Petroleum Science 23 (2026) 4082–4099

Contents lists available at ScienceDirect

Petroleum Science

journal homepage: www.keaipublishing.com/en/journals/petroleum-science

Original Paper

Intelligent early kick recognition via knowledge-guided Bi-LSTM

Dong-Yue Wang [a], Wei-Feng Sun [a,*], Yan-Liang Guo [a], Chen-Tao Gong [a], Wei-Min Huang [b]

[a] College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao, 266580, Shandong, China
[b] Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St.John's, NL A1B 3X5, Canada

ARTICLE INFO

Article history:
Received 15 July 2025
Received in revised form
8 January 2026
Accepted 10 March 2026
Available online 16 March 2026

Edited by Xi Zhang and Jia-Jia Fei

Keywords:
Early kick recognition
Knowledge-guided neural network
Bi-directional long short-term memory
Loss function
Knowledge embedding

ABSTRACT

During oil and gas drilling, the accuracy of intelligent early kick recognition models relies on their detection of abnormal variation trends in monitoring parameters. These variation trends are contained in the data sequences of monitoring parameters corresponding to early kick samples. Therefore, accurate identification of the abnormal variation trends of monitoring parameters in these samples is of great importance for early kick monitoring. This understanding can serve as a prior knowledge to guide the learning process of intelligent models. Since intelligent models use loss values calculated from their loss functions to determine which samples should be focused on during the model training process, the prior knowledge can be used to modify their loss functions. First, a knowledge-guided attention allocation network is proposed, which allocates attention weights to monitoring parameters and enables the intelligent model to pay greater attention to these critical ones. Then, a knowledge-guided bidirectional long short-term memory (KG-Bi-LSTM) model was established by embedding prior knowledge into its binary cross-entropy loss function. Firstly, in order to reduce the model's attention on the most easily identifiable samples corresponding to normal drilling, the square of the model's prediction error for each normal sample is used as an attenuation factor for loss values corresponding to normal samples. In this way, the model can learn more from kick samples. Secondly, to increase the model's focus on learning the variation trends of monitoring parameters corresponding to the early kicks, a monotonically decreasing exponential function is employed to weight the loss term for kick samples. This approach assigns greater weights to samples from the early kicks, thereby preventing underfitting by emphasizing key trend recognition. Subsequently, to prevent the model from overfitting due to the weighting of loss values, the sum of squares of the network weights is added to the loss function as a constraint term. Thus, a balance between model overfitting and underfitting is achieved through knowledge guidance and weight constraint. Thereafter, a knowledge-guided loss function is obtained by combining the above weighted loss terms and the constraint term. Finally, this loss function is applied to the Bi-LSTM-based kick recognition model to produce a KG-Bi-LSTM version. Experiments were conducted using 4,599 samples from field data, the results show that compared with the prevailing Bi-LSTM model, the proposed KG-Bi-LSTM has a 5.28% increase in recognition accuracy and an earlier alarm time of nearly 2 min.

© 2026 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Kick refers to the phenomenon that formation fluid or gas intrudes into the wellbore when the pore pressure exceeds the bottomhole pressure during the drilling process (Obi et al., 2023). If a kick is not controlled in time, it may lead to a serious blowout accident, resulting in property damage and even casualties (Zhu et al., 2023; Zhang et al., 2024b). Therefore, accurate recognition of early kicks is helpful to the timely adoption of well control measures, which is of great significance for safe and efficient drilling (Wang et al., 2023; Zhang et al., 2025).

In recent years, data-driven models have been widely used in the field of kick recognition (Cinelli et al., 2021; Xu et al., 2023). These models can be categorized into supervised learning models, semi-supervised learning models, and unsupervised learning models. Back propagation neural network (BPNN), and Long short-term memory (LSTM) are commonly used supervised learning

* Corresponding author.
E-mail address: sunwf@upc.edu.cn (W.-F. Sun).

Peer review under the responsibility of China University of Petroleum (Beijing).

https://doi.org/10.1016/j.petsci.2026.03.018

1995-8226/© 2026 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

models in kick intelligent monitoring, with BPNN being one of the first neural networks applied to this field. For example, Liang et al. (2019) constructed a BPNN-based kick recognition model using stand pipe pressure (SPP) and casing pressure as monitoring parameters. Compared with single-parameter models, BPNN has a lower false alarm rate. Muojeke et al. (2020) added downhole pressure to the monitoring parameters of BPNN to improve the timeliness of kick alarms. Elmgerbi et al. (2022) proposed different combinations of monitoring parameters for different drilling risks, including kick, and used BPNN to establish corresponding drilling risk recognition models. Considering that the intrusion of formation fluids into the annulus is a continuous process, the recognition model should determine whether a kick has occurred or not based on the variation trends in monitoring parameters over a period of time. However, BPNN processes the data received at each sampling instant individually, ignoring the temporal variation trends of monitoring parameters. This restricts its ability to learn the variation trends in data sequences from kick samples, resulting in low accuracy in early kick recognition.

Kicks were characterized by abnormal variation trends of monitoring parameters over a period of time. LSTM can capture the nonlinear and complex relationship between variations of monitoring parameter data sequences and kick occurrence, thus it has been one of the most dominant models in the field of kick recognition. The application of LSTM to kick recognition began in 2019 and has since become widespread. The recognition accuracy of LSTM networks relies on the adopted monitoring parameters. These monitoring parameters can be divided into three classes, including mud parameters, surface engineering parameters, and downhole engineering parameters. In order to improve the accuracy of LSTM in kick detection, researchers have investigated various monitoring parameter combinations, as listed in Table 1. For example, Fjetland et al. (2019) demonstrated that the LSTM employing both surface engineering and mud parameters achieved superior identification performance compared with models relying solely on either surface engineering or mud parameters. In subsequent studies, as researchers gained a deeper understanding of the parameter variation patterns of kicks, more monitoring parameters were introduced. For instance, since the primary reason for kicks is that the formation pressure exceeds the downhole pressure, the d-exponent, which could reflect changes in formation pressure, was involved as a monitoring parameter of LSTM (Osarogiagbon et al., 2020). Furthermore, after a kick occurs, mud parameters, such as differential flow out (DFO) and sum of pits (SUM), respond slower than surface engineering parameters, such as SPP, weight on bit (WOB), torque, and rate of penetration (ROP). To ensure that the kick recognition model can recognize the variation trends of monitoring parameters with different response speeds, Zhang et al. (2023a) proposed a hierarchical early kick detection model based on a cascaded gated recurrent unit (GRU). Additionally, considering that the intrusion of formation fluids could lead to significant variations in mud parameters, Wang et al. (2024) incorporated mud temperature, density, and conductivity into the LSTM's monitoring parameters. In order to improve the timeliness of kick recognition, bottomhole pressure (BHP) as well as Doppler acoustic wave amplitude were employed. To enable the LSTM to more effectively capture abnormal trends in monitoring parameters, Yin et al. (2021) introduced BHP, equivalent circulation density (ECD), weight on hook (WOH), and filtered flow out into the set of monitoring parameters. Considering that the amplitude of the Doppler acoustic wave increases when they strikes the bubbles in the mud, the Doppler amplitude (AMP) was introduced to enhance the timeliness of early gas kick detection (Yin et al., 2022). In recent years, various improvements of LSTM have been applied in the field of kick monitoring, including GRU and the bidirectional long short-term memory (Bi-LSTM). To determine the best data-driven model for kick detection, Yin et al. (2025) compared the performance of three typical neural networks. The results verified that Bi-LSTM performed better than LSTM and GRU.

Besides LSTM and BPNN, other supervised learning models such as convolutional neural network (CNN) (Chen et al., 2024), random forest (RF) (Fu et al., 2023), and support vector machine (SVM) (Feng et al., 2025) have also been widely applied in the field of kick recognition. However, supervised learning models require large amounts of labeled kick data, which are usually unavailable. Thus, unsupervised models were introduced. Autoencoder (AE) is one of the most widely used unsupervised learning models in this field, as it only learns the variation patterns of monitoring parameters from normal drilling data. Once the variation trends of monitoring parameters deviate from these learned normal patterns, a warning occurs. For instance, Zhou et al. (2025) developed an unsupervised learning-based kick risk warning model integrating bidirectional gated recurrent units (Bi-GRU) and an AE. Prior knowledge encompassing drilling operational details and the variation rates of key monitoring parameters was embedded into the model, which significantly enhanced the accuracy and practicality of kick risk warning under dynamic drilling conditions. Liu et al. (2025) proposed an unsupervised drilling anomaly detection model that fuses domain-specific prior knowledge with the affinity propagation algorithm. This model dynamically selects optimal monitoring parameters based on real-time drilling states and can be continuously updated using historical kick cases, thereby improving its adaptability to complex downhole environments. Additionally, Zhang et al. (2023b) proposed a kick and lost circulation monitoring method using a multivariate time series sparse autoencoder (SAE). It first uses the SAE to detect abnormal trends in monitoring parameters during the drilling process. Then, it integrates reconstruction analysis and time series segmentation techniques to classify these abnormal trends, determining whether the anomaly is a kick or a lost circulation. Wu et al. (2024) proposed an intelligent monitoring model based on an unsupervised time-series autoencoder. The model uses a Bi-

Table 1
Summary of selected monitoring parameters for LSTM-based kick detection methods since 2019.

Year	Authors	Mud parameters	Surface engineering parameters	Downhole engineering parameters
2019	Fjetland et al. (2019)	Flow rate in, flow rate out	SPP, choke pressure, choke opening, bit pressure, bit depth	–
2020	Osarogiagbon et al. (2020)	–	SPP, d-exponent	–
2021	Yin et al. (2021)	DFO, SUM, filtered flow out	ROP, WOB, WOH, SPP, ECD, torque	BHP
2022	Yin et al. (2022)	DFO, SUM, AMP	ROP, WOB, WOH, SPP, torque	BHP
2023	Zhang et al. (2023a)	DFO, SUM	ROP, WOB, SPP, torque	–
2024	Wang et al. (2024)	Pits volume, flow rate difference, density, temperature, conductivity	SPP, drilling time, d-exponent	–
2025	Yin et al. (2025)	DFO, AMP	ROP, WOB, SPP	BHP

LSTM as both the encoder and decoder and recognizes anomalies via the reconstruction error, enabling accurate monitoring without labeled data. Additionally, Li et al. (2021) utilized density-based spatial clustering of applications with noise (DBSCAN), another classical unsupervised learning algorithm, to cluster drilling data, aiming to determine whether the abnormal trends in the data are caused by drilling operations or downhole accidents.

Additionally, semi-supervised learning, which combines the advantages of both supervised and unsupervised learning, has also been applied to the field of kick monitoring. For example, to enable accurate kick monitoring when only a small amount of labeled data are available, Liu et al. (2022) designed a semi-supervised model based on a teacher–student framework that learns from both labeled and unlabeled data. For labeled data, the student model is trained to capture the relationship between temporal trends in monitoring parameters and their labels. For unlabeled data, the teacher model acts as a surrogate supervisor, guiding the student's training in place of ground-truth labels. This approach substantially reduces reliance on scarce labeled data.

