Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

A framework for generating anomaly analysis comments in DHI interpretation report

Xiao Han[a], Meng Gao[a,*], Weizheng Shen[a,*], Huihuan Liu[b], Baisheng Dai[a], Yongqiang He[c], Huixin Liu[a]

[a] College of Electrical and Information, Northeast Agricultural University, Harbin 150030, China
[b] Dairy Association of Heilongjiang Province, Harbin 150030, China
[c] Inner Mongolia Mengniu Dairy (Group) CO.,LTD, Hohhot 011517, China

ARTICLE INFO

Keywords:
DHI interpretation reports
Time-series data of DHI indicators
Anomaly analysis
Data-to-text
End-to-end

ABSTRACT

Efficient and high-quality writing of DHI interpretation reports is critical to realize the instructive value of time-series DHI data, in which anomaly analysis of DHI indicators is the basis of diagnosing problems and giving suggestions for dairy farms. However, this work is labour intensive and knowledge-driven, which is easy to cause inadequate even incorrect analysis due to the lack of DHI specialists. Aiming to the subtasks of automatically generating comments of static anomaly and dynamic change trend of DHI indicators, we proposed a framework to tackle them separately. Firstly, we employed a series of anomaly degree matrixes to deal with the first subtask. Secondly, we proposed an improved end-to-end data-to-text model which incorporates with three pre-processing methods, attention mechanism and contrastive penalty to deal with the second subtask. Experimental results show that our proposed model outperforms the baselines and related works both in terms of fluency and correctness of generated comments, which is suitable for the preliminary description of DHI indicators.

1. Introduction

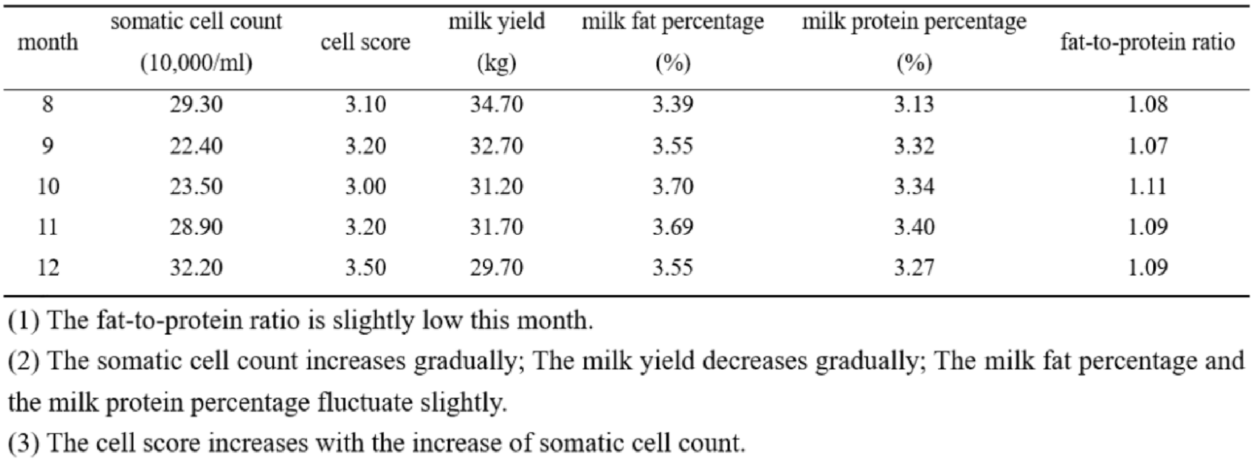
Dairy Herd Improvement (DHI) has been proved to be an efficient method for guiding the feeding, breeding and management of dairy farms, thus improving the performance of dairy herd around the world. It uses mathematic models to calculate the numerical values of a series of indicators according to the milk yield, milk components and other basic information of dairy herd, and DHI interpreters use their scientific knowledge to write DHI interpretation report for dairy farm. The report mainly consists of three parts: (1) analysis of abnormal indicators, (2) possible causes and (3) guidance suggestions for dairy farm, in which anomaly analysis is the premise of diagnosing the causes of anomalies and giving suggestions. However, writing personalized anomaly analysis comments of abnormal indicators for end users is labour intensive and requires solid professional knowledge, which is easy to cause inadequate interpretation due to incomplete analysis. Therefore, the task of generating anomaly analysis comments is of great significance to improve the efficiency and quality of writing DHI interpretation reports, and the generated comments are also valuable for automatic diagnosis of possible problems incorporated with knowledge graph and dialogue system in the future.

