et al*. (2016). Unlike conventional approaches limited to short-term predictions or heuristic delay compensation, ITCNet achieves multi-step concentration forecasting with computational efficiency and physical interpretability. This capability allows real-time prediction of future excavation states using current operational data. The framework provides operators with proactive state anticipation, significantly reducing pipeline blockage risks and preventing potential construction accidents (de Canete *et al*., 2016; Gholami *et al*., 2015; Paulsson *et al*., 2014). The main contributions of this work are summarised as follows:

(*a*) By establishing a dynamic mechanism model grounded in hydraulic transmission theory, the authors identify the fundamental operational parameters that govern slurry concentration dynamics. Their hybrid feature selection methodology combines statistical correlation analysis with importance ranking to systematically eliminate redundant features while preserving physically meaningful parameters, ensuring all selected features maintain strict alignment with actual dredging processes and providing a robust foundation for subsequent modelling.

(*b*) The authors develop a velocity-adaptive compensation approach that accurately models the slurry transport process through pipeline flow dynamics to address sensor delays. This method effectively maintains temporal correlations between different sensor signals by precisely aligning delayed measurements with real-time operational states, overcoming the limitations of conventional fixed-delay compensation techniques.

(*c*) The proposed architecture innovatively integrates sparse self-attention mechanisms with temporal convolution operations to simultaneously capture both long-term concentration trends and short-term dynamic patterns. This dual-capability design enables comprehensive modelling of non-linear interactions between operational parameters while maintaining computational efficiency, providing a complete characterisation of excavation states for reliable monitoring and prediction.

The paper is organised as follows: The dredging operation mechanism of CSDs is analysed in Section 2, where key parameters related to slurry concentration are identified and factors influencing the delay effect are examined. The proposed framework is introduced in Section 3, with its components and data preprocessing steps described in detail. A velocity-integrated time compensation method is employed to eliminate the slurry concentration delay effect. In Section 4, the proposed model is evaluated through multi-step-ahead predictions of slurry concentration and comparative experiments to validate ITCNet's performance. The results demonstrate ITCNet's ability to extract critical features from the preprocessed dataset, align with the dynamic mechanisms of CSD dredging operations, and achieve outstanding predictive performance. Finally, the findings are discussed and conclusions are drawn in Section 5.

2. Excavation mechanism and time-lag effects

The structure of the CSDs is illustrated in Figure 2. At the stern of the vessel, two spud poles are employed for positioning. The main spud pole is mounted on a movable spud carriage, while the auxiliary spud pole is offset from the dredger's centreline. Two anchors, connected to winches by way of cables, are placed on either side of the dredger. The cutter head is attached to a descending cutter ladder, with steel cables on both sides linked to sea anchors.

The dredging process of the CSDs involves four coordinated subprocesses: soil cutting, slurry transportation, transverse movement of the dredger, and incremental forward movement of the trolley (Tang *et al*., 2009). Operators guide the cutter head along an arc by alternately tightening and releasing the side cables, creating a planar cutting motion. During the swinging process, the continuously rotating cutter head cuts hard seabed soil or rock into fragments, and these mix with water to form a slurry. Afterwards, the dredge pump can suck the dredged fragments into the pipeline from an inlet underneath the cutter head and transport the material to a disposal zone (Wei *et al*., 2022). During operation, the dredger pivots around the main spud pole. After excavating a flat seabed section, the main spud pole moves longitudinally along the vessel, pushing the dredger and cutter forward using the main spud pole as a pivot. The auxiliary spud pole provides temporary positioning while the main spud pole is raised and repositioned. Once the main

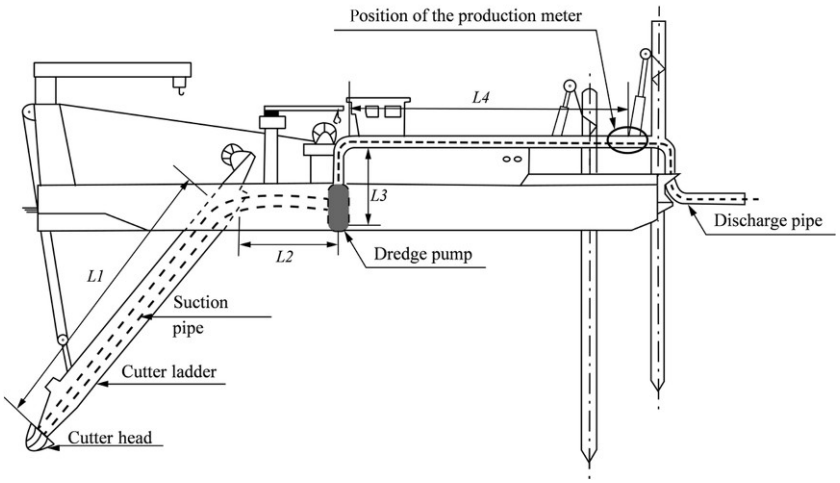

Figure 2. Illustration of the excavating process of CSDs

spud pole returns to its initial position, the dredger continues with the excavation of the next plane.

2.1 Excavation operation mechanism analysis

The suction inlet located at the bottom of the cutter head is positioned near the cutting area and connected to the dredge pump by way of a pipeline. This configuration ensures efficient slurry intake. The dredged material is divided into two parts by the cutter's centrifugal force: one part enters the suction inlet under the combined effects of water flow and pump suction, while the other is flung outward. The volume ratio of these two parts depends on the operational parameters.

The parameters governing soil cutting during the excavation process of CSDs are illustrated in Figure 3. The swing speed is denoted as v_s, the cutting width as b, and the cutting thickness per unit time as d. The volume of sediment excavated by the cutter per unit time, denoted as V, depends on cutting speed V_c and cutting area A_c. These parameters relate through the equation:

1. $$V = A_c V_c = bdv_s$$

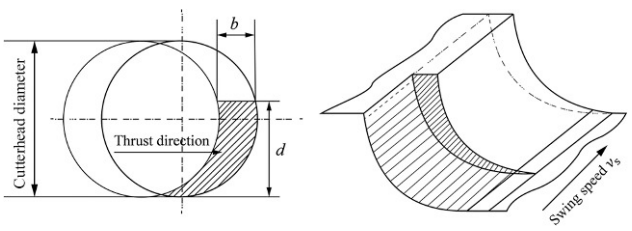

Figure 3. Excavated sediment volume

Finer sediment particles mix with water to form slurry, while larger particles settle on the riverbed. The degree of fragmentation, S_m, is defined as the ratio of the sediment volume forming slurry, V_m, to the total sediment volume, V, expressed as:

2. $$S_m = \frac{V_m}{V}$$

The degree of slurry fragmentation is influenced primarily by two factors: (*a*) the type of soil being excavated and (*b*) the thickness of the soil layer being cut, d_d. The thickness d_d can be calculated as follows:

3. $$d_d = \frac{60v_s}{Z_c \times n_c}$$

where Z_c represents the number of blade rotations, and n_c is the cutter rotation speed.

The dredging efficiency varies depending on the type of dredged soil, which determines the allowable range of cutting thickness, $[d_{d\min}, d_{d\max}]$. If $d_d < d_{d\min}$, all sediment dissolves into the water, while if $d_d < d_{d\max}$, the cutting process becomes ineffective. The degree of fragmentation S_m can be redefined as:

4. $$S_m = \frac{d_{d\max} - d_d}{d_{d\max} - d_{d\min}}$$

Sediment particles entering the pipeline are subjected to centrifugal force F_1 and water flow force F_2. The sediment's motion is

determined by the relative magnitudes of these forces, as depicted in Figure 4. For small sediment particles, where gravity and particle size effects can be ignored, these forces are calculated as follows:

5. $$F_1 = M\omega_n^2 R_c$$

6. $$F_2 = 0.5\rho_w v_m^2 C_d A$$

where M is the mass of a sediment particle, ω_n is the angular velocity of the cutter, R_c is the cutter's rotation radius, ρ_w is the water density, C_d is the drag coefficient, v_m is the slurry flow rate, and A is the cross-sectional area of the sediment. The centrifugal force and water flow force vary under different working conditions, leading to distinct slurry motion states. These states are classified using the dimensionless centrifugal-to-drag ratio (CDR) into three types: constant-diameter helical motion, diffusive spiral motion, and contractive spiral motion. CDR is defined as:

7. $$\mathrm{CDR} = \frac{F_1}{F_2} = C_p\left(\frac{\omega_n R_c}{v_m}\right)^2$$

where C_p represents a correction coefficient.