Despite excellent results have been achieved by data-driven kick recognition methods, the following challenges still need to be addressed. First, the scarcity of kick samples limits data-driven models from being sufficiently trained. Second, the abnormal variation trends of monitoring parameter data sequences corresponding to early kicks are not apparent, making early kick samples difficult to recognize. For example, at the early stage of a kick, abnormal variation trends of mud parameters such as the outflow percentage (FOP) are not obvious. However, whether FOP exhibits abnormal variation trends or not is a necessary condition for determining kick occurrence. Therefore, FOP is more important and should be the focus when the model learns from early kick samples. However, existing intelligent models do not incorporate the prior knowledge of importance differences among monitoring parameters. This may lead to the neglect of abnormal variation trends in critical monitoring parameters during the monitoring stage.

Therefore, due to the combined effect of the aforementioned multiple factors, early kick samples are difficult to identify and thus should be the key focus of the model's learning. However, existing kick recognition models treat all samples as equally important, resulting in insufficient learning of early kicks. In addition, the accuracy of data-driven models is the proportion of correctly classified samples to the total samples, and early kick samples only account for a small portion of the training samples. As a result, whether early kick samples are correctly recognized or not has a small impact on the model's recognition accuracy. Samples that models fail to recognize might be the most important early kick samples. This explains why models with high accuracy may still fail to detect early kicks in real-world applications.

Considering that the identification difficulty varies among different samples and the importance of different monitoring parameters also differs, intelligent models should pay more attention to the learning of hard-to-classify samples. Furthermore, when learning these samples, they should focus more on the variation trends of FOP. This understanding can be incorporated into data-driven models as prior knowledge, guiding it to focus on learning the abnormal variation trends in early kick samples to improve the timeliness of alarms. Specifically, considering that normal samples are numerous and easy to classify, after fully learning the variation trends of these samples, the model should transfer its attention to more important kick samples. Furthermore, given that early kicks are more difficult to recognize than other ones, intelligent models should focus more on learning the abnormal variation trends in early kick samples. Equally importantly, when learning from training samples, the model should focus more on FOP while integrating the variation trends of other monitoring parameters to make accurate predictions.

Since data-driven models use loss values calculated by a loss function to determine insufficiently fitted samples, the loss function can be modified using specific domain knowledge to guide the learning process of data-driven models, then knowledge-guided intelligent models can be produced (Von Rueden et al., 2021). Domain knowledge refers to prior knowledge such as physical equations (Meng et al., 2022), expert rules (Liu et al., 2023), and engineering experience (Nie et al., 2025). In recent years, researchers in various fields have conducted many studies on embedding domain knowledge into data-driven models and have achieved significant results.

In order to ensure that the output of the data-driven model aligns with physical laws, Raissi et al. (2019) proposed the physics informed neural network (PINN) by embedding physics equations into the loss function. With the aim of improving the accuracy of subsurface flow prediction, Wang et al. (2020) proposed a theory-guided neural network (TGNN) by embedding expert knowledge and physical equations into the loss function. Luo et al. (2021) embedded the reasonable range of power generation into the LSTM loss function to reduce unreasonable prediction results. On the basis of TGNN, Chen et al. (2021) proposed a theory-guided hard constraint projection (HCP) method to achieve better accuracy and robustness in subsurface flow prediction. Additionally, Yin et al. (2023) embeds the vibration frequency during bearing faults into the convolutional kernel of a convolutional neural network, thereby improving the accuracy of bearing fault diagnosis. Fang et al. (2025) converted the wear evolution equations into regularization terms and added them into the loss function to produce physically-constrained results. In this way, the output of the tool wear monitoring model conforms to physical rules. It could be concluded from the above studies that incorporating domain knowledge into neural networks is an effective way to improve the performance of data-driven models.

Inspired by the aforementioned studies, in order to improve the accuracy of early kick recognition, the sample importance serves as prior knowledge to guide the learning process of the intelligent kick recognition model based on a Bi-LSTM network. More specifically, the Bi-LSTM network is used as the baseline recognition model, its loss function is modified and improved using kick recognition knowledge as follows: the binary cross-entropy loss term for normal samples is weighted using the square of the prediction error to transfer the model's attention to kick samples, the loss term for kick samples is designed and weighted using a negative exponential function to enhance the model's focus on early kick samples, a constraint term is introduced using the sum of the squares of the network weights to avoid model overfitting. Additionally, before the Bi-LSTM learns from the samples, a prior knowledge-guided attention allocation network is used to assign attention weights to each monitoring parameter. This network consistently allocates higher attention to FOP to prompt the Bi-LSTM to

Table 2
List of monitoring parameters categorized by type.

Mud parameters	Engineering parameters
Inflow rate (FIR)	Well depth (DEPTH)
Outflow percentage (FOP)	Vertical depth (VDEPTH)
Inflow mud density (MWIN)	Bit depth (BITDEPTH)
Outflow mud density (MWOUT)	Hook height (HKH)
Inflow temperature (TEMPIN)	Weight on hook (WOH)
Outflow temperature (TEMPOUT)	Instantaneous rate of penetration (ROP)
Inflow conductivity (CONDIN)	Rotations per minute (RPM)
Outflow conductivity (CONDOUT)	Weight on bit (WOB)
Mud gain/loss volume (GLV)	Torque (TORQUE)
Casing pressure (CP)	Standpipe pressure (SPP)
Sum of pits (SUM)	Bit size (BITSIZE)
Total hydrocarbons (TG)	Dc exponents (DC)
	Total pump strokes (PUMPTOTAL)

monitoring parameters that reflect the patterns of early kicks. The SHAP method enables quantitative interpretation of the model's decision-making process by decomposing model predictions into the independent contribution scores of individual monitoring parameters.

Specifically, when calculating SHAP values, the intelligent recognition model should be trained first. Next, the performance of the intelligent model is tested on the validation set, and the contribution score of each monitoring parameter to the recognition results is calculated using SHAP method. Let $\mathbf{u}$ denote the data sequence of the sample. When computing the SHAP value of the i-th monitoring parameter, the SHAP formulation can be expressed as

$$\phi_\alpha(\mathbf{u}) = \sum_{S \square N_{\text{pa}} \backslash \{\alpha\}} \frac{|S|!\,(n-|S|-1)!}{n!}\,(v_{\mathbf{u}}(S \cup \{\alpha\}) - v_{\mathbf{u}}(S)) \tag{1}$$

$$v_{\mathbf{u}}(S) = \mathbb{E}[f(\mathbf{U}) |\, \mathbf{U}_S = \mathbf{u}_S] \tag{2}$$

where $\phi_\alpha(\mathbf{u})$ is the SHAP value of the α-th monitoring parameter. $N_{\text{pa}} = \{1, 2, \ldots, n\}$ denotes the index set of all monitoring parameters. S denotes any subset of N_{pa} that does not contain α. $v_{\mathbf{u}}(\square)$ is the contribution function for monitoring parameter subsets of $\mathbf{u}$. It quantifies the contribution of different parameter subsets to the model's prediction results, serving as a basis for calculating the SHAP values of each monitoring parameter. $\mathbb{E}$ denotes the mathematical expectation operator, and $f(\square)$ represents the output of the intelligent recognition model. $\mathbf{U}$ denotes the set of all possible value combinations of the monitoring parameters. $\mathbf{U}_S$ denotes the components of $\mathbf{u}$ indexed by S.

Once the SHAP values of each monitoring parameter for every sample in the validation set are obtained, the SHAP values are aggregated to derive the final contribution of each parameter to the recognition result. The aggregation formula is given by

$$U_\alpha = \frac{1}{m}\sum_{d=1}^{m} |\phi_\alpha(\mathbf{u}_d)| \tag{3}$$

where m is the number of validation samples and U_α is the contribution score of the α-th monitoring parameter. Subsequently, U_α was ranked in descending order, and the results are presented in Fig. 2.

It can be observed that the contribution scores of FOP, SPP, SUM, and ROP are significantly higher than those of the other monitoring parameters. Therefore, these four parameters were selected as the monitoring parameters employed in this paper. It should be noted that FOP refers to the percentage of the outlet flow rate measured by a paddle flowmeter, rather than the differential flow rate between the inlet and outlet (DFO). Since the maximum flow rate of this flowmeter was not available, DFO cannot be calculated. Thus, FOP was adopted as the monitoring parameter.

These results reveal which monitoring parameters the model focuses on most when recognizing early kicks. Specifically, the model primarily prioritizes SUM and FOP, as abnormal increases in these parameters are the most salient features of early kicks. In addition, the abnormal variation trend of ROP serves as an additional indicator of early kicks, resulting in a relatively high corresponding contribution score. In contrast, SPP receives the least attention from the model, likely because its fluctuations can also be caused by drilling operations and other factors. Consequently, SPP may exhibit similar patterns under both normal drilling conditions and kick scenarios.

It is worth noting that although the intelligent model assigns greater importance to changes in SUM, the variation in SUM is not obvious at the very early stage of a kick. In comparison, FOP begins to show a trend earlier than SUM, and this trend becomes more pronounced over time. Therefore, the intelligent model should be guided to place greater emphasis on the trend of FOP. This requirement is exactly what the parameter attention mechanism introduced in the next subsection is intended to address.

2.2. Construction of the knowledge guided attention allocation module for monitoring parameters

At the initial stage of a kick event, the abnormal variation trend of monitoring parameters, especially that of FOP, is not obvious. Therefore, intelligent models should prioritize the variation trend of FOP when monitoring kicks, while combining the variation trends of other parameters to determine whether FOP's abnormal trend is caused by a kick. Based on this idea, a knowledge-guided attention allocation module for monitoring parameters was proposed, enabling intelligent models to adaptively identify which monitoring parameter deserves greater focus. During this process, constraints are imposed on attention allocation to ensure that FOP receives more attention than other monitoring parameters.

The structure of the attention allocation module, consisting of two fully connected neural network (FCNN) layers and one output layer, is illustrated in Fig. 3. FCNN is adopted because the weight allocation model focuses on the relative importance of each monitoring parameter rather than the variation trends of data sequences. A simple FCNN can adequately learn the mapping relationship between monitoring parameters and attention weights, and it requires fewer computational resources.

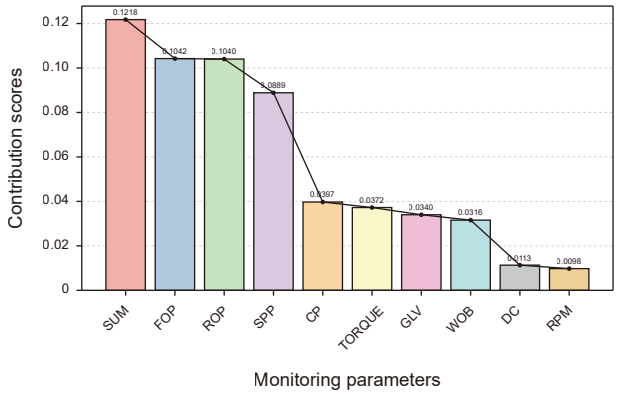

Fig. 2. Ranking of contribution scores for each monitoring parameter.