We illustrate the three characteristic problems of generating anomaly analysis comments from time-series data of DHI indicators in Fig. 1. The first problem is that whether an indicator is normal at present needs to be clear. For example, the comment describes that *the fat-to-protein ratio is slightly low this month* in this figure. The second problem is that an interpreter also needs to consider the recent trends of key indicators. For example, in Fig. 1, we can see that *the somatic cell count increases gradually, the milk yield decreases gradually*. The third problem is that anomaly analysis comments often mention the joint change trends of multiple indicators furtherly. For example, the comment in this figure states that *the cell score increases with the increase of the somatic cell count.*

To address these issues, we proposed a method for automatically generating anomaly analysis comments of DHI interpretation report from time-series numerical values of multiple DHI indicators. To tackle the first problem, we used anomaly degree matrixes of indicators to assist in detecting anomalies and generating corresponding comments. For the second problem, we employed two kinds of pre-processing methods to capture global and local trends of indicators. To address the third problem, we used a bidirectional recurrent neural network (Bi-

[*] Corresponding authors.
E-mail addresses: gaomeng@neau.edu.cn (M. Gao), wzshen@neau.edu.cn (W. Shen).

https://doi.org/10.1016/j.compag.2023.108331
Received 29 July 2023; Received in revised form 27 September 2023; Accepted 13 October 2023
Available online 25 October 2023



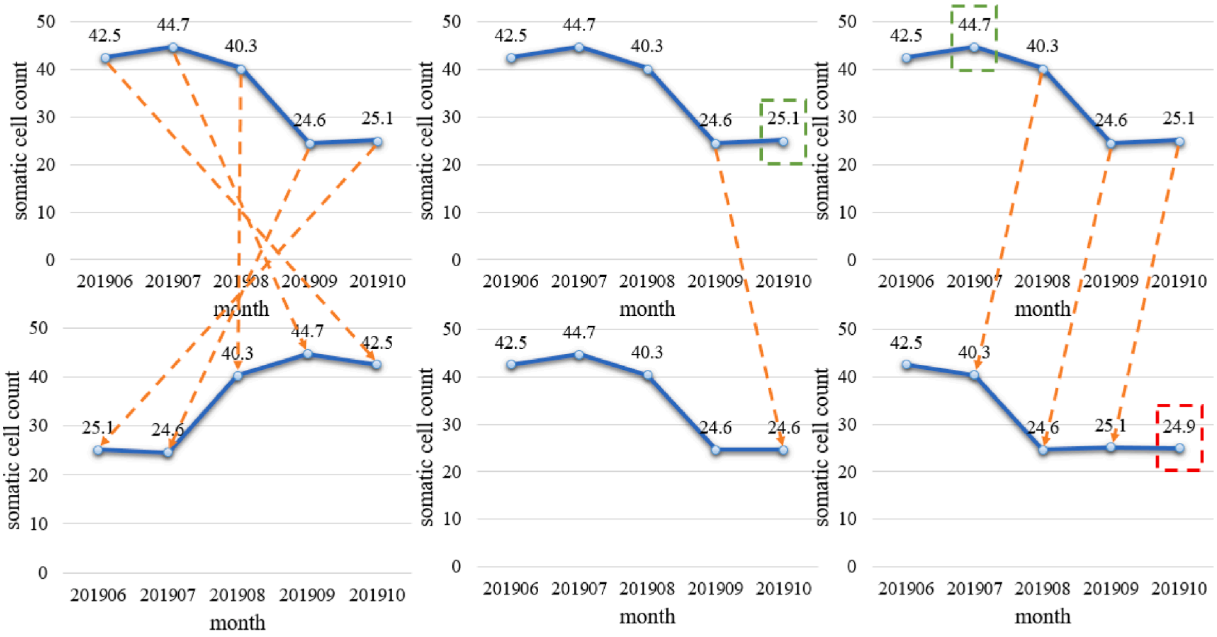
month	somatic cell count (10,000/ml)	cell score	milk yield (kg)	milk fat percentage (%)	milk protein percentage (%)	fat-to-protein ratio
8	29.30	3.10	34.70	3.39	3.13	1.08
9	22.40	3.20	32.70	3.55	3.32	1.07
10	23.50	3.00	31.20	3.70	3.34	1.11
11	28.90	3.20	31.70	3.69	3.40	1.09
12	32.20	3.50	29.70	3.55	3.27	1.09

(1) The fat-to-protein ratio is slightly low this month.

(2) The somatic cell count increases gradually; The milk yield decreases gradually; The milk fat percentage and the milk protein percentage fluctuate slightly.

(3) The cell score increases with the increase of somatic cell count.

Fig. 1. An anomaly analysis comment written by an expert according to the performance of some indicators.