The parameter $J_c = \frac{\omega_n \cdot R_c}{v_m}$ characterises the slurry flow condition. When $J_c < J_{cmin}$, all slurry enters the pipeline. For a given type of cutter, ignoring sediment particle gravity and size, the value of J_{cmin} is constant for a specific soil type. The expressions for J_{cmin} and J_{cmax} are given by:

8. $$J_{cmin} = k\frac{\omega_{ne} R_c}{v_{me}}$$

9. $$J_{cmax} = \frac{\omega_{nmax} R_c}{v_{m0}}$$

where k is a correction coefficient, ω_{ne} is the cutter's rated speed, v_{me} is the slurry flow velocity at rated speed, ω_{nmax} is the cutter's maximum speed, and v_{m0} is the critical flow velocity. The critical flow velocity v_{m0} is calculated using the Durand model as:

10. $$v_{m0} = F_L\left[2gD_p\left(\frac{\rho_s - \rho_w}{\rho_w}\right)\right]^{1/2}$$

where F_L is the sedimentation correction coefficient, $\boldsymbol{g}$ is gravitational acceleration, ρ_s is sediment density, and D_p is the discharge-pipe diameter.

The ratio of slurry volume entering the pipeline to the total slurry in the cutter, denoted as γ, is expressed as:

11. $$\gamma = \frac{0.7}{J_{cmin} - J_{cmax}} J_c + 1 - \frac{0.7 J_{cmin}}{J_{cmin} - J_{cmax}}$$

The sediment volume entering the pipeline per unit time, V_i, is given by:

12. $$V_i = \gamma V_m$$

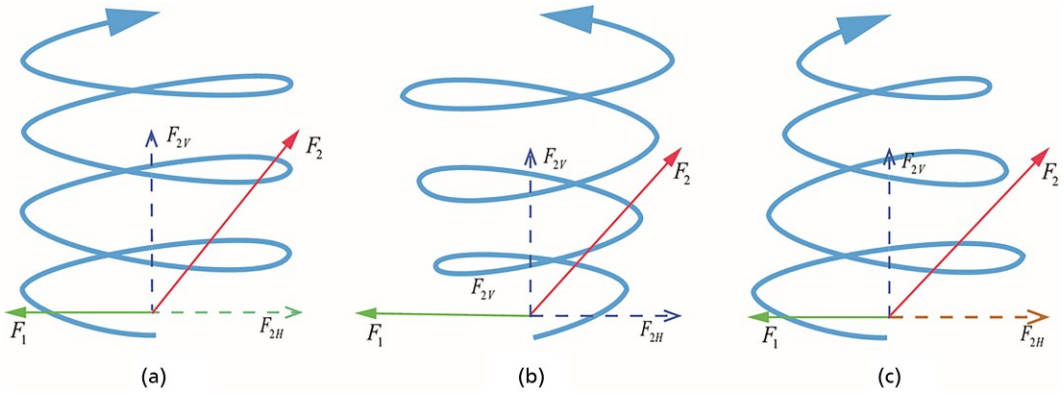

Figure 4. Sediment stress analysis: (a) equal-diameter spiral motion emerges from balanced centrifugal and water flow forces, maintaining constant spiral diameter; (b) diffusive spiral motion develops under dominant centrifugal force, driving outward particle diffusion and diameter expansion; (c) contractive spiral motion forms when water flow force prevails, causing inward particle contraction and diameter reduction. Arrows indicate flow direction

In conclusion, the sediment content in the slurry is described using volume concentration C_v, defined as the ratio of sediment volume V_i to mud flow rate Q per unit time. The slurry concentration expression is:

$$C_v = \frac{V_i}{Q} = \frac{4V_i}{\pi D_p^2 v_m} = 4\gamma S_m \frac{V}{\pi D_p^2 v_m}$$

$$= 4\left(\frac{0.7\omega_n R_c}{(J_{cmin} - J_{cmax})v_m} + 1 - \frac{0.7J_{cmin}}{J_{cmin} - J_{cmax}}\right) \left(\frac{d_{dmax} - d_d}{d_{dmax} - d_{dmin}}\right) \frac{bdv_s}{\pi D_p^2 v_m} \quad (13)$$

From the above analysis, it can be concluded that the slurry concentration in the dredger pipeline is influenced by various factors, including cutter rotation speed, swing speed, slurry flow rate, cutting thickness, suction vacuum, and discharge-pipe diameter. The cutter motor's current and torque directly reflect the cutting force, while the suction vacuum is a critical indicator of slurry concentration. A higher vacuum generally correlates with a higher slurry concentration, and vice versa.

2.2 Time lag analysis

Slurry concentration serves as a critical indicator of underwater excavation conditions, reflecting the operational states of the CSDs. However, during dredging operations, the transportation of slurry through the pipeline introduces a time lag, as it takes time for the slurry to travel from the cutting point to the concentration meter. As illustrated in Figure 5, the variations in slurry concentration lag behind changes in suction vacuum, confirming the time lag effect.

To address this issue, predictive models can be developed using data analysis and modelling techniques. These models allow for the estimation of slurry concentration at the detection point in advance, mitigating the impact of the time lag effect and enabling real-time monitoring and control during dredging operations.

3. LSTF for underwater excavation states

3.1 Overall architecture

This section outlines the design of a LSTF framework aimed at predicting slurry concentration during dredging operations. The approach begins with an in-depth analysis of the excavation mechanism of the CSD to preselect concentration-related feature parameters, followed by thorough data preprocessing. To reduce dimensionality and reveal intrinsic relationships among features, Spearman correlation analysis is combined with mutual information (MI) methods. The RF algorithm is then employed to quantify feature importance, highlighting the correlation between various features and slurry concentration. This step refines the feature subset, ensuring it includes only the most relevant predictors for the task. The refined feature set is subsequently fed into ITCNet, which employs a sparse self-attention mechanism to capture long-term temporal dependencies. In addition, the integration of a temporal convolutional network (TCN) enhances the precision of slurry concentration predictions during underwater excavation by dredgers. The overall architecture of the proposed framework is depicted in Figure 6.

3.2 Data preprocessing

3.2.1 Data cleaning

During the operation of CSDs, data anomalies and missing values frequently occur due to sensor malfunctions, mechanical vibrations, or non-standard crew operations. These issues result