The proposed module's detailed workflow is as follows. First, the network takes training samples as input, multiplies them by network parameters and biases, and applies the sigmoid activation function to limit output weights to (0, 1). The output layer has 4 neurons to output respective weights for the four monitoring parameters. The formula for the FCNN can be expressed as

$$\mathbf{F} = \sigma(\mathbf{W}\mathbf{u} + \mathbf{b}) \tag{4}$$

where **F, W, b** denote the output vector, weight matrix, and bias vector of the first hidden layer, respectively. $\sigma(\cdot)$ represents the sigmoid activation function. Subsequently, **F** is fed to the output layer to output the attentional weights of the four monitoring parameters. The equation for the output layer can be formulated as

$$\left[w_s, w_f, w_r, w_a\right]^T = \sigma(\mathbf{W}_{\text{out}}\mathbf{F} + \mathbf{b}_{\text{out}}) \tag{5}$$

where w_s, w_f, w_r and w_a represent the attention weights for the monitoring parameters SPP, FOP, ROP, and SUM, respectively. $\mathbf{W}_{\text{out}}$ denotes the weight matrix of the output layer. $\mathbf{b}_{\text{out}}$ represents the bias vector of the output layer.

Then, attention fusion between data sequences and attention weights is performed. This enhances the contribution of key parameters, especially FOP, to subsequent kick detection. Specifically, the original data sequences of each monitoring parameter are multiplied by its corresponding attention weight. For the normalized data sequences (each with length l) of the four monitoring parameters $\mathbf{R}_s = [u_{s1}, u_{s2}, \ldots, u_{sl}]$, $\mathbf{R}_f = [u_{f1}, u_{f2}, \ldots, u_{fl}]$, $\mathbf{R}_r = [u_{r1}, u_{r2}, \ldots, u_{rl}]$, and $\mathbf{R}_a = [u_{a1}, u_{a2}, \ldots, u_{al}]$ in a training sample, the weighted data sequences can be calculated as

$$\begin{cases} \mathbf{R}'_s &= w_s \mathbf{R}_s = [w_s u_{s1}, w_s u_{s2}, \ldots, w_s u_{sl}] \\ \mathbf{R}'_f &= w_f \mathbf{R}_f = \left[w_f u_{f1}, w_f u_{f2}, \ldots, w_f u_{fl}\right] \\ \mathbf{R}'_r &= w_r \mathbf{R}_r = [w_r u_{r1}, w_r u_{r2}, \ldots, w_r u_{rl}] \\ \mathbf{R}'_a &= w_a \mathbf{R}_a = [w_a u_{a1}, w_a u_{a2}, \ldots, w_a u_{al}] \end{cases} \tag{6}$$

Following the weighted sequences calculation above, it is necessary to clarify the mechanism of how these attention weights guide the model's focus on specific monitoring parameters. Neural networks inherently tend to focus on monitoring parameters with larger data ranges. This is precisely why data normalization is commonly required. It prevents models from excessive focus on monitoring parameters with larger numerical ranges. When different attention weights are multiplied by data sequences, the data range of each parameter's sequence is adjusted. Sequences of parameters with larger weights, such as FOP, will have their numerical ranges amplified. Therefore, the model is naturally inclined to allocate more attention to FOP during feature extraction and classification.

Importantly, since all data points within a monitoring parameter's sequence are multiplied by the same weight, the original variation trends remain unchanged. Only the magnitude of the values is scaled. This preserves the temporal patterns necessary for kick detection.

It is worth noting that the weights assigned by the FCNN require optimization to ensure they can enhance the recognition performance of the KG-Bi-LSTM. Thus, in designing the FCNN's loss function, the KG-Bi-LSTM's loss function ($L_{\text{knowledge}}$) should be included as a component. Additionally, the weight allocation network's output should align with the prior knowledge that FOP is more critical. Accordingly, a loss function guidance term is proposed. When the attention weight assigned to FOP by the FCNN is lower than that assigned to the other monitoring parameters, the corresponding loss value is increased, i.e., the FCNN is penalized, which forces it to update its network parameters and optimize its output to maintain high attention to FOP. Thus, the loss function of the attention weight allocation network consists of $L_{\text{knowledge}}$ and a guidance term, and its formula can be expressed as

$$\begin{aligned} L_{\text{Att}} &= L_{\text{knowledge}} + L_{\text{FOP}} \\ &= L_{\text{knowledge}} + \gamma \max\left(0, \max(w_s, w_r, w_a) - w_f\right) \end{aligned} \tag{7}$$

where L_{Att} denotes the loss function of the weight allocation network, and γ is the penalty coefficient. After completing the allocation of attention weights, the weighted data sequences are fed into the intelligent recognition model for training.

2.3. Construction of the Bi-LSTM network based intelligent early kick recognition module

The intrusion of formation fluid into the annulus is a continuous process after a kick incident occurs (Tang et al., 2019). Thus, the intelligent kick recognition model should recognize and memorize the temporal variation trends of monitoring parameters. Consequently, the LSTM neural network becomes a suitable choice due to its ability to capture and retain abnormal variation patterns in monitoring parameters.

To effectively capture and memorize the variation patterns in monitoring parameter variations, the LSTM network employs two components for memory storage: the cell state for long-term memory and the hidden state for short-term memory. Additionally, the LSTM uses the input gate, forget gate, and output gate to selectively remember and forget variation trends of monitoring parameters. Specifically, variation trends corresponding to kicks are remembered, while irrelevant ones are forgotten. To effectively learn these variation trends, the LSTM does not directly remember data sequences of monitoring parameters. Instead, it encodes the variation trends in the input data and stores them as features.

The structure of the LSTM unit is shown in Fig. 4. The core components of the LSTM are the memory cell C_t and the hidden state h_t, which respectively store long-term and short-term memory. Their formulas can be expressed as

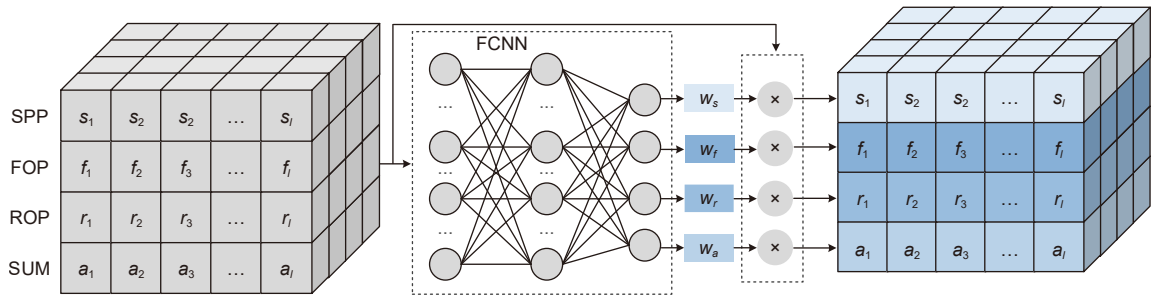

Fig. 3. Structural diagram of the attention allocation module for monitoring parameters.

$$\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{8}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t \tag{9}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{10}$$

where $\widetilde{C}_t$ denotes the candidate cell state, which is used to update C_t; W_C is the weight of $\widetilde{C}_t$; b_C is the bias term; tanh is the hyperbolic tangent activation function; f_t, i_t, o_t represent the forget gate, the input gate, and the output gate, respectively, which are used to update long-term and short-term memories; x_t is the input at time t; h_{t-1} is the hidden state at time $t-1$.

Specifically, f_t is used to discard trends of monitoring parameters in C_t that are irrelevant to kicks or normal drilling. i_t filters which inputs at the current time t need to be stored in C_t, while o_t is used to generate the short-term memory (h_t) for the update of C_{t+1} at the next time $t+1$. The formulas for f_t, i_t, o_t can be expressed as

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{11}$$

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{12}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{13}$$

where x_t is the input at time t; h_{t-1} represents the hidden state at time $t-1$; W_i, W_f, W_o are the weights of i_t, f_t, o_t, respectively; b_i, b_f, b_o are the biases of i_t, f_t, o_t, respectively.

After the two LSTMs have extracted the trends of the detected parameters from the forward and backward directions, respectively, the two trend features are concatenated to generate the feature matrix. This process can be formulated as

$$H = \text{concat}\left(\left[\overrightarrow{h}_1, \overrightarrow{h}_2, \ldots, \overrightarrow{h}_T\right], \left[\overleftarrow{h}_1, \overleftarrow{h}_2, \ldots, \overleftarrow{h}_T\right]\right) \tag{14}$$

where H denotes the concatenated feature matrix. $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ represent the hidden states at time t ($t = 1, 2, \ldots, T$) from the forward LSTM and backward LSTM, respectively. T is the total number of time steps in the input sequence. concat(·, ·) denotes the concatenation operation. The structure of the Bi-LSTM is shown in Fig. 5. Finally, H is fed into the output layer to generat

where $\mathcal{D}_{nh}$ and $\mathcal{D}_{ne}$ represent hard-to-recognize and easy-to-recognize normal samples, respectively.

The above knowledge of kick recognition can be used to guide the data-driven model to focus on learning the critical variation trends of monitoring parameters during the early stages of kicks. Therefore, the prior knowledge will be transformed and embedded into the loss function in the following subsection to construct a knowledge-guided intelligent kick recognition model.

2.4.2. The construction of a knowledge-guided loss function for the intelligent kick recognition model

According to the analysis of the challenges in early kick recognition, the knowledge-guided loss terms are designed based on the following principles. First, since normal samples are numerous and relatively easy to classify, their contribution to the total loss should be reduced. This allows the kick recognition model to focus more on learning from kick samples. Second, to improve the model's accuracy in identifying early kick samples, it is necessary to emphasize the loss values corresponding to these samples, thereby encouraging the model to focus more on learning early kick samples. The detailed procedure for designing knowledge-guided loss terms is as follows.

(1) **The formulation of the loss term for kick samples**

Increasing the loss values corresponding to kick samples helps elevate their contribution to the total loss. This approach could guide the model to focus more on learning from these samples. Additionally, given that early kick samples are more difficult to recognize than other ones, it becomes a good choice to employ a monotonically decreasing function to assign different loss value weights to different kick samples. Samples corresponding to the early stage of kicks are given larger weights. The selection of a monotonically decreasing function as the weighting function for the loss values can take various forms, such as linear decay or exponential decay, and the choice among these options should be determined based on domain knowledge.

The abnormal variation trends of monitoring parameters after a kick occurrence can serve as knowledge for guiding the selection of an appropriate weighting function. Kick recognition requires the identification of the variation trends of FOP and SUM. However, SUM exhibits a slower response than FOP when a kick occurs. Therefore, whether the variation trends of FOP are apparent or not is directly related to the classification difficulty of the sample. The design of the weighting function should follow the variation patterns of FOP after kicks occur. The variation patterns of FOP are directly related to the causes and development patterns of kicks. The direct cause of a kick is the imbalance of downhole pressures. During the normal drilling process, the downhole pressure is equal to the sum of the hydrostatic column pressure and the annular pressure loss (Liu et al., 2024). Downhole pressure can be expressed as

$$\begin{aligned} P_{\text{downhole}} &= P_{\text{hydro}} + P_{\text{annular}} \\ &= \rho g h_{\text{well}} + \frac{2 f_{\text{fluid}} \rho v_{\text{ave}}{}^2 h_{\text{well}}}{D_{\text{inner}}} \end{aligned} \tag{19}$$

where P_{hydro} and P_{annular} represent the hydrostatic column pressure and the annular pressure loss, respectively. ρ represents drilling fluid density, and h_{well} represents the well depth. The gravitational acceleration is represented by g, and f_{fluid} stands for the friction coefficient. The average flow rate of the drilling fluid is denoted by v_{ave}, and D_{inner} denotes the inner diameter of the drill pipe.