(a) flipping **(b) clipping** **(c) interpolation**

Fig. 2. Operations for data construction. Orange line represents data replacement; green box represents deleting the value of this month; red box represents data filling with the value of next month.

RNN) to capture representations of multiple indicators and input them into a long short-term memory (LSTM) combined with attention mechanism to generate proper contents. In addition, we introduced a contrastive penalty mechanism to help our model generate not only fluent sentences, but also correct sentences.

Our work is consistent with the real-word scenarios, in which specialists describe anomaly analysis comments by interpreting a sequence of sustainably recorded and interrelated numerical values of multiple DHI indicators. The proposed method was evaluated on the task of generating Chinese anomaly analysis comments of DHI interpretation report. Automatic evaluation results indicate that our model improves the fluency and correctness of generated comments compared with baselines and related works.

2. Related work

In this paper, we focus on the task of generating anomaly analysis comments of DHI interpretation report from numerical data, which belongs to the field of data-to-text generation. The task of generating descriptions from time-series or structured data has been tackled in many fields such as sports (Liang et al., 2009; Wiseman et al., 2017), weather forecasts (Cascallar-Fuentes et al., 2022; Murakami et al., 2021), health monitoring (Hunter et al., 2012; Banaee et al., 2013). The traditional pipeline-based approaches (Konstas and Lapata, 2013; Banaee and Loutfi, 2015; Riza et al., 2019) divided the task into several subtasks and solve them separately, which can generate fluent and accurate texts. However, they require hand-crafted rules which are heavily relied on domain experts, in addition, the text generation system needs to be rebuilt when a new scenario comes up.

More recently, end-to-end approach based on deep learning, primarily using an encoder-decoder model, has been proved to be able to generate descriptions from time-series or structured data by solving the subtasks jointly in a single framework. Murakami et al. (2017) introduced a time embedding vector into an encoder-decoder model to generate the description about delivery time of the Nikkei 225 market comments, and realized the arithmetic operations of numerical values

Table 1
The anomaly degree matrix of somatic cell count.

	somatic cell count (10,000/ml)	degree of abnormality
1	0.00	normal
2	30.00	slightly high
3	50.00	relatively high
4	150.00	too high

Table 2
The anomaly degree matrix of fat-to-protein ratio.

	fat-to-protein ratio	degree of abnormality
1	0.00	excessive low
2	0.80	generally low
3	1.00	slightly low
4	1.12	normal
5	1.30	slightly high
6	1.40	relatively high
7	1.60	too high

by modifying the copy mechanism, however, they only tackled a single stock price indicator. Aoki et al., 2018 proposed an improved encoder-decoder model taking into account external resources besides the Nikkei 225 so as to enrich the content of the generated market comments. Although multiple indicators were employed in this study, the performance of the model did not improve significantly. Gong et al. (2019) used a hierarchical encoder to generate NBA event report on the ROTOWIRE dataset, they combined representations from row, column and time dimension into one dense vector to extend the table representation, which was effective in generating more enriched descriptions. Studies above were able to generate relatively fluent descriptions from time-series or structured data, but they did not concern the rationality of the generated descriptions, in other words, sometimes they may generate fluent but incorrect sentences or terms. To deal with this problem, Uehara et al. (2020) exploited contrastive examples (Noji and Takamura, 2020; Welleck et al., 2019; Huang et al., 2018) to improve the accuracy of the keywords in the generated market comments, which was capable to generate sentences with better lexical choices, without degrading the fluency. However, the work was powerless in describing the joint change trends of multiple indicators. Yan et al. (2021) proposed a weakly supervised contrastive loss (WCL) for radiology report generation using chest X-ray images, which was helpful to generate semantically-close and relatively correct reports. These studies prove that the proper use of negative samples is beneficial for generating correct contents, but the performances of these models still need to be improved.

3. Materials and methods

3.1. Dataset

DHI specialists obtain the numerical values of multiple indicators and their historical values, and interpret them together to come up with the anomaly analysis comments. To reproduce this, we used two types of DHI data: DHI reports and annotated comments.