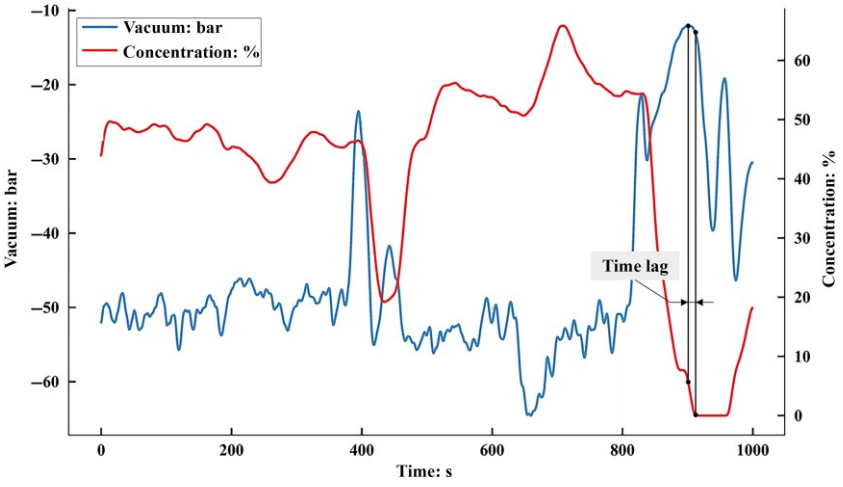

Figure 5. Lagging effect of concentration relative to vacuum

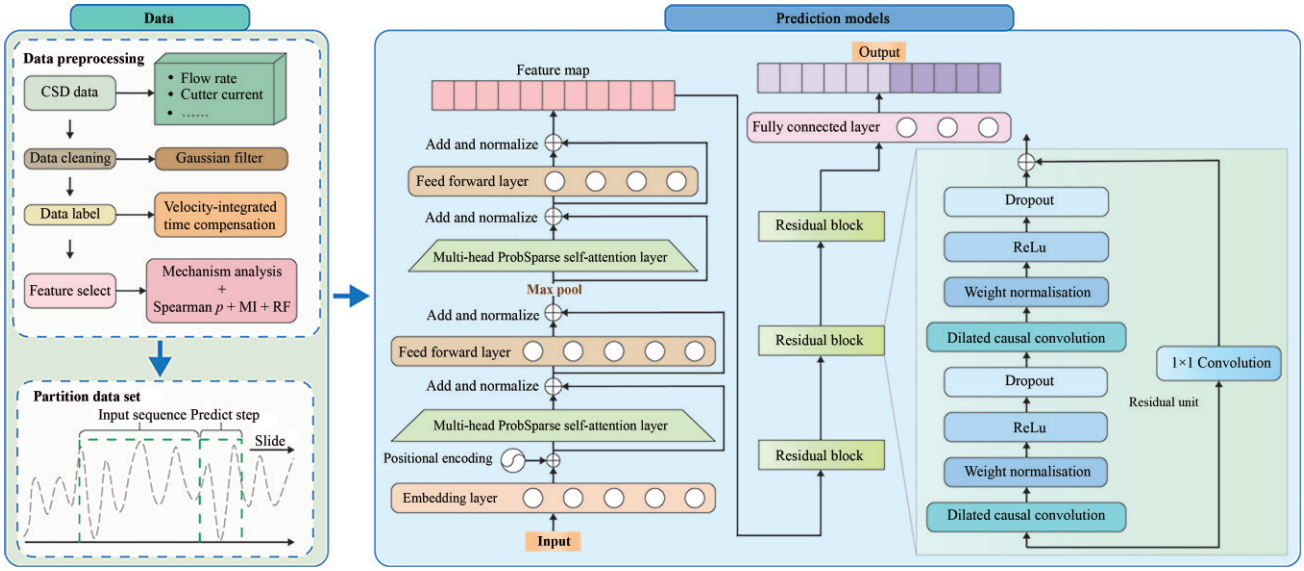

Figure 6. The overall structure of the authors' LSTF framework

in incomplete and disorganised datasets, which, if left unaddressed, can significantly undermine the accuracy of predictive models. In this study, the maximum likelihood estimation (MLE) method is employed to handle missing data. MLE effectively estimates the distribution and parameters of the missing data by leveraging the statistical properties of the available data (Enders, 2022). Unlike simpler methods, such as interpolation or mean imputation, MLE takes into account the probability distribution of the observed data points to infer missing values, providing more accurate estimates. This approach is particularly effective for large datasets, especially when the missing data exhibit certain patterns. Given the sequential nature of long time-series data, MLE is well suited to addressing temporal correlations, ensuring that the imputed data retain the underlying characteristics of the original dataset. Consequently, this method lays a reliable foundation for subsequent slurry concentration predictions.

In addition, sensor data noise, often caused by harsh operational environments and sensor instability, can obscure the true state of underwater excavation. An essential step in preprocessing is selecting appropriate filtering techniques to reduce noise while preserving the essential trends in the data. In this study, the Gaussian filter algorithm is applied for noise suppression and data smoothing (Geusebroek *et al*., 2003). The Gaussian filter kernel is defined based on the Gaussian distribution function, with its one-dimensional expression given by:

$$14. \quad G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

where x represents the distance from the core, and σ is the standard deviation, which controls the width of the kernel. By adjusting the size of the standard deviation σ, it is possible to control the degree of smoothing of the Gaussian filter. For a time series X_t with N observations $0 \leq n \leq N$, Gaussian filtering involves convolving each point with the Gaussian kernel. The convolution process is described as follows:

$$15. \quad Y_t = \sum_{x=-k_f}^{k_f} X_{t+x} G(x)$$

where Y_t is the filtered value at time t, X_{t+x} is the original value at position $t+x$ in the input sequence, t indicates the current time index ($0 \leq t \leq N$), $G(x)$ is the weight within the Gaussian kernel, and k_f determines the kernel radius.

The Gaussian filter is preferred in this study due to its effectiveness in reducing random noise while retaining key trends and features. Its spatially varying weights allow the preservation of edge information and identify rapid changes in the signal. Thus, it is applied to selected feature parameters, ensuring high-quality inputs for predictive modelling.

3.2.2 Data labelling

During dredging operations, slurry concentration signals lag behind other feature signals, with the lag time varying non-linearly with flow velocity. According to the principles of hydraulic transport, the relationship between flow velocity and concentration can be understood through the movement of fluid in the pipeline. Therefore, in this study, a velocity-integrated time compensation method is used to

estimate the lag time between the cutter head and the concentration meter position, making it possible to preprocess the concentration data and eliminate the lag effect.

Specifically, consider a slurry concentration time series $\{C(t)\}$ values along a pipeline of length l (from cutter head to concentration sensor) with a sampling interval Δt. To compensate for the lag, the data are divided into continuous time windows or 'segments' $C_1, C_2, \ldots, C_R$. Each segment C_i contains n sampling points. For segment C_i, the average flow velocity $\overline{v}_i$ is calculated by integrating the instantaneous flow velocity $v(t_{i,j})$ as follows:

$$\overline{v}_i = \frac{1}{n}\sum_{j=1}^{n} v(t_{i,j}) \tag{16}$$

where $v(t_{i,j})$ represents the instantaneous flow velocity at the jth sampling point of C_i. A dimensionless flow correction factor S_r is introduced to model the effective transport distance D_i for each segment:

$$D_i = S_r \overline{v}_i (n\Delta t) \tag{17}$$

where S_r accounts for pipeline geometry and slurry rheology and can be calibrated experimentally by comparing theoretical and actual transport times. The total lag time t_{lag} is determined when the cumulative distance $\sum_{k=1}^{\Theta} D_k$ first exceeds l. Linear interpolation refines the temporal resolution:

$$t_{\text{lag}} = \left(\Theta - 1 + \frac{l - \sum_{k=1}^{\Theta-1} D_k}{D_\Theta}\right) n\Delta t \tag{18}$$

where Θ is the smallest integer satisfying $\sum_{k=1}^{\Theta} D_k \geq l$. This eliminates discretisation errors caused by segmented averaging. Concentration signals are aligned by way of forward time shifting with linear interpolation:

$$C_i'(t) = \alpha C(t_j) + (1-\alpha) C(t_{j+1}) \tag{19}$$

$$\alpha = \frac{t + t_{\text{lag}}}{\Delta t} - \left\lfloor \frac{t + t_{\text{lag}}}{\Delta t} \right\rfloor \tag{20}$$

Segments failing to accumulate sufficient distance $\left(\sum D_k < l\right)$ are flagged as invalid and excluded. Based on the aforementioned method, it is possible to determine the actual time corresponding to the data point in the original dataset, and subsequently shift the concentration signal forward to align with other features.

This preprocessing step enhances data stability and reliability by compensating for non-linear lags, providing accurate and consistent inputs for the ITCNet model. The resulting concentration data, $\{C_1', C_2', \ldots, C_R'\}$, better reflects real-time conditions, improving operational efficiency and safety.

3.3 Feature selection

The original dataset includes numerous features, some of which are irrelevant to the target variable or redundant due to overlapping information. These features not only fail to improve prediction accuracy but can also degrade model performance. Furthermore, an excessive number of features complicates result interpretability, increases computational costs, and prolongs processing time (Theng and Bhoyar, 2024). Feature selection aims to identify a subset of relevant features based on specific criteria, making it a critical step in data preprocessing and dimensionality reduction. In this study, the authors used a systematic feature selection process that begins with correlation analysis.

From the mechanistic analysis in Section 2, an approximate mathematical model for slurry concentration was derived, identifying features directly related to concentration. However, this analysis did not address potential redundancy among the features. To mitigate this, the Spearman correlation analysis was first applied to assess inter-feature correlations (Schober *et al*., 2018). Spearman correlation analysis is a non-parametric statistical method used to evaluate the rank-based association between two variables. It is particularly effective for data with non-linear relationships or outliers, as it does not assume a specific data distribution. Given that the data in this study exhibit non-linear relationships and do not follow a normal distribution, Spearman correlation analysis is especially suitable for identifying strong associations or independence among variables. The formula for calculating the Spearman correlation coefficient is:

$$\rho = 1 - \frac{6\sum d_i^2}{\eta(\eta^2 - 1)} \tag{21}$$

where ρ is the correlation coefficient, d_i is the difference in rank values for the ith data pair, and η is the total number of observations. Features with a correlation coefficient greater than 0.8 are considered highly correlated. For such feature pairs, only one feature is retained to reduce redundancy.

To further evaluate feature relevance, MI is used to quantify the dependence between features and the target variable (Belghazi

et al., 2018). For continuous random variables X and Y, the MI is defined as:

$$\text{22.} \quad \mathrm{MI}(X;Y) = \int_Y \int_X p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) \mathrm{d}x\mathrm{d}y$$

where $p(x,y)$ is the joint probability density function of X and Y, and $p(x)$ and $p(y)$ are their respective marginal probability density functions. Here, X represents a feature, and Y represents the target variable. The MI score indicates the strength of dependence between a feature and the target variable. Among highly correlated feature pairs, the feature with the higher MI score is retained, as it has a stronger relationship with the target variable.