When $P_{\text{formation}}$ is greater than the sum of P_{hydro} and P_{annular}, kick occurs and the intrusion of lower density formation fluid into the wellbore decreases ρ. The decrease in ρ causes a further imbalance in downhole pressures and an increasing rate of formation fluid intrusion. As a result, the FOP exhibited a slight upward trend at the early stage of a kick, and then the overflow rate of drilling fluid became increasingly faster, as shown in Fig. 7(a). Therefore, it becomes a good choice to use a monotonically decreasing exponential function to weight the loss values corresponding to kick samples. The loss function after weighting the loss values is formulated as

$$L_{\text{kick}} = -\sum_{c=1}^{C}\sum_{j=1}^{J}(1 + w_{\max}\exp(-r(j-1)))\log\left(\widehat{p}_j\right) \tag{20}$$

where C and J represent the number of kick cases and samples per kick case, respectively. The samples in each kick case are assigned an additional loss value. The weight of the additional loss value for the first sample after a kick is $w_{\max}$, and the weights for the remaining samples decrease over time. r is the decay rate of the exponential function that controls the weight decay rate. exp is the natural base. $\widehat{p}_j$ is the probability that the current sample belongs to a kick sample, as predicted by Bi-LSTM.

(2) **The formulation of the loss term for normal samples**

Normal samples are less difficult to classify and more abundant than kick samples. The class of samples with the larger number contributes more to the total loss value, thereby dominating the updating of the network parameters. As a result, the model tends to learn from the samples of the majority class, leading to insufficient learning of kick samples.

However, balancing the proportion of different classes by reducing the number of normal samples in the training set is not appropriate. Data-driven models should learn the variation trends of monitoring parameters from all available samples. Reducing the number of normal samples may result in the underutilization of limited data resources. When the training set is imbalanced, reducing the loss values for the majority class is an effective way to balance the model's attention. A commonly used method is to multiply the loss value of normal samples by an attenuation factor between (0, 1). The formula for the normal sample loss term after weighing using the weight reduction factor can be expressed as

$$L_{\text{normal}} = -\sum_{i=1}^{I} w_{\text{scale}}\log(1-\widehat{y}_i) \tag{21}$$

where w_{scale} represents the attenuation factor. I is the total number of normal samples. $\widehat{y}_i$ represents the prediction of the model for the i-th normal sample. However, there is also a difference in importance among the normal samples. Hard-to-classify normal samples are more important than easy-to-classify ones. The reduction in loss values for all normal samples may prevent the intelligent model from adequately learning the variation trends of monitoring parameters in hard-to-classify normal samples, potentially leading to false alarms in real applications.

A reasonable approach to weight the loss values of normal samples is to dynamically adjust the weights based on the classification difficulty of each sample. Specifically, the loss values of easily classifiable normal samples should be significantly reduced, while the loss value reduction should be less for those that are harder to classify. In data-driven models, the magnitude of the prediction error reflects the extent to which the model fits samples. Since normal samples are labeled as 0, the model's output $\widehat{y}_i$ represents the prediction error. In order to reduce the loss value of the easy-to-classify samples while ensuring loss values for hard-

to-classify samples do not decrease excessively, the square of the $\widehat{y}_i$ is used as the attenuation factor for the loss value corresponding to normal samples. In this way, the model's attention on easily classifiable samples can be effectively reduced, thereby encouraging the model to focus more on the hard-to-identify samples. The loss term of normal samples can be expressed as

$$L_{\text{normal}} = -\sum_{i=1}^{I} \widehat{y}_i^2 \log(1-\widehat{y}_i) \tag{22}$$

This dynamic weighting method allows the model to reduce its focus on easy-to-classify samples and put more focus on kick samples.

early kicks. The constraint term based on weight decay can be formulated as

$$L_{\text{constraint}} = \lambda \sum_{z=1}^{Z} \theta_z^2 \tag{23}$$

where λ represents the constraint strength. θ_z represents the z-th weight value of the network. Z is the total number of weight values.

Finally, the proposed knowledge-guided loss function can be formulated as follows

During the training process, L_{normal} reduces the model's focus

$$\begin{aligned} L_{\text{knowledge}} &= L_{\text{normal}} + L_{\text{kick}} + L_{\text{constraint}} \\ &= -\sum_{i=1}^{I} \widehat{y}_i^2 \log(1-\widehat{y}_i) + \left(-\sum_{c=1}^{C} \sum_{j=1}^{J} (1 + w_{\max} \exp(-r(j-1))) \log\left(\widehat{p}_j\right) \right) + \lambda \sum_{z=1}^{Z} \theta_z^2 \end{aligned} \tag{24}$$

(3) **The constraint term formulation**

In addition to employing knowledge to guide the learning process of data-driven models, it is also crucial to impose constraints on the model's learning process. L_{normal} and L_{kick} are designed to prevent the model from underfitting key trends, such as the abnormal monitoring parameter variation trends of early kicks. However, the increase of loss value for certain samples may cause the neural network to overlearn these samples. This results in the overfitting problem of intelligent models, characterized by high accuracy on training samples but poor recognition performance on unseen ones. This phenomenon is especially common in application scenarios where the number of training samples is limited. It is essential to constrain the model's tendency to overfit during its learning process.

Consequently, a constraint term based on weight decay is introduced into the loss function to prevent overfitting that may caused by the guidance of knowledge. As a regularization technique, weight decay serves to prevent overfitting by constraining the excessive growth of network weights. During the training process, the neural network adjusts network weights according to the loss values using the back-propagation algorithm. The values of the network weights rise during the above process. If the kick recognition model overfits to the training samples, the magnitudes of its network parameters tend to increase substantially. Therefore, to effectively prevent overfitting, a commonly used approach is to incorporate the sum of squared network parameters into the loss function as a constraint term. When the model overfits, the loss value calculated from the constraint term rises significantly. In order to minimize the total loss, intelligent models tend to reduce the degree to which they fit the training samples, thereby achieving suppression of overfitting. During the training process, the knowledge-guided loss terms and the constraint term work collaboratively to help the model avoid both underfitting and overfitting, and to facilitate the learning of generalized features of on easily identifiable normal samples, and L_{kick} encourages the model to focus on learning early kick samples. $L_{\text{constraint}}$ constrains the training process from overfitting.

3. Experimental results and analysis

To validate the effectiveness of the proposed early kick recognition model based on KG-Bi-LSTM, 4,599 samples derived from 10 kick cases were utilized for the training, validation, and testing of the method. Both the Bi-LSTM and KG-Bi-LSTM models were trained using these samples, followed by a comparison of their recognition accuracy and alarm timeliness on the test set. Ablation experiments were subsequently performed to verify the effectiveness of each proposed improvement in enhancing recognition performance. A fixed random seed was used to control experimental variables and ensure the reproducibility of the experiments. The experiment was conducted on a platform equipped with an AMD Ryzen 7 7500F CPU, an AMD Radeon RX 7800 XT GPU (16 GB VRAM), and 32 GB of RAM. PyCharm Community version 2022.2.2 was used.

3.1. Data preparation

3.1.1. Data normalization

Monitoring parameter data sequences are collected by different sensors, each with a distinct scale. Data with larger values has a greater impact on the weight update of the network. Therefore, the data should be mapped to the same range by normalization. The Min–Max method is used to normalize the monitoring parameters, and its equation can be formulated as

$$x_i^{\text{norm}} = \frac{\tau_i - \min\limits_{1\le i\le l_n}(\tau_i)}{\max\limits_{1\le i\le l_n}(\tau_i) - \min\limits_{1\le i\le l_n}(\tau_i)} \tag{25}$$

Table 3
Recognition accuracies corresponding to different L_w.

	L_w = 10 (50 s)	L_w = 20 (100 s)	L_w = 30 (150 s)	L_w = 40 (200 s)	L_w = 50 (250 s)
Accuracy, %	88.07	90.34	92.61	92.05	90.91

where I_n is the number of data. τ_i and x_i^{norm} are the values of the i-th data before and after normalization, respectively. $\max(\tau_i)$ and $\min(\tau_i)$ represent the maximum and minimum values of data sequences, respectively.

3.1.2. Sample generation and dataset segmentation

The data used in this study were collected from ten wells in Blocks M and K of the SL Oilfield between 2020 and 2021. The sampling interval is 5 s, corresponding to a sampling frequency of 0.2 Hz. Prior to training the intelligent model, it is necessary to use a sliding window over the original data sequence to segment and label data fragments for sample generation. The length of the sliding window directly determines the time coverage corresponding to the monitoring parameter data sequence in each sample. Given that the data sampling interval is 5 s, the window length should therefore range between 20 and 30. In order to determine the optimal window length, the samples corresponding to different window lengths from 10 to 50 were used to train the developed model, with recognition accuracies produced. The recognition accuracies corresponding to different window lengths are presented in Table 3.

The results indicate that the samples generated using a window length of 30 correspond to the highest recognition accuracy. Thus, the window length was determined as 30. The process of generating training samples using sliding windows is illustrated in Fig. 9.

After the samples are generated, they need to be divided into training, validation, and test sets. A total of 10 kick cases were used, with 8 for training, and the remaining 2 for testing and validation.

The well names of the selected cases, the number of normal and kick samples, as well as the duration of each case, are presented in Table 4. The training, test, and validation sets consist of a total of 4,599 samples. The training set comprises 4,063 samples, including 2,416 normal samples and 1,647 kick samples, with a ratio of approximately 1.5:1. This training set is an imbalanced sample set with unequal sample sizes across different classes, and it is used to evaluate the performance of the proposed method in this application scenario.