DHI reports. A DHI report, which is usually made by a professional software (called CNDHI in China), is composed of a series of tables, among which several tables are critical for management of dairy farms and each of them includes a sequence of numerical values of multiple indicators. In this study, we used DHI reports of 8 dairy farms provided by the Dairy Association of Heilongjiang Province (DAHLJ), which is one of the partners of our research group. These reports are dated from January 2019 to December 2022, each dairy farm receives a DHI report at each month, therefore, our dataset includes 384 real-world DHI reports. Since specialists usually focus on the change in recent months of an indicator, we took 5 consecutive months as a unit and divided our data into 336 segments. In order to enhance the robustness and generalization ability of our model, we extended our raw dataset by 3 kinds of operations: flipping, clipping and interpolation, where flipping represents converting data into its reverse order, as shown in Fig. 2(a), the raw data of somatic cell count from June 2019 to October 2019 was [42.5, 44.7, 40.3, 24.6, 25.1] and the flipped data should be [25.1, 24.6, 40.3, 44.7, 42.5]; clipping represents deleting the value of latest month and then filling it with the value of previous month, as shown in Fig. 2 (b), the value of somatic cell count on October 2019 was 25.1, we deleted the latest value and used 24.6 on September 2019 to fill it; interpolation represents randomly selecting a value of one month in addition to the initial month, deleting the value and then moving the values of subsequent months forward in sequence, as shown in Fig. 2(c), the value of somatic cell count on July 2019 (namely 44.7) was deleted and we used the values on August to November 2019 (namely 40.3, 24.6, 25.1 and 24.9) to sequentially fill the values on July to October 2019.

We finally obtained 3024 constructed records through 9 types of operations: raw-flipping, raw-clipping, raw-interpolation, raw-clipping-interpolation, raw-interpolation-clipping, raw-flipping-clipping, raw-flipping-interpolation, raw-flipping-clipping-interpolation and raw-flipping-interpolation-clipping, where each operation conducted its sub-operations in order, for example, raw-flipping means flipping the raw data, raw-clipping-interpolation means clipping the raw data first and then interpolating the clipped data.

Annotated comments. Actually, each DHI report corresponds to a DHI interpretation report which includes the anomaly analysis comments, however, these comments were written in different time-scales. Therefore, we invited 2 DHI specialists of Dairy Association of Heilongjiang Province (DAHLJ) to reorganize the comments of the 336 raw records, and we wrote new comments for the other 3024 constructed records.

3.2. Anomaly degree matrix

In the real-world scenarios, a DHI interpreter needs to clarify whether the value of an indicator is normal at present firstly. We defined this subtask as the static anomaly detection in this paper. For example, the normal value of fat-to-protein ratio is usually between 1.12 and 1.30, if its value of the latest month is outside of this normal range, then we consider that it is in the state of static anomaly. In addition, the anomaly analysis comments not only describe the abnormal state of an indicator, but also present its abnormal degree, such as *the fat-to-protein ratio is relatively high this month.*

We used an anomaly degree matrix to jointly realize the static anomaly detection and description of an indicator. For an indicator $q_i(0 \le i \le n)$, we set its two-dimensional anomaly degree matrix $p^{(i)}$ which has m kinds of anomaly types according to domain knowledge, among which $p^{(i)}_{k2}(0 \le k \le m-1)$ is the k th anomaly type of q_i, $p^{(i)}_{k1}$ is the lower threshold of $p^{(i)}_{k2}$, we illustrate it in Table1 and Table2 using somatic cell count and fat-to-protein ratio as examples. Given a value of q_i denoted as v_i, we located its anomaly type in $p^{(i)}$ by comparing v_i with the threshold $p^{(i)}_{k1}(0 \le k \le m-1)$ from high to low until we found a threshold $p^{(i)}_{t1}$ which was smaller than v_i, then we regarded $p^{(i)}_{t2}$ as the anomaly type of v_i, and the description could be correctly generated by inserting the indicator name and its anomaly type into a rule-based template, namely "The { } is { } this month". For example, the latest value of fat-to-protein ratio is 1.09, we compared 1.09 with the thresholds in Table 2 from high to low until we found 1.00 which is smaller than 1.09, and then we generated a sentence "*The fat-to-protein ratio is slightly low this month*" according to the template.