Finally, the RF algorithm is utilised to calculate feature importance based on Gini index reduction (Disha and Waheed, 2022). The importance score of each feature reflects its average contribution across all decision trees. This step ensures the retained features have the most significant impact on model performance. For a dataset split based on feature X, the reduction in the Gini index for a split at node j is:

$$\text{23.} \quad \Delta G_j = G_{\text{parent}} - \left(\frac{N_{\text{left}}}{N_{\text{total}}} G_{\text{left}} + \frac{N_{\text{right}}}{N_{\text{total}}} G_{\text{right}}\right)$$

where G_{parent}, G_{left}, and G_{right} represent the Gini index values of the parent and child nodes, N_{left} and N_{right} denote the number of samples in the left and right child nodes, respectively, and N_{total} is the total number of samples in the parent node.

By combining Spearman correlation analysis, MI, and RF feature importance, a refined subset of features has been identified. This multi-layered feature selection strategy enhances the model's predictive performance, improves interpretability, and optimises computational efficiency.

3.4 ITCNet network

This paper aims to accurately predict pipeline slurry concentration using a long-term sequence prediction network. By establishing a robust predictive model, it is possible to reduce reliance on manual intervention, as deep learning algorithms provide more objective and consistent predictive analyses. To achieve this, the authors propose a novel prediction model, ITCNet. This section introduces the principles of the ITCNet architecture. The ITCNet model adopts the encoder structure from Informer, based on Transformer. The encoder converts input data into feature maps, representing its refined understanding of the historical feature data. For the decoder structure, ITCNet integrates a TCN. The TCN establishes the relationship between the feature maps and historical slurry concentrations, ultimately generating future predicted values.

3.4.1 Informer-encoder

The Transformer model, based on the self-attention mechanism, is a deep learning architecture with an encoder–decoder structure. However, it faces limitations when applied to LSTF. Informer is an enhanced version of Transformer, specifically designed for LSTF tasks. It can handle time-series data with long-range dependencies and irregular intervals. Similar to Transformer, Informer also follows the typical encoder–decoder model structure, capturing temporal features in time-series data using ProbSparse Self-attention mechanisms and convolutional layers. The classical Informer network architecture is presented in Figure 7.

In the ITCNet model, the encoder processes historical feature sequences that have undergone preprocessing and filtering. The input data are first passed through an embedding layer. Given the strong temporal dependencies in slurry concentration prediction, it is essential to capture the continuity of the data. To preserve temporal relationships, positional embeddings are added to each input sample. For position i in the input sequence, the final input vector is expressed as:

$$\text{24.} \quad X_t^{\text{feed}}[i] = \alpha u_t^i + \mathrm{PE}(i) + \sum_p SE_{\mathrm{p}}(i)$$

where αu_t^i represents the projection vector of the context at position i in the feature space. This vector is achieved by scaling the

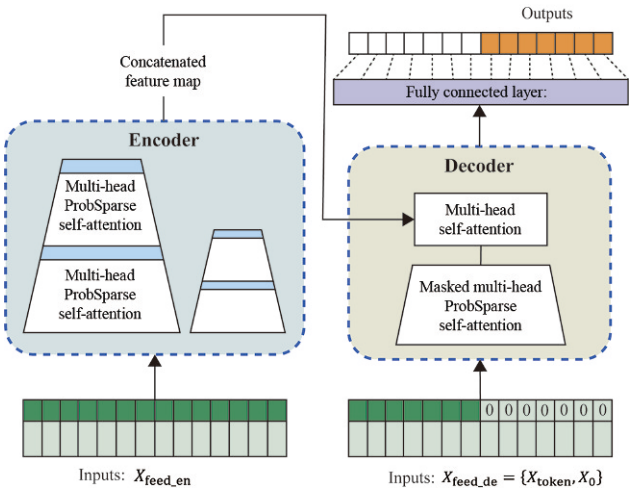

Figure 7. Informer overall architecture. Here, $X_{\text{feed\_en}}$ denotes the encoder input vector. The decoder input $X_{\text{feed\_de}} = \{X_{\text{token}}, X_0\}$ consists of the start token X_{token} and the placeholder X_0 for the target time series, guiding the prediction of future time steps

original feature with a learnable scaling factor α_u. $\mathrm{PE}(i)$ denotes the positional embedding vector obtained by way of sinusoidal encoding and $\sum_p SE_{\mathrm{p}}(i)$ is the sum of global timestamp embedding vectors. This combination of global and local timestamp embeddings enhances the representation of both global positional information and local temporal patterns. The output of the embedding layer is passed through two multi-head ProbSparse self-attention layers and a feedforward layer, generating the feature map for the decoder. As shown in Figure 6, the input data are processed through the ProbSparse self-attention mechanism, residual connections, and normalisation layers within the Attention Block.

Traditional self-attention mechanisms suffer from quadratic computational complexity and memory bottlenecks. The dot-product self-attention shows long-tail sparsity, with only a few dot-products significantly influencing the attention score, while others can be ignored. ProbSparse self-attention addresses this by using Kullback–Leibler (KL) divergence to compute the relative entropy between the query matrix's attention and a uniform distribution. Higher relative entropy identifies more critical query matrices for attention scores. By selecting the top-u query matrices, ProbSparse reduces complexity to $O(L\log(L))$, resolving the quadratic complexity issue. Residual connections link the 'input' and 'output' of layers, mitigating vanishing gradient problems and enabling faster convergence. Layer normalisation on hierarchical network outputs improves generalisation and helps the model find optimal solutions during gradient descent more easily. The feedforward layer applies the ReLU activation function to perform two linear projections, followed by residual connections and layer normalisation. After learning more complex features and patterns from the input sequence, the data are passed to the next encoder.

Repeatedly stacking N_{enc} layers in the Encoder is beneficial for extracting deep feature representations but increases memory usage. To address this, as in Informer, convolutional pooling is introduced between adjacent Attention Blocks. This operation downsamples the features, halving the input sequence length in each layer and reducing spatial complexity. Such downsampling extracts primary features, generating a focused feature map in the subsequent layer while preserving the overall feature distribution.

The encoder structure designed through the aforementioned steps offers notable advantages. It efficiently captures global dependencies in long sequences while retaining critical positional information. Furthermore, it supports multi-modal inputs, making it well-suited for modelling the complex temporal dynamics of slurry concentration. By analysing interrelationships among nodes in the input sequence, the encoder transforms raw input signals into intermediate representations, yielding refined encoded features.

3.4.2 TCN-decoder

The primary function of the decoder is to convert the intermediate representations from the encoder into the target signal. Traditional Transformer-based decoders rely on a step-by-step dynamic decoding process, which requires iterative outputs and consequently reduces efficiency. In contrast, the original Informer decoder utilises the known sequence preceding the target sequence as the starting input while masking the target sequence. Instead of dynamic decoding, it uses a forward process to predict all outputs in a single pass. The decoder takes two key inputs: (*a*) the encoded features from the encoder, which are processed through a multi-head self-attention mechanism, and (*b*) the masked target sequence, represented as positional embeddings with zero-padding for missing values. The decoding process integrates masked sparse attention with self-attention mechanisms to enhance efficiency and effectiveness. Although sparse attention reduces computational complexity, it may also discard crucial information related to local time dependencies, thereby weakening the model's ability to capture fine-grained details such as fluctuation patterns or sudden changes. Moreover, its increased design complexity leads to higher training and debugging costs. To address these challenges, this study adopts the more lightweight TCN as the decoder. TCN offers a simpler architecture while effectively capturing local features and preserving long-term historical information, making it a more efficient alternative.

TCNs rely on dilated causal convolutions to ensure that predictions are made using only past and present information, avoiding information leakage. Specifically, for the output y_t at time step t, the input data include only $x_0, x_1, \ldots, x_t$. This eliminates the need for additional masking mechanisms typically required in sequence models. As the history length increases, the model requires deeper hidden layers to process long-term dependencies. To efficiently handle extended historical data, TCN introduces a dilation factor. This increases the receptive field (i.e. the range of historical information accessible to the model) without sacrificing time resolution. Dilated convolutions insert gaps ('dilations') between convolution kernel elements, enabling coverage of a broader range of inputs. Unlike standard convolutions that process continuous regions, dilated convolutions enlarge the receptive field by introducing gaps between kernel elements, preserving temporal resolution. While traditional CNNs often achieve larger receptive fields by employing pooling layers, this approach can lead to information loss. Dilated convolutions, by contrast, provide expanded coverage without this drawback, making them ideal for time-series data with varying temporal scales. This concept is illustrated in Figure 8.