3.2. Evaluation metrics

In order to comprehensively evaluate the influence of the proposed knowledge-guided loss function on the performance of the kick recognition model, five evaluation metrics, including accuracy, recall, precision, F1-Score, and area under the curve (AUC), were used. To calculate the aforementioned evaluation metrics, the confusion matrix as listed in Table 5 should be computed first. It consists of true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

Among all evaluation metrics, accuracy is the most intuitive metric. It is defined as the proportion of correctly classified samples to the total number of samples, and can be expressed mathematically as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{26}$$

Recall measures the model's ability to correctly identify kick samples. It represents the proportion of correctly identified kick samples to the total number of actual kick samples. The mathematical expression of recall can be formulated as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{27}$$

Precision measures the proportion of actual kick samples among those predicted as kick samples by the model. A high precision indicates a low possibility that the model misclassifies a normal sample as a kick sample. The formula for precision is as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{28}$$

The F1-Score evaluates the model's performance in identifying samples, particularly in cases of sample class imbalance. It is calculated using recall and precision, and can be formulated as

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{29}$$

In addition, AUC reflects the model's distinguish ability for two classes of samples. The value of AUC is equal to the area under the curve plotted with false positive rate (FPR) as the horizontal axis and true positive rate (TPR) as the vertical axis. The closer the AUC is to 1, the better the model's classification ability is. FPR and TPR can be formulated as

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{30}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{31}$$

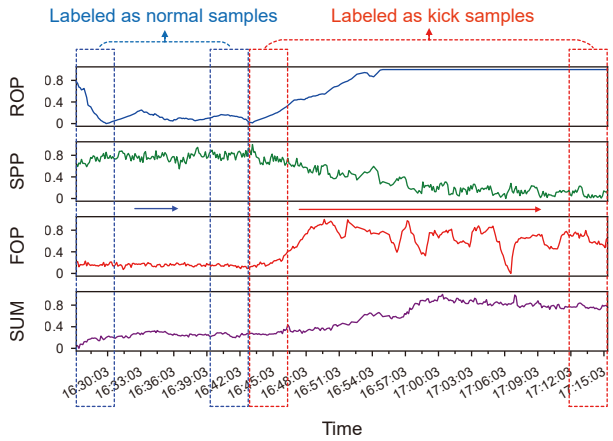

Fig. 9. Workflow for generating samples using sliding window.

3.3. Determination of hyperparameters

The proposed knowledge-guided attention allocation model and the loss function involve 6 hyperparameters, which are listed in Table 6. Grid search is adopted to determine the optimal values of these hyperparameters. Specifically, the value range for each hyperparameter is first defined, with several candidate values generated within these ranges. Subsequently, different hyperparameter combinations are created by combining the candidate values of all hyperparameters. A corresponding model is then trained for each combination, and its recognition accuracy on the validation set is evaluated. Finally, the hyperparameter combination corresponding to the highest recognition accuracy is determined as the optimal one.

However, with 6 hyperparameters to optimize, adjusting all six hyperparameters simultaneously would lead to an excessive number of hyperparameter combinations, significantly increasing computational complexity. Notably, the attention weight

Table 4
Details of the kick cases used in the experiment.

Well name	Number of normal samples	Number of kick samples	Total number of samples	Usage	Duration of the case, h
LD4	338	199	537	Training set	0.75
TY6	86	476	562	Training set	0.78
QA2	484	96	580	Training set	0.81
YX4	21	213	234	Training set	0.33
BX7	782	259	1,041	Training set	1.45
GQ2	447	180	627	Training set	0.87
CX	99	12	111	Training set	0.15
BM	159	212	371	Training set	0.52
TE1	118	242	360	Test set	0.50
TW1	88	88	176	Validation set	0.24

Table 5
Confusion matrix.

	Kick	Non-kick
Kick	TP	FN
Non-kick	FP	TN

Table 6
List of hyperparameters to be optimized.

Hyperparameters of the attention allocation model	Hyperparameters of the proposed loss function
Number of network layers for FCNN: N_f	The maximum additional loss value weight: w_{max}
Number of neurons in each layer of FCNN: N_u	The decay rate of the loss value: r
Coefficient of the L_{FOP}: γ	Coefficient of the $L_{constraint}$: λ

allocation network serves as an auxiliary module, assisting the Bi-LSTM in monitoring early kicks. In contrast, the aforementioned loss function ($L_{knowledge}$) is specifically designed for the Bi-LSTM, which serves as the core network of the proposed method. Therefore, to reduce the complexity of the hyperparameter value search process, the hyperparameter values of the loss function are first determined, followed by the optimization of hyperparameters for the attention allocation network.

In the loss function, r denotes the decay rate of the weight assigned to the additional loss. Too large r leads to overly rapid weight decay, with loss of weights only allocated for the first few early kick samples. Conversely, too small r results in slow weight decay, where all kick samples have weights close to w_{max}. This may make the model overlearn kick samples and overfit. In most adopted cases, the number of kick data points ranges from 100 to 500. Fig. 10 illustrates the weight decay trends for r values between 0.001 and 0.1 within this data point range. As shown, when r falls in [0.001, 0.1], the weight distribution across different kick data point ranges (from $[0-100]$ up to $[0-500]$) reveals that early kick samples have the highest weights, with subsequent samples showing a time-dependent decay. The weight differences between individual kick samples are distinct but not excessively large. Beyond this range, r leads to either excessively fast or slow decay. Consequently, r is set to the values [0.1, 0.01, 0.001].

w_{max} denotes the maximum weight assigned to the additional loss values of kick samples. During model training and validation, it was observed that when w_{max} is set to 1, 2, or 3, the constraint term does not fail due to an excessively small w_{max}, nor does the model suffer from overfitting induced by an excessively large w_{max}. Thus, these values were selected as the preset values for w_{max}. λ denotes the constraint strength of the regularization term. An excessively large λ inhibits the model's ability to learn sample features, while an overly small λ fails to suppress overfitting.

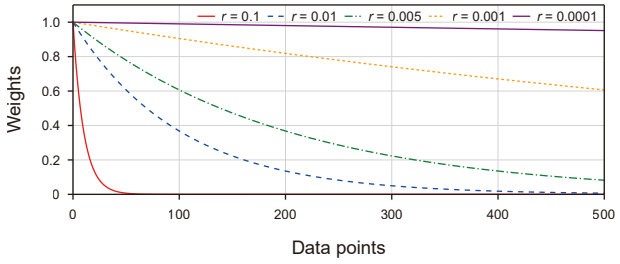

Fig. 10. Curves of the exponential decay function corresponding to different r.

Typically, λ is set to negative integer powers of 10. Thus, the preset values here are 0.1, 0.01, and 0.001. Experimental results are presented in Table 7, which shows the highest model recognition accuracy when $r = 0.01$, $w_{max} = 2$, and $\lambda = 0.01$. These values are therefore determined as the hyperparameters for the loss function.

The hyperparameter values of the attention allocation model are optimized in a similar way. Since this auxiliary network does not undertake the primary task of early kick recognition, its complexity is typically lower than that of the Bi-LSTM. In practice, the number of layers and neurons per layer in such auxiliary networks is generally fewer. Thus, the number of layers is set to 1 or 2, and the number of neurons per layer is set to 16 or 32. Additionally, the regularization weight γ of the attention allocation network controls the strength of knowledge constraints. Similar to the regularization term in $L_{knowledge}$, γ is typically set to negative integer powers of 10. Thus, it is set to 0.1, 0.01, and 0.001, respectively. Experimental results in Table 8 indicate that the highest model recognition accuracy is achieved when N_f, N_u, and γ are 2, 16, and 0.01, respectively. These values are therefore adopted.

3.4. Performance comparison between KG-Bi-LSTM and Bi-LSTM

To compare the performance of KG-Bi-LSTM and Bi-LSTM in kick recognition, both models were trained on the training set, and their evaluation metrics were calculated according to their recognition results on the test set. The changes in the loss values of each loss term during the training process of KG-Bi-LSTM are shown in Fig. 11. The experimental results are listed in Table 9.

The recognition results of the two networks for the test set are shown in Fig. 12. This case was collected from Well TE1 in Block M of the SL Oilfield in November 2020. At approximately 16:44 on the same day, a kick occurred while drilling this well at a depth of 3,388.5 m. The Bi-LSTM model reported an alarm at 16:48:16, whereas the proposed KG-Bi-LSTM model issued an alarm at 16:46:30, advancing the alarm time by nearly 2 min.

Compared with the Bi-LSTM, the accuracy, recall, F1-Score, and AUC of the proposed KG-Bi-LSTM are improved by 5.28%, 7.86%, 4.40%, and 7.12%, respectively. Notably, the accuracy and recall

Table 7
Accuracy performance under different hyperparameter combinations of $L_{knowledge}$.

$w_{max} = 1$			$w_{max} = 2$			$w_{max} = 3$		
λ	r	Accuracy, %	λ	r	Accuracy, %	λ	r	Accuracy, %
0.1	0.1	82.95	0.1	0.1	89.77	0.1	0.1	90.91
0.1	0.01	75.57	0.1	0.01	88.07	0.1	0.01	85.80
0.1	0.001	88.64	0.1	0.001	92.05	0.1	0.001	88.07
0.01	0.1	84.66	0.01	0.1	90.34	0.01	0.1	89.20
0.01	0.01	81.25	0.01	0.01	93.18	0.01	0.01	83.52
0.01	0.001	89.20	0.01	0.001	92.61	0.01	0.001	79.55
0.001	0.1	88.64	0.001	0.1	91.48	0.001	0.1	90.34
0.001	0.01	85.23	0.001	0.01	90.91	0.001	0.01	84.09
0.001	0.001	77.84	0.001	0.001	86.36	0.001	0.001	87.50

Table 8
Accuracy performance under different hyperparameter combinations of the attention allocation module.

$N_f = 1$			$N_f = 2$		
N_u	γ	Accuracy, %	N_u	γ	Accuracy, %
16	0.1	92.05	16	0.1	95.45
16	0.01	94.32	16	0.01	96.59
16	0.001	90.91	16	0.001	94.89
32	0.1	93.18	32	0.1	94.32
32	0.01	90.34	32	0.01	91.48
32	0.001	93.75	32	0.001	92.61

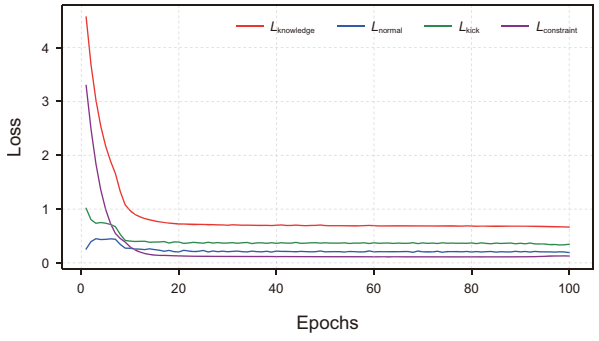

Fig. 11. Variation of loss values during the KG-Bi-LSTM training process.

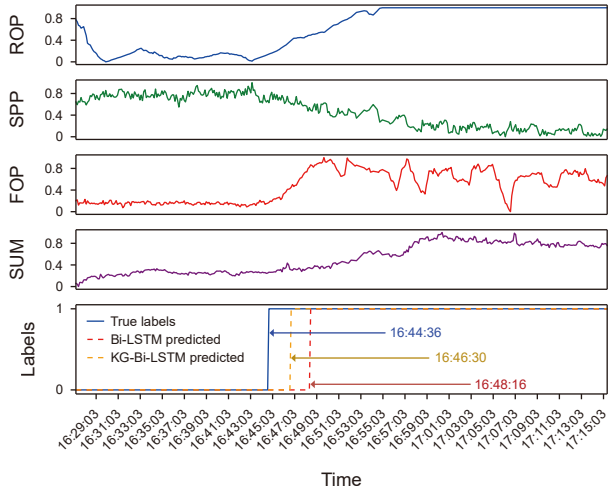

Fig. 12. Comparison of recognition results of Bi-LSTM and KG-Bi-LSTM on a case in the test set.

exhibit a significant increase, indicating a substantial enhancement in the model's ability to correctly identify kick samples. Meanwhile, both F1-Score and AUC show varying degrees of improvement, demonstrating a significant enhancement in the model's overall classification performance.