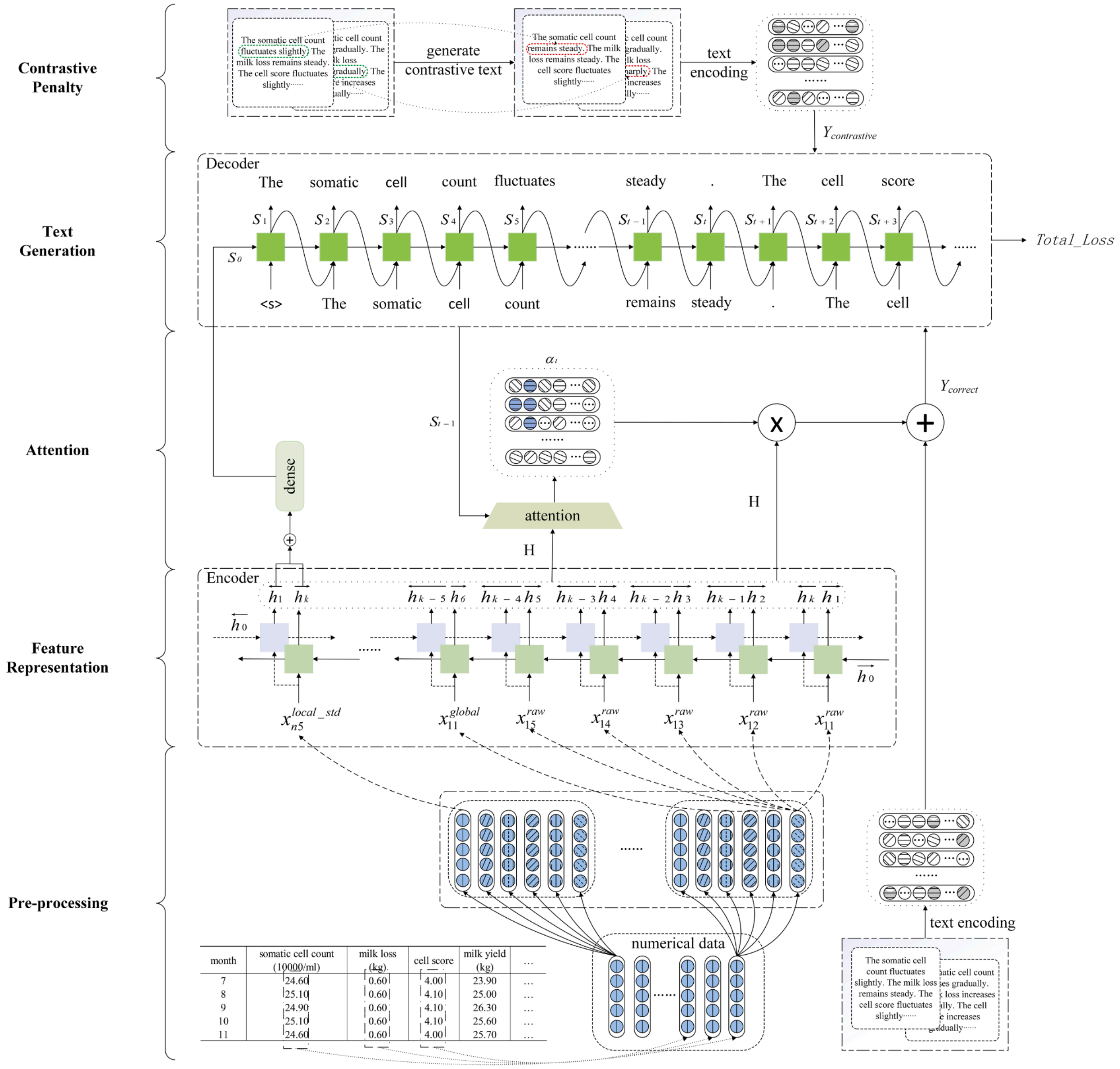

Contrastive Penalty
Text Generation
Attention
Feature Representation
Pre-processing
The somatic cell count fluctuates slightly. The milk loss remains steady. The cell score fluctuates slightly······
generate contrastive text
The somatic cell count remains steady. The milk loss remains steady. The cell score fluctuates slightly······
text encoding
$Y_{contrastive}$
Decoder
The somatic cell count fluctuates steady . The cell score
S_1 S_2 S_3 S_4 S_5 S_{t-1} S_t S_{t+1} S_{t+2} S_{t+3}
S_0
<s> The somatic cell count remains steady . The cell
Total_Loss
α_t
S_{t-1}
$Y_{correct}$
dense
attention
H
H
Encoder
h_1 h_k h_{k-5} h_6 h_{k-4} h_5 h_{k-3} h_4 h_{k-2} h_3 h_{k-1} h_2 h_k h_1
$\overleftarrow{h}_0$
$\overrightarrow{h}_0$
$x_{n5}^{local\_std}$ x_{11}^{global} x_{15}^{raw} x_{14}^{raw} x_{13}^{raw} x_{12}^{raw} x_{11}^{raw}
numerical data
text encoding
The somatic cell count fluctuates slightly. The milk loss remains steady. The cell score fluctuates slightly······
month | somatic cell count (10000/ml) | milk loss (kg) | cell score | milk yield (kg) | …
7 | 24.60 | 0.60 | 4.00 | 23.90 | …
8 | 25.10 | 0.60 | 4.10 | 25.00 | …
9 | 24.90 | 0.60 | 4.10 | 26.30 | …
10 | 25.10 | 0.60 | 4.10 | 25.60 | …
11 | 24.60 | 0.60 | 4.00 | 25.70 | …

Fig. 3. The overall architecture of our model.