Handling long sequences often necessitates deeper networks, which can introduce challenges such as vanishing or exploding gradients and a decline in overall network performance. In addition, as the number of layers increases, the network's ability to

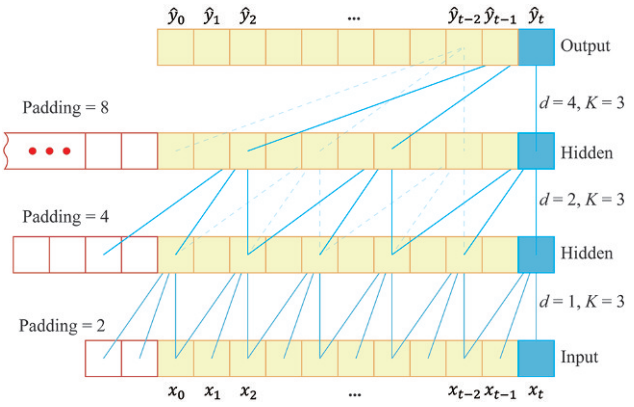

Figure 8. A dilated causal convolution with filter size $K = 3$ and dilations [1, 2, 4]. The input sequence $x_0, x_1, \ldots, x_t$ is processed through the dilated causal convolution to generate the output sequence $\hat{y}_0, \hat{y}_1, \ldots, \hat{y}_t$

converge to an optimal solution weakens. In stochastic gradient descent (SGD)-based training, deeper models are more likely to become trapped in local optima rather than reaching the global optimum. To mitigate these issues, residual blocks are incorporated, inspired by residual blocks that reformulate the learning objective from directly mapping inputs to outputs into learning residual functions. These blocks allow the network to train redundant layers as identity mappings, thereby alleviating degradation problems and improving training stability. The residual module used in the TCN-decoder is illustrated in Figure 9.

This design enhances the generalisation capability of the TCN architecture. To ensure that the input and output sequences maintain the same length, zero-padding is applied during the convolution process. Unlike traditional autoregressive methods, which generate outputs sequentially, TCNs can compute all time steps in parallel, effectively eliminating cumulative error propagation and significantly improving computational efficiency. Moreover, TCN excels at modelling local temporal dependencies, making it particularly effective in capturing fine-grained patterns within sequential data. This capability enables TCN to seamlessly integrate both local details and long-term temporal relationships, resulting in a robust and efficient decoding process.

3.5 Model evaluation metrics

To assess the predictive performance of the model, several evaluation metrics are used, including mean absolute error (MAE), root mean square error (RMSE), normalised root mean square error (nRMSE), R-squared (R^2), weighted absolute percentage error (WAPE), and model runtime. The definitions and formulas for these metrics are as follows:

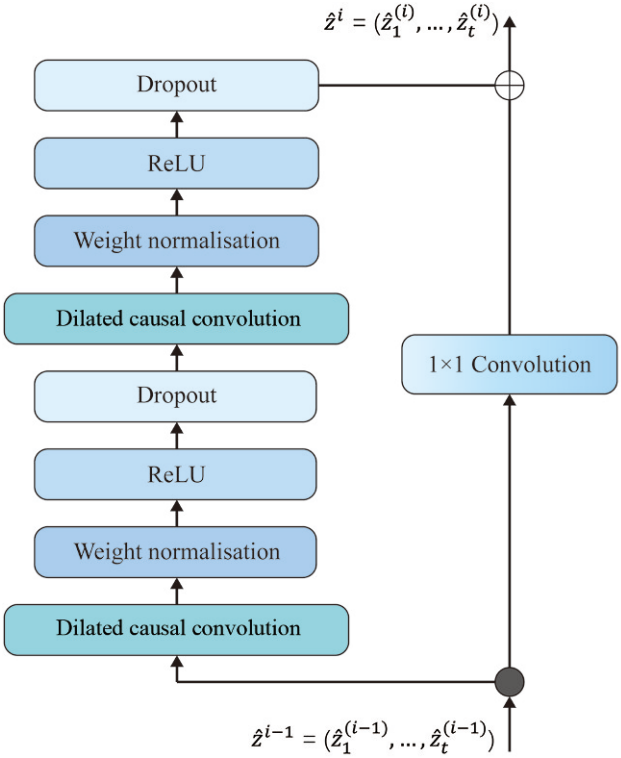

Figure 9. Residual module. A 1 × 1 convolution is added when the input $\hat{\mathbf{z}}^{i-1} = \left(\hat{z}_1^{(i-1)}, \ldots, \hat{z}_t^{(i-1)}\right)$ and output $\hat{\mathbf{z}}^{i} = \left(\hat{z}_1^{(i)}, \ldots, \hat{z}_t^{(i)}\right)$ have different dimensions

25. $$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m}|y_i - y_i'|$$

26. $$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - y_i')^2}$$

27. $$\mathrm{nRMSE} = \frac{\mathrm{RMSE}}{\sigma_y} = \frac{\sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - y_i')^2}}{\sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \bar{y})^2}}$$

28. $$R^2 = 1 - \frac{\sum_{i=1}^{m}(y_i - y_i')^2}{\sum_{i=1}^{m}(y_i - \bar{y})^2}$$

29. $$\mathrm{WAPE} = \frac{\sum_{i=1}^{m}|y_i - y_i'|}{\sum_{i=1}^{m}|y_i|} \times 100\%$$

where y_i is the actual value at the ith time step, y_i' is the predicted value, $\bar{y} = \frac{1}{m}\sum_{i=1}^{m} y_i$ is the mean of all actual values, $\sigma_y = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \bar{y})^2}$ is the standard deviation of observed data, and m is the total number of time steps.

Among these metrics, MAE and RMSE are indicators for longitudinal error evaluation, which mainly measure the overall performance and deviation of the model in long-term operation. While MAE captures the average magnitude of the errors, RMSE emphasises larger deviations due to its quadratic component. To eliminate the influence of slurry concentration data variability, the nRMSE is adopted, which normalises RMSE by the standard deviation of observed data, providing a more objective assessment of the model's ability to capture data fluctuations. A model demonstrates effective performance when nRMSE < 1, indicating its prediction errors are smaller than the natural variability in the data, while nRMSE ≥ 1 suggests that the model's errors exceed the data's intrinsic variability, implying reduced reliability in dynamic representation.

WAPE offers a weighted measure of relative prediction accuracy that accounts for the proportional significance of errors across different value ranges. R^2 evaluates the proportion of variance explained, with values approaching 1 indicating superior explanatory power. Lower values of MAE, RMSE, nRMSE, and WAPE correspond to better predictive performance, while $R^2 \in [0, 1]$ with values closer to 1 denoting stronger model fitting capability.

In addition to these error metrics, the runtime of the model during the evaluation process is considered as a significant metric, particularly in engineering applications. The running time reflects the model's efficiency and directly impacts the real-time data update rate in slurry concentration prediction tasks. Faster runtime ensures that the model can meet the demands of real-time monitoring and prediction, which is crucial for practical applications in engineering.

4. Experiments and results

This section evaluates the performance of ITCNet using real-world data collected from a real CSD. The experimental set-up and data preprocessing steps are first described, followed by the feature selection process using the proposed framework. Finally, ITCNet's performance is analysed for LSTF and compared with alternative algorithms, including LSTM, Transformer, Informer, and TCN, in predicting underwater excavation states.

4.1 Data collection

This study utilises operational records from a CSD involved in a specific dredging project. A 1 week construction dataset was selected as the basis for analysis. From this dataset, a continuous 12 h work segment was extracted as the initial input dataset, representing a typical operational scenario. These records encompass a range of operational parameters and performance metrics, as detailed in Table 1. The selection of this specific time segment ensures a representative sample for the training and testing of the state prediction model, facilitating the development of an accurate and generalisable framework.

Based on the analysis of the CSDs excavation operation mechanism, the feature parameters directly related to the CSDs excavation operation are identified as follows: cutter rotation speed (CRS), swing speed (CSS), slurry flow rate, cutting thickness (CDD), vacuum (IV), cutter motor current (CMC), cutter torque (CT), cutter depth (CD), measured concentration, and discharge pipe diameter. The slurry flow rate includes the pipeline flow rate (PFV), the pump flow rate (FV), and the outlet flow rate (OFV). Since the discharge pipe diameter is constant for the same dredger, it was excluded from the feature subset. The initial feature parameters were selected from the original dataset for further analysis.

4.2 Data filtering and labelling

4.2.1 Data filtering

In practical construction recognition, real-time signals of key indicators must undergo preprocessing to meet the requirements of real-time monitoring and rapid decision making. By focusing on these indicators, the system improves response speed, reduces computational load, and minimises delays caused by inefficient processing. Gaussian filtering applies localised smoothing with dynamically adjusted standard deviation σ, balancing noise suppression and signal fidelity. Broader bandwidths process transient features (e.g. CMC, IV) to accommodate abrupt variations, while narrower bandwidths preserve stable features (e.g. CSS). This adaptive approach prioritises computational efficiency without sacrificing responsiveness. Figure 10 shows filtered data (dashed lines) retaining critical trends while enhancing stability. Grey regions indicate raw signal fluctuations, with filtered trajectories constrained within these ranges. To explicitly visualise outlier suppression, transient anomalies in the CMC profile – identified as deviations exceeding $\pm 2\sigma$ of the baseline distribution – are marked by red circular markers. These annotations highlight the filter's capacity to attenuate high-amplitude disturbances while preserving the underlying operational signatures.