Additionally, the KG-Bi-LSTM adopts a constraint term $L_{constraint}$ to mitigate overfitting. To demonstrate the anti-overfitting capability of KG-Bi-LSTM, the variations in recognition accuracy of Bi-LSTM and KG-Bi-LSTM on the training and test sets during training are presented, as shown in Figs. 13 and 14. It can be observed from Fig. 13 that the accuracy of Bi-LSTM on the training set approaches 100%. This phenomenon is typically a sign of overfitting. In contrast, the recognition accuracy of KG-Bi-LSTM on the training set during training is approximately 90%, which indicates that it may not overfit to the training samples.

To further validate this assumption, the recognition accuracy trends of both models are analyzed on the test set, whose results are illustrated in Fig. 14. It can be seen that the accuracy of KG-Bi-LSTM gradually increases with the increase of epochs and reaches its maximum when epoch = 100. Subsequently, the model starts to overfit, and its recognition accuracy for the test set decreases. This indicates that under the combined effect of the guidance terms and the constraint term, the model achieves a balance between underfitting and overfitting when training is completed at the epoch of 100, at which point it attains optimal performance. When epoch >100, this balance is disrupted, and the constraint term can no longer inhibit overfitting induced by overtraining, leading to a degradation in model performance. Thus, this result indicates that

Table 9
Performance comparison of Bi-LSTM and KG-Bi-LSTM.

Evaluation metrics	Bi-LSTM model	KG-Bi-LSTM model	Performance improvement, %
Accuracy, %	90.00	95.28	5.28 ↑
Recall, %	85.12	92.98	7.86 ↑
Precision, %	100.00	100.00	0.00
F1-Score, %	91.96	96.36	4.40 ↑
AUC, %	91.22	98.34	7.12 ↑

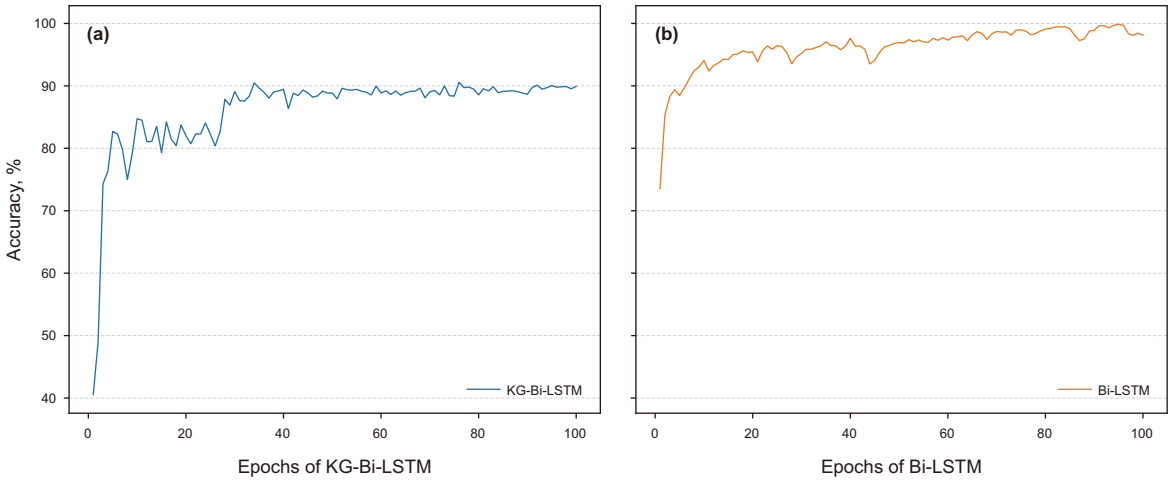

Fig. 13. Trends in recognition accuracy of KG-Bi-LSTM **(a)** and Bi-LSTM **(b)** on the training set during training.

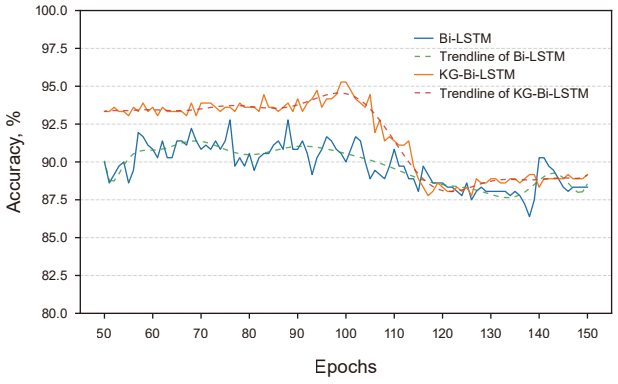

Fig. 14. Trends in recognition accuracy of KG-Bi-LSTM and Bi-LSTM on the test set during training.

increasing the model's training epochs requires simultaneous enhancement of constraint strength, thereby preventing overfitting. In contrast, the Bi-LSTM without constraint terms exhibited a declining trend in recognition accuracy around 100 training iterations, indicating overfitting.

3.5. Ablation experiments and results analysis

3.5.1. Ablation experiments and results analysis for the KG-attention and $L_{knowledge}$

In this paper, a prior knowledge-guided monitoring parameter attention allocation module (KG-Attention) and a knowledge-guided loss function (i.e., $L_{\text{knowledge}}$) are proposed to improve the accuracy of intelligent models in early kick identification. To verify the effectiveness of the proposed components, ablation experiments were conducted on four model variants, with the results presented in Table 10. These results show that the model integrating KG-Attention, Bi-LSTM, and $L_{\text{knowledge}}$ (i.e., the proposed KG-Bi-LSTM) achieves the highest performance (accuracy: 95.28%). It outperforms the baseline Bi-LSTM model (accuracy: 90.00%), as well as the single-component variants including the Bi-LSTM with KG-Attention (accuracy: 91.11%) and the Bi-LSTM with $L_{\text{knowledge}}$ (accuracy: 91.67%). This demonstrates that each of the two proposed modules is capable of enhancing the accuracy of early kick detection. Notably, the combination of these two modules yields the most significant improvement in accuracy, thereby confirming a synergistic effect between them.

Furthermore, the attention scores assigned by the attention allocation network to the four monitoring parameters during the testing process are illustrated in Table 11. Among these parameters, FOP has the highest weight, indicating that the proposed prior knowledge-constrained loss term (i.e., L_{FOP}) is effective. The weights of the remaining three parameters are automatically allocated by the attention allocation network. It can be seen that SUM has the second-highest weight, followed by ROP and SPP. This

Table 11
Attention weights allocated by the attention allocation network in the testing process.

Monitoring parameters	ROP	SPP	FOP	SUM
Attention weights	0.659	0.585	0.732	0.729

Table 12
Candidate loss terms for the loss function.

Candidate loss terms for normal samples	Candidate loss terms for kick samples	Constraint term
L_{BN}	L_{BK}	–
L_{normal}	L_{kick}	$L_{\text{constraint}}$

Table 10
Performance comparison of different ablation models.

Models	Bi-LSTM	KG-Attention + Bi-LSTM	Bi-LSTM + $L_{\text{knowledge}}$	KG-Attention + Bi-LSTM + $L_{\text{knowledge}}$
Accuracy, %	90.00	91.11	91.67	95.28
Precision, %	1.00	1.00	1.00	1.00
Recall, %	85.12	86.78	87.60	92.98
F1-Score, %	91.96	92.92	93.39	96.36
AUC, %	91.22	98.82	99.18	98.34

demonstrates that during the recognition process of the proposed method, the model is more sensitive to the changes in mud parameters. Meanwhile, the model does not completely ignore changes in engineering parameters. Instead, it makes comprehensive judgments by integrating the variation trends of mud and engineering monitoring parameters.

3.5.2. Ablation experiments and results analysis for loss terms in $L_{knowledge}$

In order to investigate the influence of the proposed L_{normal}, L_{kick}, and $L_{constraint}$ on the performance of the early kick recognition model, ablation experiments were designed to evaluate their contributions. The loss terms corresponding to normal and kick samples in Eq. (15) are denoted by L_{BN} and L_{BK}, respectively. The loss functions used in the ablation experiments consist of three components, including loss terms for normal and kick samples, as well as a constraint term. As listed in Tables 12 and 13, there are two options available for each loss term, leading to a total of eight possible combinations. Excluding L_{BCE} and $L_{knowledge}$, the remaining six loss term combinations are named Combo-1 to Combo-6. The combinations and their corresponding loss terms are listed in Table 13. Additionally, the performance changes of Bi-LSTM using different combinations of loss terms compared with those using L_{BCE} are listed in Table 14.

It can be seen from Table 14 that the Bi-LSTM using Combo-1 as the loss function demonstrated improvements among evaluation metrics compared with the one using L_{BCE}. The loss term for normal samples in Combo-1 employs the proposed L_{normal}. It reduces the loss values corresponding to normal samples, thereby guiding the model's attention to kick samples. Compared with the Bi-LSTM using L_{BCE} as the loss function, the one with Combo-1 achieved a 2.90% increase in recall. This verifies that the Bi-LSTM's recognition performance for kick samples has been significantly improved. After incorporating $L_{constraint}$ into Combo-1, Combo-4 was derived. Notably, compared to Combo-1, Combo-4 exhibits significant improvements in recognition accuracy (+2.12%), recall (+4.63%), and F1-Score (+2.56%). This confirms that $L_{constraint}$ effectively prevents model overfitting, and its incorporation is the key reason for the significant enhancement in model performance.

Among all the combinations of loss terms, Combo-3 and Combo-6 are the most representative. Combo-3 employs L_{BN} and L_{BK}, along with a constraint term, but does not utilize the proposed guidance terms. In contrast, Combo-6 incorporates the guidance terms (L_{normal} and L_{kick}), but lacks the constraint term. As a result, Combo-3 only constrains the model's training process without providing guidance, while Combo-6 guides the training process without imposing any constraints. The Bi-LSTM models using Combo-3 and Combo-6 as loss functions are referred as Model-3 and Model-6, respectively. The recognition performance of Model-3 and Model-6 is compared with that of Bi-LSTM and KG-Bi-LSTM, as shown in Fig. 15.

It is evident that the model using Combo-3 as the loss function performs the worst. This may be attributed to Combo-3 only including a constraint term without any guidance term. During the training process, the guidance term and the constraint term collaborate to ensure the model achieves a balance between underfitting and overfitting. However, the absence of the guidance term in Combo-3 disrupts the balance between knowledge guidance and weight constraint. In this case, the constraint term acts to suppress the model's learning from the training samples, leading to a decline in recognition performance. However, in application scenarios such as kick monitoring, where the number of available samples is limited, applying constraints with appropriate strength during the training process helps to prevent the model from overfitting. Fig. 16 illustrates the changes in model recognition performance as the constraint strength in Combo-3 decreases. The model's recognition performance initially increases and then decreases as λ decreases, indicating that an appropriate constraint strength positively enhances model performance.

The performance comparison between Combo-6 and the $L_{knowledge}$-corresponding model further confirms that constraints with appropriate strength can enhance model performance by suppressing overfitting. Specifically, after incorporating constraint terms into Combo-6, which only contains guidance terms, the model's evaluation metrics, such as recognition accuracy (+4.45%) and recall (+6.62%), improved significantly.