3.3. Encoder-Decoder model

After clarify the static anomaly of an indicator, a DHI interpreter needs to further compare its value of latest month with historical values. For example, the milk yield is 25.30 kg last month and 25.75 kg this month, we consider the milk yield increases this month. However, the descriptions of change trends of an indicator are more complex in actually, such as *the milk yield increases sharply* or *the milk yield decreases gradually*. Therefore, it is necessary to detect the changes within a period of time rather than two adjacent months, and also present the change degree. Moreover, in addition to describing the change trend of a single indicator, the joint change trends of multiple indicators should also be described, such as *the milk fat percentage increases with the increase of lactation days*. We defined this subtask as the dynamic trend detection and text generation from multi-dimensional time series data in this paper, and we proposed an improved end-to-end comment generation model for DHI indicators (DHICG) from encoder-decoder to jointly tackle this subtask.

Fig. 3 shows an overview of our proposed model. First (section 3.3.1), we used three pre-processing methods to capture the global and local changes of indicators as well as standardize them. Second (Section 3.3.2), we used several encoding methods for time-series data to obtain feature representations of the pre-processed values. Third (Section 3.3.3 and Section 3.3.4), we employed a Long Short-term Memory (LSTM) network as decoder to generate descriptions of change trends incorporated with attention mechanism which made the decoder to select more suitable contents to generate comments. Finally (section 3.3.5), we extended the decoder by introducing a contrastive penalty mechanism, so as to push our model to generate more correct keywords in anomaly analysis comments.

3.3.1. Pre-processing

For the time-series data of an indicator, a vector consists of the values for a period of time and has N elements, we denote it as $\boldsymbol{x} = (x_0, x_1, \cdots, x_{N-1})$. Since the standard encoder-decoder model is unable to operate numerical comparisons, we used three kinds of pre-processing methods: global moving reference, local moving reference and standardization, thus recognizing the data pattern as well as removing noises and enhancing the generalizability of our model.

Global moving reference. The method substituted each element

Table 3
Rules for generating contrastive examples.

	Original keywords	Keywords used for substitution
1	increase gradually	{ decrease gradually }
2	decrease gradually	{ increase gradually }
3	increase sharply	{ increase gradually; decrease gradually; decrease sharply }
4	decrease sharply	{ increase gradually; increase sharply; decrease gradually }
5	remain steady	{ increase gradually; increase sharply; decrease gradually; decrease sharply }
6	fluctuate slightly	{ remain steady; increase gradually; increase sharply; decrease gradually; decrease sharply }
7	increase	{ decrease }
8	decrease	{ increase }

$x_i(0 \le i \le N-1)$ of input $\boldsymbol{x}$ by

$$x_i^{global} = x_i - x_0 \tag{1}$$

which was introduced to capture the long-term fluctuation from the initial month.

Local moving reference. The method substituted each element $x_i(1 \le i \le N-1)$ of input $\boldsymbol{x}$ by

$$x_{\mathrm{i}}^{local} = x_i - x_{i-1} \tag{2}$$

which was introduced to capture the short-term fluctuation from the previous month.

Standardization. We substituted each element $x_i(0 \le i \le N-1)$ of input $\boldsymbol{x}$ by

$$x_i^{std} = \frac{x_i - \mu}{\sigma} \tag{3}$$

where μ and σ are the mean and standard deviation of the values in the training data, respectively. The standardized values were less affected by data dimensions.

We denote the original data of indicator q_i as $\boldsymbol{x}_i^{raw}$, by applying the global and local moving reference methods to $\boldsymbol{x}_i^{raw}$, we obtained two vectors of pre-processed values $\boldsymbol{x}_i^{global}$ and $\boldsymbol{x}_{\mathrm{i}}^{local}$, and by applying the standardization method to $\boldsymbol{x}_i^{raw}$, $\boldsymbol{x}_i^{global}$ and $\boldsymbol{x}_{\mathrm{i}}^{local}$, we obtained other three vectors $\boldsymbol{x}_i^{raw\_std}$, $\boldsymbol{x}_i^{global\_std}$ and $\boldsymbol{x}_i^{local\_std}$. These vectors were concatenated into one vector which was denoted as $\boldsymbol{X} = (\boldsymbol{x}_1^{raw}, \boldsymbol{x}_1^{global}, \boldsymbol{x}_1^{local}, \boldsymbol{x}_1^{raw\_std}, \boldsymbol{x}_1^{global\_std}, \boldsymbol{x}_1^{local\_std}, \cdots, \boldsymbol{x}_n^{raw}, \boldsymbol{x}_n^{global}, \cdots, \boldsymbol{x}_n^{global\_std}, \boldsymbol{x}_n^{local\_std})$, and we fed it into our neural network model.