Table 1. Original data of CSD during construction

Time: s	Swing speed: m/s	Production: m^3	Cutter rotation speed: r/min	Cutter depth: m
00:00:01	12.63	1277.78	27.70	13.11
00:00:03	12.62	1278.72	27.72	13.10
00:00:05	13.39	1279.69	27.48	13.04
00:00:07	12.80	1280.66	27.49	13.10
...	...	...	...	13.12
11:59:59	10.73	49 531.82	27.41	13.06

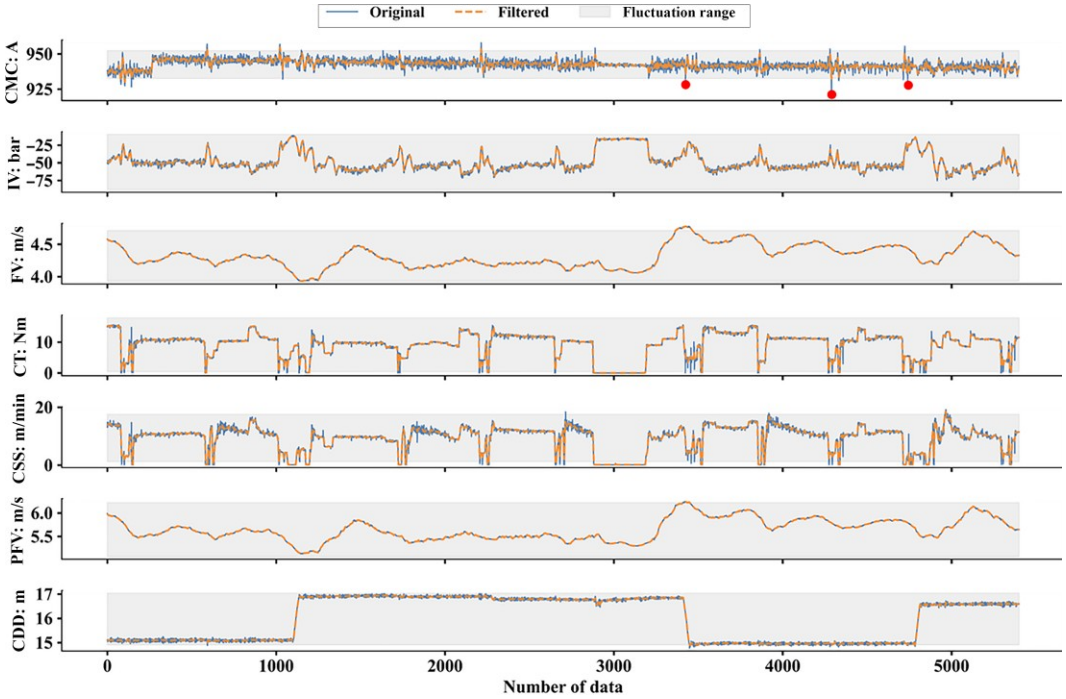

Figure 10. Comparison of data visualisation before and after Gaussian filtering

This preprocessing pipeline ensures that filtered outputs align with the statistical characteristics of raw signals, enabling robust feature extraction for downstream modelling. By adaptively addressing domain-specific noise patterns, the method achieves a pragmatic equilibrium between smoothing intensity and detail preservation.

4.2.2 Data labelling

The density meter of a CSD is typically installed at the stern of the vessel, leading to a delay between the measured density values and the actual density at the suction inlet. To address this lag, the velocity-integrated time compensation method outlined in Subsection 3.2.2 is employed. As depicted in Figure 2, the pipeline length between the cutter head and the production metre ($L_1 + L_2 + L_3 + L_4$) is fixed at approximately 80 m. This fixed configuration reflects the physical constraints of the dredging system described in Subsection 4.1, where the analysed 12 h operational dataset captures a representative scenario under stable pipeline conditions. With a sensor sampling frequency of 30 Hz and an average slurry velocity of approximately 6 m/s, and disregarding the effects of the relative slip ratio, the actual density at each moment and the associated delay time are calculated. The results indicate that the measured density lags the actual suction inlet density by approximately 15 s. This fixed pipeline geometry ensures constant temporal offsets during the short-term monitoring period. Figure 11 shows a comparison between the adjusted lagged density values and the actual density, demonstrating the effectiveness of the time compensation method.

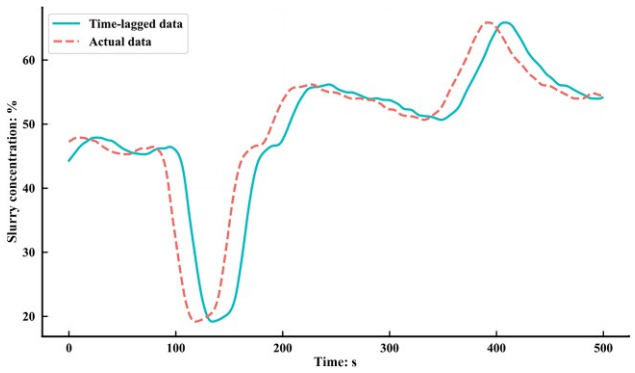

Figure 11. Concentration values that take into account the effects of time lag

4.3 Feature selection

To enhance training efficiency and reduce the computational burden during the deep learning training process (Sze *et al*., 2017), this study combines Spearman correlation analysis, MI, and RF feature importance for feature parameter selection. The results of the Spearman correlation analysis are presented in Figure 12. The highly correlated feature pairs identified include: pump flow

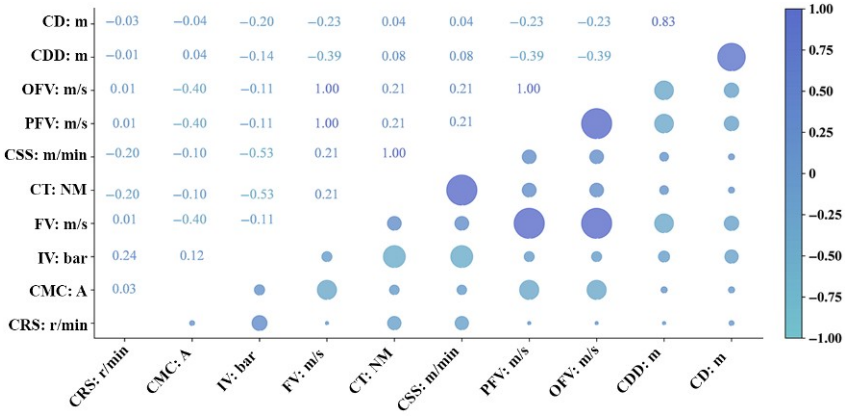

Figure 12. Spearman correlation analysis of the dataset

rate and pipeline flow rate, pump flow rate and outlet flow rate, cutter head cutting thickness and cutter head depth, and pipeline flow rate and outlet flow rate. The MI scores are shown in Figure 13(a). Based on the MI results, features such as pump flow rate, outlet flow rate, and cutter head depth are excluded. The feature importance ranking generated by the RF algorithm is displayed in Figure 13(b). By analysing these importance scores, it is determined that the final set of seven features accounts for over 80% of the total importance score, making them the most representative for the model. The selected features include: cutter motor current, intake vacuum, cutter head torque, swing speed, pipeline flow rate, cutter head cutting thickness, and measured concentration. The filtered feature subset is then used for model training.