Table 13
Performance comparison of Bi-LSTM using different combinations of loss terms.

Loss functions	Loss term for normal sample	Loss term for kick sample	Constraint term	Accuracy, %	Recall, %	Precision, %	F1-Score, %	AUC, %
L_{BCE}	L_{BN}	L_{BK}	–	90.00	85.12	100.00	91.96	91.22
Combo-1	L_{normal}	L_{BK}	–	91.94	88.02	100.00	93.63	99.23
Combo-2	L_{BN}	L_{kick}	–	90.00	85.12	100.00	91.96	98.67
Combo-3	L_{BN}	L_{BK}	$L_{constraint}$	87.50	81.40	100.00	89.75	99.25
Combo-4	L_{normal}	L_{BK}	$L_{constraint}$	94.06	92.65	100.00	96.19	98.10
Combo-5	L_{BN}	L_{kick}	$L_{constraint}$	94.44	91.74	100.00	95.69	99.08
Combo-6	L_{normal}	L_{kick}	–	90.83	86.36	100.00	92.68	98.93
$L_{knowledge}$	L_{normal}	L_{kick}	$L_{constraint}$	95.28	92.98	100.00	96.36	98.34

Table 14
Performance changes of Bi-LSTM using different combinations of loss terms compared with that using L_{BCE}.

Loss functions	Loss term for normal sample	Loss term for kick sample	Constraint term	ΔAccuracy, %	ΔRecall, %	ΔPrecision, %	ΔF1-Score, %	ΔAUC, %
Combo-1	L_{normal}	L_{BK}	–	+1.94	+2.90	+0.00	+1.67	+8.01
Combo-2	L_{BN}	L_{kick}	–	+0.00	+0.00	+0.00	+0.00	+7.45
Combo-3	L_{BN}	L_{BK}	$L_{constraint}$	−2.50	−3.72	+0.00	−2.21	+8.03
Combo-4	L_{normal}	L_{BK}	$L_{constraint}$	+4.06	+7.53	+0.00	+4.23	+6.88
Combo-5	L_{BN}	L_{kick}	$L_{constraint}$	+4.44	+6.62	+0.00	+3.73	+7.86
Combo-6	L_{normal}	L_{kick}	–	+0.83	+1.24	+0.00	+0.72	+7.71
$L_{knowledge}$	L_{normal}	L_{kick}	$L_{constraint}$	+5.28	+7.86	+0.00	+4.40	+7.12

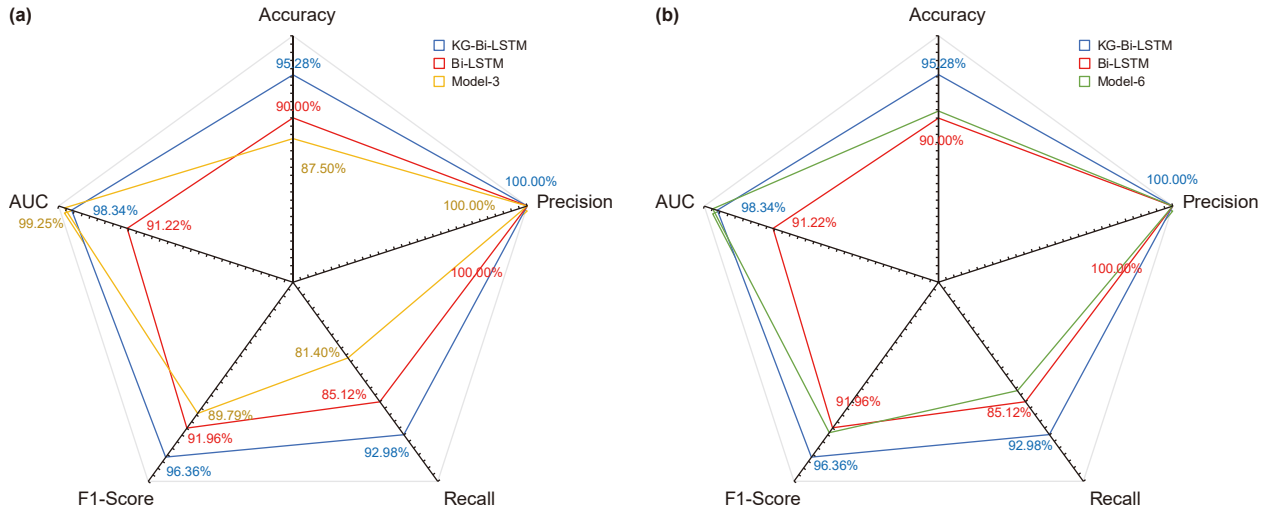

Fig. 15. Comparison of the recognition performance of the Model-3, Model-6, KG-Bi-LSTM and Bi-LSTM. (**a**) Comparison of the recognition performance of the Model-3, KG-Bi-LSTM and Bi-LSTM. (**b**) Comparison of the recognition performance of the Model-6, KG-Bi-LSTM and Bi-LSTM.

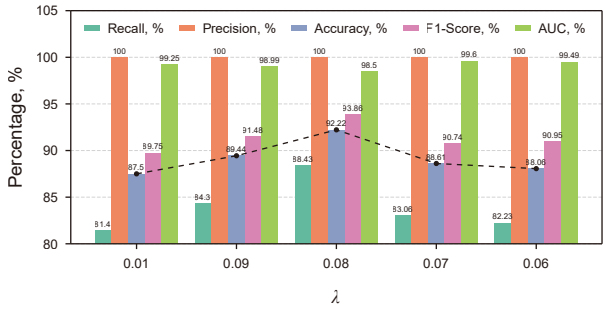

Fig. 16. Recognition performance of Bi-LSTM using Combo-3 as the loss function as λ decreases.

Furthermore, from the aforementioned performance comparisons, it can be observed that although KG-Bi-LSTM incorporates a loss weight function for normal samples, its recognition accuracy for normal samples does not degrade. The reason lies in that the adoption of the square of the error as the weight scaling factor ensures that weight decay is not significant when the error is large. During model training, the reduction in prediction error indicates that the model has gradually learned the features of normal samples. At this stage, the loss values corresponding to normal samples are significantly decreased to shift the model's focus toward samples with higher loss values, while avoiding overfitting to normal samples.

In summary, ablation experiment results demonstrate that the dual guidance mechanism of L_{normal} and L_{kick} effectively addresses the low recognition accuracy of early kick detection models, which is caused by the insufficient attention to early kick samples. $L_{\text{constraint}}$ constrains model overfitting by restricting the excessive growth of network weights. These three components work synergistically to ensure that the model using $L_{\text{knowledge}}$ can accurately identify early kicks.

4. Conclusions

In order to improve the accuracy of early kick recognition, a knowledge-guided bidirectional long short-term memory (KG-Bi-LSTM) model is proposed, integrating a knowledge-guided attention allocation module, a Bi-LSTM-based recognition module, and a knowledge-guided network weight optimization module. The model leverages prior knowledge regarding the varying importance of different monitoring parameters and that among distinct training samples to guide the Bi-LSTM's learning process. Experimental results using 4,599 field samples confirm that the proposed method outperforms the conventional Bi-LSTM, with a 5.28% increase in recognition accuracy and an earlier alarm time of nearly 2 min. The key conclusions drawn from the experiment and analysis are as follows:

(1) The variation trends of monitoring parameters as a kick develops are of different importance. The difficulty and importance of recognizing the abnormal monitoring parameter variation trends after a kick occurs are significantly greater than that during normal drilling. In addition, the importance of abnormal monitoring parameter variation trends corresponding to early kicks is the strongest.

(2) The results of the attention allocation module on the test set reveal that intelligent models pay greater attention to the variation trends of mud parameters (e.g., FOP, SUM) when monitoring early kicks. Their attention to engineering parameters (e.g., ROP and SPP) is significantly lower than that to mud parameters. This indicates that intelligent models primarily rely on the abnormal variation trends of mud parameters to determine the occurrence of kicks, with engineering parameters acting as auxiliary. Therefore, the role of engineering parameters in early kick monitoring is to help the models determine whether the abnormal variation trends of mud parameters are caused by early kicks or not.

(3) In the application scenario of sample class imbalance, the reduction of loss value according to the recognition difficulty of normal samples can effectively balance the model's attention to the two classes of samples. Besides, the enhancement of the loss value for samples at the early stage of a kick can effectively improve the model's accuracy.

(4) It is important to add constraints to the training process of knowledge-guided neural networks. The purpose of knowledge guidance is to improve the model's ability to recognize samples corresponding to the early stage of kicks. The constraints on the guidance process serve to prevent the model from overfitting. Knowledge guidance and constraints collaborate with each other to help the model find a balance between underfitting and overfitting, and thus learn

generalized features of early kicks under the limited number of samples.

However, limited by the small number of available kick cases, many intelligent models cannot fully realize their potential. Therefore, addressing the issue of sample scarcity is one of the key priorities in future research. For instance, transfer learning can effectively reduce the data demand by pre-training intelligent models. Furthermore, data augmentation techniques such as temporal generative adversarial networks (TimeGAN) can be employed to expand the number of kick samples. Data augmentation methods can also mitigate the model's overbias toward majority class samples to a certain extent. However, achieving class balance in training samples does not mean that knowledge guidance is unnecessary for model training, as these two aspects are not contradictory. Even if the numbers of normal and kick samples are balanced, the difference in classification difficulty among samples still persists. Therefore, future research should focus on integrating data augmentation methods with knowledge guidance, leveraging data augmentation to provide more samples for knowledge-data dual-driven models such as KG-Bi-LSTM.

Additionally, with the development of downhole data acquisition equipment, downhole engineering parameters such as bottomhole pressure are gradually incorporated. These parameters are more sensitive and responsive to early kicks, the samples corresponding to these parameters should be assigned greater importance. Meanwhile, the design of the loss function should take into account the variation characteristics of downhole engineering parameters, ensuring that the model can capture their abnormal variation trends to achieve timely and accurate alarms for early kicks.

CRediT authorship contribution statement

Dong-Yue Wang: Writing – original draft, Validation, Methodology, Investigation. **Wei-Feng Sun:** Supervision, Resources, Project administration, Funding acquisition, Conceptualization. **Yan-Liang Guo:** Software, Data curation. **Chen-Tao Gong:** Visualization, Formal analysis. **Wei-Min Huang:** Writing – review & editing.

Data availability

The authors do not have permission to share data.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 42472378).

References

Chen, G., Wang, X., Ji, P., Zhong, J., Zhang, J., Sun, X., Sun, B., Wang, Z., 2024. A kick monitoring method for deepwater open-circuit drilling based on convolutional neural network. Geoenergy Sci. Eng. 234, 212656. https://doi.org/10.1016/j.geoen.2024.212656.

Chen, Y., Huang, D., Zhang, D., Zeng, J., Wang, N., Zhang, H., Yan, J., 2021. Theory-guided hard constraint projection (HCP): A knowledge-based data-driven scientific machine learning method. J. Comput. Phys. 445, 110624. https://doi.org/10.1016/j.jcp.2021.110624.