3.3.2. Feature representation

We used a single layer bi-directional gate recurrent unit (BiGRU) (Chung et al., 2014) network to encode our pre-processed time-series data. Each element of $\boldsymbol{X}$ was fed into the GRU network as input for a time step. The hidden state H_τ at step τ is:

$$H_\tau = \left[\overrightarrow{h_\tau}; \leftarrow h_{k-\tau+1}\right] \tag{4}$$

$$h_\tau = Z_\tau \odot h_{\tau-1} + (1 - Z_\tau) \odot \widetilde{h_\tau} \tag{5}$$

$$\widetilde{h_\tau} = \tanh(X_\tau W_{xh} + (R_\tau \odot h_{\tau-1}) W_{hh} + b_h) \tag{6}$$

where $\widetilde{h_\tau}$ is the one-way candidate hidden state. h_τ and $h_{\tau-1}$ are the one-way hidden states at time step τ and $\tau - 1$, respectively. X_τ is the input data. Z_τ is the update gate. W_{xh} and W_{hh} are weight matrixes. b_h is the bias. R_τ is a reset gate. $\overrightarrow{h_\tau}$ is the forward hidden state at time step τ. $\overset{\leftarrow}{h}_{k-\tau+1}$ is the backward hidden state at time step τ.

The hidden states of all the time steps and the output of the encoder, which are denoted as H and S_0, respectively, are computed as follows:

$$H = \{H_1, H_2, H_3, \cdots H_k\} \tag{7}$$

$$S_0 = tanh(dense(H_k)) \tag{8}$$

Table 4
Overview of the models compared in our experiments.

	Model	Encoder	Pre-processing			Attention	Contrastive
			global	local	standardization		
baselines	MLP-ENC	MLP	√	√	√	×	×
	CNN-ENC	CNN	√	√	√	×	×
	BiGRU-ENC	BiGRU	√	√	√	×	×
our model	DHICG	BiGRU	√	√	√	√	√
ablation models	-global	BiGRU	×	√	√	√	√
	-local	BiGRU	√	×	√	√	√
	-std	BiGRU	√	√	×	√	√
	-att	BiGRU	√	√	√	×	√
	–con	BiGRU	√	√	√	√	×
related works	Murakami et al. (2017)	MLP	√	×	√	×	×
	Aoki et al., 2018	MLP	√	×	√	×	×
	Uehara et al. (2020)	MLP	√	×	√	×	√

Table 5
Evaluating results of 7 models using BLEU as well as correctness of keywords extracted from the generated comments in accuracy (A%), precision (P%), recall (R%), and F-measure scores (F%), *tendency_accuracy* (TA%) and *consistency_accuracy* (CA%).

Model	overall	single indicator					multiple indicators		
	BLEU	A%	P%	R%	F%	TA%	A%	F%	CA%
MLP-ENC	55.46	61.25	60.14	61.70	60.45	66.35	47.13	48.21	57.27
CNN-ENC	38.26	54.11	42.77	35.14	40.99	50.78	50.00	41.55	0.00
BiGRU-ENC	58.40	62.31	60.41	59.64	60.25	70.69	47.45	48.24	67.41
Murakami et al. (2017)	32.75	53.17	41.38	33.92	36.64	52.83	50.45	41.79	0.00
Aoki et al., 2018	54.08	60.99	58.45	59.84	58.72	73.56	47.60	51.62	54.34
Uehara et al. (2020)	54.15	71.87	67.66	69.35	67.99	83.51	67.34	70.93	76.99
Our model	61.28	77.49	76.20	69.94	74.86	87.87	70.83	73.84	80.46