4.4 Performance analysis

The preprocessed features form the final dataset, which is split into 70% for training, 10% for validation, and 20% for testing before being fed into the predictive model. ITCNet is used to predict the density at the suction inlet using input features from both current and historical moments. Figure 14 demonstrates strong alignment between predictions (dashed line) and actual measurements (solid line), with the shaded region showing ± RMSE error bounds. Test set evaluation yields a MAE of 0.159, with 68% of prediction errors within the [0.03, 0.29] range, confirming model accuracy. Table 2 summarises the model's performance across various prediction horizons. The results demonstrate a clear pattern where prediction errors increase gradually with longer horizons: MAE rises from 0.3414 (2 s) to 1.2495 (64 s), while RMSE increases from 0.5221 to 1.8455 over the same range. Despite this expected degradation, the model maintains excellent accuracy with WAPE below 3.71% and consistently high R^2 values ($\geq$ 0.9891) across all horizons. The nRMSE remains below 10.42%, further confirming the model's robust performance and strong generalisation capability. Computational efficiency is maintained with calculation times under 1.77 s even for the 64 s horizon,

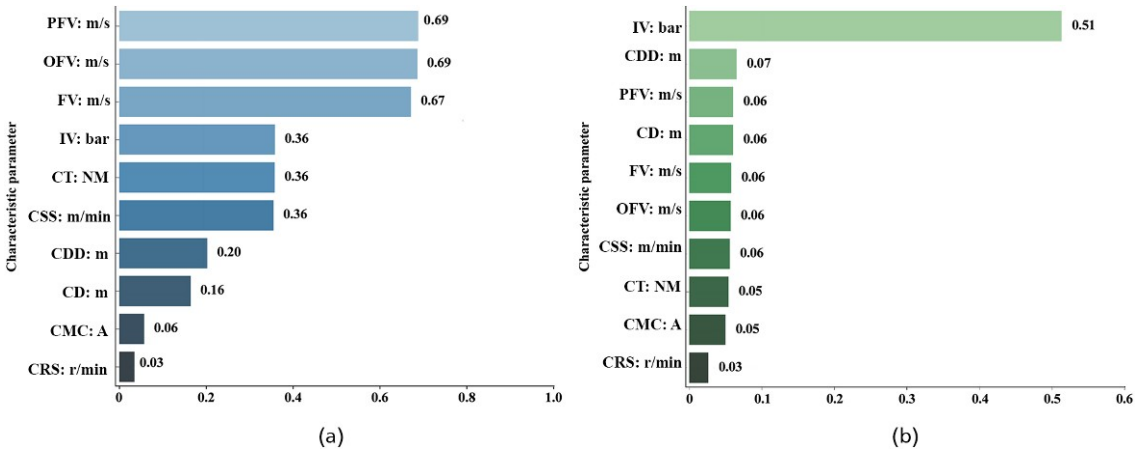

Figure 13. Feature importance analysis results

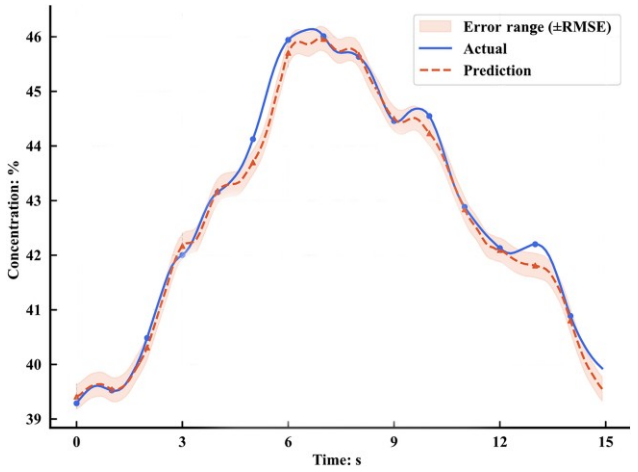

Figure 14. A part of the result of the proposed algorithm

demonstrating the model's practical applicability for real-time forecasting.

4.5 Comparison of model prediction results

In this section, the performance of ITCNet is compared with several widely recognised baseline models, including LSTM, TCN, Transformer, and Informer. These models represent diverse approaches to time series forecasting and are selected to assess ITCNet's relative strengths. To ensure a fair comparison, all models are trained using samples generated from a 20 s sliding retrospective window on the operation time records. Specifically, LSTM, a classic recurrent neural network, is included for its strong ability to capture temporal dependencies and handle both short- and long-term relationships in sequence data. TCN employs convolutional operations to efficiently extract local features from time series while addressing long-term dependencies through an extended receptive field, making it ideal for efficient prediction tasks. Transformer, with its self-attention mechanism, excels at capturing long-range dependencies and complex dynamic relationships, particularly in modelling non-linear temporal features. Informer, an improved version of Transformer, enhances computational efficiency and memory usage, making it specifically suited for long-horizon time series forecasting tasks and offering robust prediction results even with high-dimensional inputs. The comparison of these baseline models provides a comprehensive analysis of ITCNet's advantages in prediction accuracy, stability, and computational efficiency, particularly for slurry concentration forecasting in complex operational scenarios.

As shown in Figure 15, all four baseline models effectively capture the underwater excavation state of the CSD. Among them, the performance of LSTM and Informer is second only to ITCNet. However, LSTM experiences significant accuracy degradation during intervals with strong non-linear features (e.g. from 200 to 300 s), showing notable fluctuations in its predictions. Informer follows the global trend well and aligns with actual values but struggles with local feature instability. In contrast, ITCNet's predictions closely track the actual data across multiple critical intervals, exhibiting minimal fluctuations and demonstrating a strong ability to handle complex non-linear features and long-term dependencies.

To quantitatively evaluate the predictive performance of each model, Table 3 compares model performance across key metrics. ITCNet achieves the lowest prediction errors with MAE = 0.3664 and RMSE = 0.7792. These values represent 35.9% and 29.5% reductions compared to LSTM. The model's nRMSE (0.0441) shows its prediction error accounts for only 4.41% of data variability, confirming superior operational monitoring capabilities.

ITCNet achieves the highest R^2 value among all models, demonstrating superior explanatory power. While LSTM and Informer show competitive temporal modelling capabilities (ranking second and third in R^2, respectively), their performance gaps become evident in other metrics. LSTM shows particular sensitivity to extreme values, while Informer struggles with prediction consistency during critical operational phases. These limitations are effectively addressed by ITCNet's hybrid architecture.

ITCNet demonstrates significant error reduction in nRMSE (28.4% lower than Informer) through its hybrid architecture. This advantage highlights its enhanced capability in handling data variability compared to temporal-focused models.

In computational efficiency, ITCNet maintains optimal balance with 0.81 s runtime – 27.5% faster than TCN while ensuring

Table 2. Performance of different prediction horizons

Prediction horizon: s	MAE	RMSE	nRMSE	R^2	WAPE	Calculation time: s
2	0.3414	0.5221	0.0295	0.9991	1.0115%	0.74
4	0.5160	0.7138	0.0403	0.9984	1.5285%	0.73
8	0.6492	0.8511	0.0481	0.9977	1.9236%	1.01
16	0.7581	0.9148	0.0517	0.9973	2.2460%	1.04
32	0.8703	1.2035	0.0680	0.9954	2.5795%	1.66
64	1.2495	1.8455	0.1042	0.9891	3.7059%	1.77

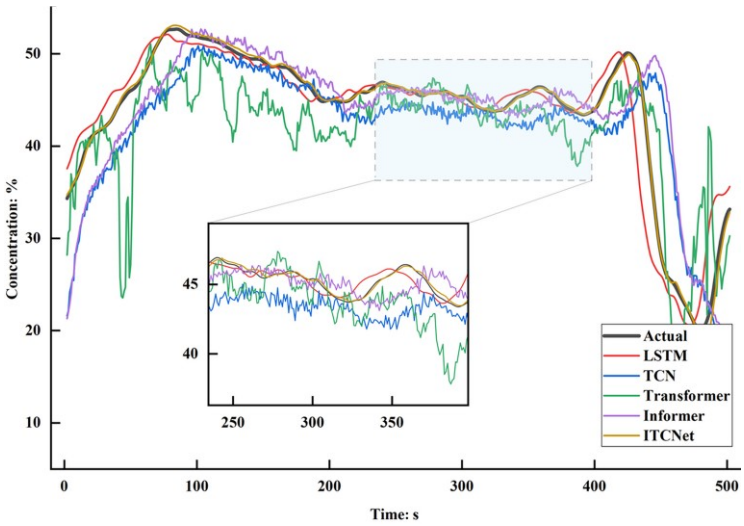

Figure 15. Prediction comparison across models

Table 3. Performance comparison of different models. Bold values indicate the best performance, and underlined values indicate the second-best performance

Model	MAE	RMSE	nRMSE	R^2	WAPE	Calculation time: s
LSTM	0.5711	1.1057	0.0626	0.9961	1.6934%	0.1334
Informer	0.6774	1.0877	0.0616	0.9962	2.0081%	1.06
Transformer	0.7048	1.1721	0.0664	0.9956	2.0894%	**0.0608**
TCN	0.8084	1.3441	0.0761	0.9942	2.3966%	1.1342
ITCNet	**0.3664**	**0.7792**	**0.0441**	**0.9981**	**1.0862%**	0.81

higher accuracy. Notably, Informer requires 30.9% more computation time (1.06 s) than ITCNet despite comparable accuracy, limiting its real-time applicability.

The WAPE metric further validates ITCNet's superiority (1.0862% vs LSTM's 1.6934%). This 35.9% improvement reflects stable performance under complex operating conditions, contrasting with LSTM's fluctuations during concentration changes.

Synthesising all metrics, ITCNet outperforms alternatives with nRMSE = 0.0441 (34.4–42.1% lower than others) and R^2 = 0.9981. The observed limitations in baseline models (LSTM's high RMSE, Informer's elevated WAPE) emphasise the necessity of hybrid architectures for industrial prediction tasks.