Cinelli, L.P., de Oliveira, J.F., de Pinho, V.M., Passos, W.L., Padilla, R., Braz, P.F., Galves, B., Dalvi, D.P., Lewenfus, G., Ferreira, J.O., et al., 2021. Automatic event identification and extraction from daily drilling reports using an expert system and artificial intelligence. J. Petrol. Sci. Eng. 205, 108939. https://doi.org/10.1016/j.petrol.2021.108939.

Duan, S., Song, X., Cui, Y., Xu, Z., Liu, W., Fu, J., Zhu, Z., Li, D., 2023. Intelligent kick warning based on drilling activity classification. Geoenergy Sci. Eng. 222, 211408. https://doi.org/10.1016/j.geoen.2022.211408.

Elmgerbi, A., Thonhauser, G., 2022. Holistic autonomous model for early detection of downhole drilling problems in real-time. Process Saf. Environ. Prot. 164, 418–434. https://doi.org/10.1016/j.psep.2022.06.035.

Fang, X., Song, Q., Wang, X., Li, Z., Ma, H., Liu, Z., 2025. An intelligent tool wear monitoring model based on knowledge-data-driven physical-informed neural network for digital twin milling. Mech. Syst. Signal Process. 232, 112736. https://doi.org/10.1016/j.ymssp.2025.112736.

Feng, K., Liu, S., Yin, Z., Xu, Y.l., Ren, M., Tian, D., Yin, B., Sun, B., 2025. Gas kick and lost circulation risk identification method with multi-parameters based on support vector machine for drilling in deep or ultradeep waters. Eng. Sci. Technol. Int. J. 64, 102007. https://doi.org/10.1016/j.jestch.2025.102007.

Fjetland, A.K., Zhou, J., Abeyrathna, D., Gravdal, J.E., 2019. Kick detection and influx size estimation during offshore drilling operations using deep learning. In: 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA). IEEE, pp. 2321–2326. https://doi.org/10.1109/ICIEA.2019.8833850.

Fu, J., Liu, W., Zheng, X., Han, X., 2023. Transfer forest: A deep forest model based on transfer learning for early drilling kick detection. Energies 16, 2100. https://doi.org/10.3390/en16052100.

Li, Y., Cao, W., Hu, W., Wu, M., 2021. Detection of downhole incidents for complex geological drilling processes using amplitude change detection and dynamic time warping. J. Process Control 102, 44–53. https://doi.org/10.1016/j.jprocont.2021.04.002.

Liang, H., Zou, J., Liang, W., 2019. An early intelligent diagnosis model for drilling overflow based on GA-BP algorithm. Clust. Comput. 22, 10649–10668. https://doi.org/10.1007/s10586-017-1152-5.

Liu, W., Fu, J., Deng, S., Huang, P., Zou, Y., Shi, Y., Cai, C., 2024. Overflow identification and early warning of managed pressure drilling based on series fusion data-driven model. Processes 12. https://doi.org/10.3390/pr12071436.

Liu, W., Fu, J., Liang, Y., Cao, M., Han, X., 2022. A well-overflow prediction algorithm based on semi-supervised learning. Energies 15, 4324. https://doi.org/10.3390/en15124324.

Liu, Z., Jiang, M., Zhang, S., Zhang, J., Liu, Y., 2023. A smart contract vulnerability detection mechanism based on deep learning and expert rules. IEEE Access 11, 77990–77999. https://doi.org/10.1109/ACCESS.2023.3298048.

Liu, Z., Song, X., Kuru, E., Li, H.A., Zhu, Z., Li, G., 2025. A new workflow of drilling anomaly detection based on prior knowledge and unsupervised learning. SPE J. 30, 5974–5997. https://doi.org/10.2118/228414-PA.

Luo, X., Zhang, D., Zhu, X., 2021. Deep learning based forecasting of photovoltaic power generation by incorporating domain knowledge. Energy 225, 120240. https://doi.org/10.1016/j.energy.2021.120240.

Meng, H., An, X., Xing, J., 2022. A data-driven bayesian network model integrating physical knowledge for prioritization of risk influencing factors. Process Saf. Environ. Prot. 160, 434–449. https://doi.org/10.1016/j.psep.2022.02.010.

Muojeke, S., Venkatesan, R., Khan, F., 2020. Supervised data-driven approach to early kick detection during drilling operation. J. Petrol. Sci. Eng. 192, 107324. https://doi.org/10.1016/j.petrol.2020.107324.

Nie, J., Jiang, J., Li, Y., Wang, H., Ercisli, S., Lv, L., 2025. Data and domain knowledge dual-driven artificial intelligence: Survey, applications, and challenges. Expert Syst. 42, e13425. https://doi.org/10.1111/exsy.13425.

Obi, C., Falola, Y., Manikonda, K., Hasan, A., Hassan, I., Rahman, M., 2023. A machine learning approach for gas kick identification. SPE Drill. Complet. 38, 663–681. https://doi.org/10.2118/215831-PA.

Osarogiagbon, A., Muojeke, S., Venkatesan, R., Khan, F., Gillard, P., 2020. A new methodology for kick detection during petroleum drilling using long short-term memory recurrent neural network. Process Saf. Environ. Prot. 142, 126–137. https://doi.org/10.1016/j.psep.2020.05.046.

Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045.

Sun, W., Li, W., Zhang, D., Liu, K., Wang, C., Dai, Y., Huang, W., 2023. Lost circulation monitoring using bi-directional lstm and data augmentation. Geoenergy Sci. Eng. 225, 211660. https://doi.org/10.1016/j.geoen.2023.211660.

Tang, H., Zhang, S., Zhang, F., Venugopal, S., 2019. Time series data analysis for automatic flow influx detection during drilling. J. Petrol. Sci. Eng. 172, 1103–1111. https://doi.org/10.1016/j.petrol.2018.09.018.

Von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., et al., 2021. Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. IEEE Trans. Knowl. Data Eng. 35, 614–633. https://doi.org/10.1109/TKDE.2021.3079836.

Wang, N., Zhang, D., Chang, H., Li, H., 2020. Deep learning of subsurface flow via theory-guided neural network. J. Hydrol. 584, 124700. https://doi.org/10.1016/j.jhydrol.2020.124700.

Wang, X., Wu, P., Chen, Y., Zhang, E., Ye, X., Huang, Q., Peng, C., Fu, J., 2024. An intelligent kick detection model for large-hole ultra-deep Wells in the Sichuan Basin. Processes 12, 2589. https://doi.org/10.3390/pr12112589.

Wang, Z., Chen, G., Zhang, R., Zhou, W., Hu, Y., Zhao, X., Wang, P., 2023. Early monitoring of gas kick in deepwater drilling based on ensemble learning method: A case study at South China Sea. Process Saf. Environ. Prot. 169, 504–514. https://doi.org/10.1016/j.psep.2022.11.024.

Wu, L., Wang, X., Zhang, Z., Zhu, G., Zhang, Q., Dong, P., Wang, J., Zhu, Z., 2024. Intelligent monitoring model for lost circulation based on unsupervised time series autoencoder. Processes 12, 1297. https://doi.org/10.3390/pr12071297.

Wu, S., Zhang, L., Fan, J., Zheng, W., Zhou, Y., 2019. Real-time risk analysis method for diagnosis and warning of offshore downhole drilling incident. J. Loss Prev. Process. Ind. 62, 103933. https://doi.org/10.1016/j.jlp.2019.103933.

Xu, Y.Q., Liu, K., He, B.L., Pinyaeva, T., Li, B.S., Wang, Y.C., Nie, J.J., Yang, L., Li, F.X., 2023. Risk pre-assessment method for regional drilling engineering based on deep learning and multi-source data. Pet. Sci. 20, 3654–3672. https://doi.org/10.1016/j.petsci.2023.06.005.

Yin, Q., Yang, J., Tyagi, M., Zhou, X., Hou, X., Cao, B., 2021. Field data analysis and risk assessment of gas kick during industrial deepwater drilling process based on supervised learning algorithm. Process Saf. Environ. Prot. 146, 312–328. https://doi.org/10.1016/j.psep.2020.08.012.

Yin, Q., Yang, J., Tyagi, M., Zhou, X., Wang, N., Tong, G., Xie, R., Liu, H., Cao, B., 2022. Downhole quantitative evaluation of gas kick during deepwater drilling with deep learning using pilot-scale rig data. J. Petrol. Sci. Eng. 208, 109136. https://doi.org/10.1016/j.petrol.2021.109136.

Yin, Q., Zhu, Q., Song, Z., Guo, Y., Yang, J., Xu, Z., Chen, K., Sun, L., Tyagi, M., 2025. Deep learning based early warning methodology for gas kick of deepwater drilling using pilot-scale rig data. Process Saf. Environ. Prot., 106844 https://doi.org/10.1016/j.psep.2025.106844.

Yin, T., Lu, N., Guo, G., Lei, Y., Wang, S., Guan, X., 2023. Knowledge and data dual-driven transfer network for industrial robot fault diagnosis. Mech. Syst. Signal Process. 182, 109597. https://doi.org/10.1016/j.ymssp.2022.109597.

Zhang, D., Sun, W., Dai, Y., Bu, S., Feng, J., Huang, W., 2024a. Intelligent kick detection using a parameter adaptive neural network. Geoenergy Sci. Eng. 234, 212694. https://doi.org/10.1016/j.geoen.2024.2126944.

Zhang, D., Sun, W., Dai, Y., Liu, K., Li, W., Wang, C., 2023a. A hierarchical early kick detection method using a cascaded gru network. Geoenergy Sci. Eng. 222, 211390. https://doi.org/10.1016/j.geoen.2022.211390.

Zhang, G., Li, J., Yang, H., Huang, H., 2024b. A new detection method of co-existence of well kick and loss circulation based on thermal behavior of wellbore-formation coupled system. Int. J. Hydrogen Energy 58, 239–258. https://doi.org/10.1016/j.ijhydene.2024.01.167.

Zhang, G., Yang, H.W., Li, J., Zhang, H., Huang, H.L., Wang, B., Wang, W.X., Chen, H., 2025. Numerical simulation of gas kick evolution and wellbore pressure response characteristics during the deepwater dual gradient drilling. Pet. Sci. 22, 398–412. https://doi.org/10.1016/j.petsci.2024.12.010.

Zhang, Z., Lai, X., Du, S., Yu, W., Wu, M., 2023b. Early warning of loss and kick for drilling process based on sparse autoencoder with multivariate time series. IEEE Trans. Ind. Inf. 19, 11019–11029. https://doi.org/10.1109/TII.2023.3242772.

Zhou, D., Zhu, Z., Zhou, C., Song, X., Li, G., Zhang, C., Li, Q., Li, S., 2025. Unsupervised learning–based kick risk warning with dynamic thresholds and knowledge embedding. SPE J. 30, 6584–6602. https://doi.org/10.2118/230293-PA.

Zhu, Z., Zhou, D., Yang, D., Song, X., Zhou, M., Zhang, C., Duan, S., Zhu, L., 2023. Early gas kick warning based on temporal autoencoder. Energies 16, 4606. https://doi.org/10.3390/en16124606.