5. Conclusions

This article addresses the challenges posed by complex underwater dredging environments during CSD operations, which hinder construction personnel from accurately assessing the underwater excavation states. Given the inherent lag in concentration measurements, the authors propose the ITCNet algorithm for precise prediction of the CSD excavation states.

The main contributions of this study are as follows. First, the dynamic mechanism of the CSD dredging process is analysed to identify features directly related to slurry concentration. Spearman correlation analysis is applied to filter out highly correlated feature pairs and reduce redundancy. Then, MI and RF algorithms are used to compute feature importance, refining the feature subset. To address the lag in slurry concentration measurements, the velocity-integrated time compensation method is introduced to align the feature parameters over time, effectively eliminating delays. To achieve early perception of the CSD's excavation states, a LSTF algorithm is employed for multi-step predictions of underwater excavation conditions. Evaluation metrics such as MAE, RMSE, R^2, and MAPE are introduced to comprehensively assess the predictive accuracy of the proposed model compared to baseline models. In the experimental studies, the proposed ITCNet algorithm demonstrates stable multi-step prediction performance for the underwater excavation state of CSDs, with minimal prediction errors. Comparative experiments with other algorithms show

that ITCNet outperforms the baseline models in key metrics, allowing operators to anticipate underwater excavation states and make proactive, informed decisions.

Considering the safety aspects of CSD operations, accurately predicting and proactively perceiving the underwater excavation states is crucial. The approach presented in this paper not only addresses the lag issue in concentration prediction but also achieves early perception of the excavation states. With its high prediction accuracy, the model assists operators in making timely decisions and adjusting construction parameters based on real-time underwater conditions, contributing to safer and more efficient dredging operations. Future work will explore the integration of prediction results into downstream tasks, such as fault detection, to further enhance the practical applicability of the proposed approach.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under grant no. 52371275, and in part by the National Key R&D Program of China under grant no. 2024YFC3211001.

REFERENCES

Bai S, Li M, Kong R *et al.* (2019) Data mining approach to construction productivity prediction for cutter suction dredgers. *Automation in Construction* **105**: 102833.

Belghazi MI, Baratin A, Rajeshwar S *et al.* (2018) Mutual information neural estimation. In *International conference on machine learning, PMLR*, pp. 531–540.

Biau G and Scornet E (2016) A random forest guided tour. *Test* **25(2)**: 197–227.

Box GE and Pierce DA (1970) Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association* **65(332)**: 1509–1526.

de Canete JF, del Saz-Orozco P, Baratti R *et al.* (2016) Soft-sensing estimation of plant effluent concentrations in a biological wastewater treatment plant using an optimal neural network. *Expert Systems with Applications* **63**: 8–19.

Chen H, Yuan Z, Wang W *et al.* (2024) Data-driven method for vacuum prediction in the underwater pump of a cutter suction dredger. *Processes* **12(4)**: 812.

Chung J, Kastner K, Dinh L *et al.* (2015) A recurrent latent variable model for sequential data. In *Proceedings of the 29th International Conference on Neural Information Processing Systems – Volume 2*, pp. 2980–2988.

De Ville B (2013) Decision trees. *Wiley Interdisciplinary Reviews: Computational Statistics* **5**: 448–455.

Disha RA and Waheed S (2022) Performance analysis of machine learning models for intrusion detection system using Gini impurity-based weighted random forest (Giwrf) feature selection technique. *Cybersecurity* **5(1)**.

Enders CK (2022) *Applied Missing Data Analysis*. Guilford Publications.

Geusebroek JM, Smeulders AW and Van De Weijer J (2003) Fast anisotropic Gauss filtering. *IEEE Transactions on Image Processing: a Publication of the IEEE Signal Processing Society* **12(8)**: 938–943.

Gholami A, Shahbazian M and Safian G (2015) Soft sensor development for distillation columns using fuzzy c-means and the recursive finite Newton algorithm with support vector regression (RFN-SVR). *Industrial & Engineering Chemistry Research* **54(48)**: 12031–12039.

Gu GF and Zhou WX (2010) Detrending moving average algorithm for multifractals. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics* **82(1 Pt 1)**: 011136.

Han K, Xiao A, Wu E *et al.* (2021) Transformer in transformer. *Advances in Neural Information Processing Systems* **34**: 15908–15919.

Han S, Li H, Li M *et al.* (2022) Intelligent short-term forecasting for mud concentration in CSD dredging construction. *Ocean Engineering* **266**: 113151.

Helmons R, Miedema S, Grima MA and van Rhee C (2016) Modeling fluid pressure effects when cutting saturated rock. *Engineering Geology* **211**: 50–60.

Hewamalage H, Bergmeir C and Bandara K (2021) Recurrent neural networks for time series forecasting: current status and future directions. *International Journal of Forecasting* **37(1)**: 388–427.

Lea C, Vidal R, Reiter A and Hager GD (2016) Temporal convolutional networks: A unified approach to action segmentation. In *Computer Vision–ECCV 2016 Workshops, Amsterdam, The Netherlands, October 8–10 and 15–16, 2016, Proceedings, Part III 14*. Springer International Publishing, pp. 47–54.

Li S, Jin X, Xuan Y *et al.* (2019) Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 5243–5253.

Liu J, Lin H, Liu X, *et al.* (2019) Transformer-based capsule network for stock movement prediction. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*, pp. 66–73.

Noble WS (2006) What is a support vector machine? *Nature Biotechnology* **24(12)**: 1565–1567.

O'Shea K and Nash R (2015) An introduction to convolutional neural networks. *arXiv Preprint arXiv:1511.08458*.

Paulsson D, Gustavsson R and Mandenius CF (2014) A soft sensor for bioprocess control based on sequential filtering of metabolic heat signals. *Sensors (Basel, Switzerland)* **14(10)**: 17864–17882.

Schober P, Boer C and Schwarte LA (2018) Correlation coefficients: appropriate use and interpretation. *Anesthesia and Analgesia* **126(5)**: 1763–1768.

Sze V, Chen YH, Yang TJ and Emer JS (2017) Efficient processing of deep neural networks: a tutorial and survey. *Proceedings of the IEEE* **105(12)**: 2295–2329.

Tang J and Wang Q (2008) Online fault diagnosis and prevention expert system for dredgers. *Expert Systems with Applications* **34(1)**: 511–521.

Tang J, Wang Q and Bi Z (2008) Expert system for operation optimization and control of cutter suction dredger. *Expert Systems with Applications* **34(3)**: 2180–2192.

Tang J, Wang Q and Zhong T (2009) Automatic monitoring and control of cutter suction dredger. *Automation in Construction* **18(2)**: 194–203.

Teräsvirta T (1994) Specification, estimation, and evaluation of smooth transition autoregressive models. *Journal of the American Statistical Association* **89**: 208–218.

Theng D and Bhoyar KK (2024) Feature selection techniques for machine learning: a survey of more than two decades of research. *Knowledge and Information Systems* **66(3)**: 1575–1637.

Wang B, Fan S, Jiang P *et al.* (2023) Cutting state estimation and time series prediction using deep learning for cutter suction dredger. *Applied Ocean Research* **134**: 103515.

Wang B, Fan S, Jiang P *et al.* (2020) A novel method with stacking learning of data-driven soft sensors for mud concentration in a cutter suction dredger. *Sensors* **20(21)**: 6075.

Wang B, Fan S, Xiong T and Zhu H (2022) Comparison of cutter suction dredger slurry concentration soft measurement method based on the mechanism and data dual-driven model, In *International Conference*

on Offshore Mechanics and Arctic Engineering. American Society of Mechanical Engineers: p. V05BT06A024.

Wei C, Bai H, Wei Y, Ji Z and Liu Z (2022) Learning manipulation skills with demonstrations for the swing process control of dredgers. *Ocean Engineering* **246**: 110545.

Wei C, Wang H, Bai H, Ji Z and Liu Z (2023) PPLC: data-driven offline learning approach for excavating control of cutter suction dredgers. *Engineering Applications of Artificial Intelligence* **125**: 106708.

Yue P, Zhong D, Miao Z and Yu J (2015) Prediction of dredging productivity using a rock and soil classification model. *Journal of Waterway, Port, Coastal, and Ocean Engineering* **141(4)**: 06015001.

Zarifianshafiei Z and Amrit C (2023) Using machine learning of sensor data to estimate the production of cutter suction dredgers. In *International Conference on Intelligent Human Computer Interaction*. Springer, pp. 244–255.

Zhou H, Zhang S, Peng J *et al.* (2021) Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 11106–11115.

Zhou T, Ma Z, Wen Q *et al.* (2022) Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning, PMLR*, pp. 27268–27286.