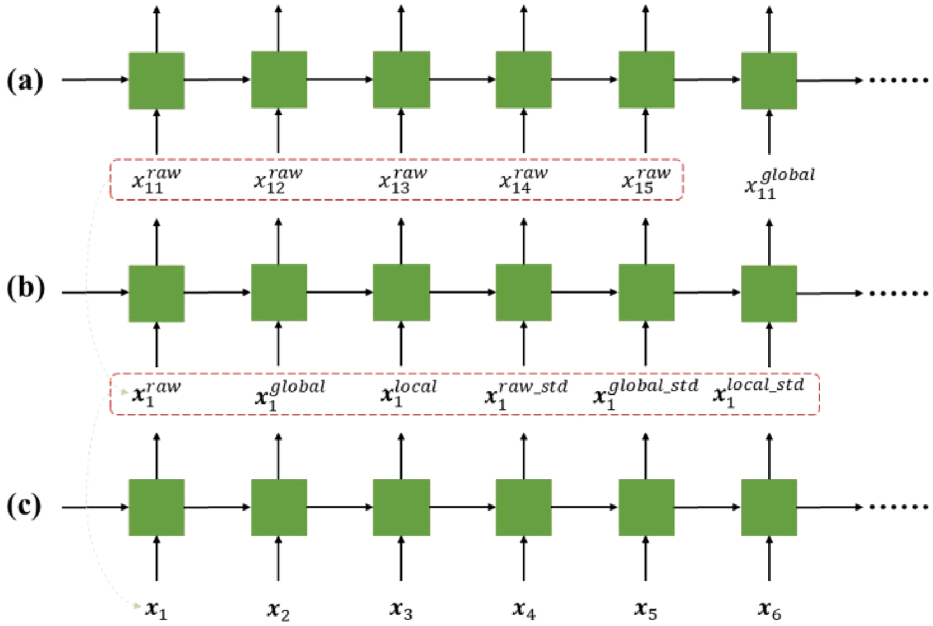

(a)
x_{11}^{raw} x_{12}^{raw} x_{13}^{raw} x_{14}^{raw} x_{15}^{raw} x_{11}^{global}
(b)
x_1^{raw} x_1^{global} x_1^{local} $x_1^{raw\_std}$ $x_1^{global\_std}$ $x_1^{local\_std}$
(c)
x_1 x_2 x_3 x_4 x_5 x_6

Fig. 4. 3 types of data format input to an encoder, where (a) takes a value $x_{ij}^{p}(j = 1, 2, \cdots, 5)$ of vector x_i^p as input at each time step sequentially, (b) takes a vector x_i^p(p ∈ {raw, global, local, raw_std, global_std, local_std}) as input at each time step sequentially, and (c) takes a vector $x_i = (x_i^{raw}, x_i^{global}, x_i^{local}, x_i^{raw_std}, x_i^{global_std}, x_i^{local_std})$as input at each time step sequentially. The green box represents a time step of the encoder.

Table 6
Comparison results of 3 models using different formats of data as input.

Model	Data format	overall	single indicator					multiple indicators		
		BLEU	A%	P%	R%	F%	TA%	A%	F%	CA%
BiGRU-ENC	(a)	58.40	62.31	60.41	59.64	60.25	70.69	47.45	48.24	67.41
BiGRU-6	(b)	54.74	56.84	50.86	51.84	51.05	60.70	46.70	47.97	63.64
BiGRU-1	(c)	36.57	27.47	28.66	37.99	30.14	38.79	11.81	0.00	0.00

Table 7
Ablated experiment results of our model.

Model	overall	single indicator					multiple indicators		
	BLEU	A%	P%	R%	F%	TA%	A%	F%	CA%
DHICG	61.28	77.49	76.20	69.94	74.86	87.87	70.83	73.84	80.46
-global	58.11	75.07	72.35	67.19	71.26	84.56	67.82	71.71	59.02
-local	58.96	74.90	71.35	66.50	70.32	85.19	64.69	68.39	77.95
-std	50.66	69.41	55.92	48.22	54.19	68.34	64.99	68.71	17.85
-att	58.41	73.40	75.11	69.43	73.90	84.11	69.93	73.64	78.62
–con	61.26	65.79	60.44	61.15	60.58	73.85	49.42	50.52	73.56

Table 8
Early rules for generating contrastive examples.

	Original keywords	Keywords used for substitution
1	increase gradually	{ decrease gradually; increase sharply; decrease sharply; remain steady }
2	decrease gradually	{ increase gradually; increase sharply; decrease sharply; remain steady }
3	increase sharply	{ increase gradually; decrease gradually; decrease sharply; remain steady }
4	decrease sharply	{ increase gradually; decrease gradually; increase sharply; remain steady }
5	remain steady	{ increase gradually; decrease gradually; increase sharply; decrease sharply; fluctuate slightly}
6	fluctuate slightly	{ increase gradually; decrease gradually; increase sharply; decrease sharply; remain steady }
7	increase	{ decrease }
8	decrease	{ increase }

where $H_k = \left[\overrightarrow{h_k}; \overleftarrow{h_1}\right]$ is the hidden state at the last time step. *tanh* is the activation function. *dense* is the linear layer.

3.3.3. Attention

In order to select more suitable contents to generate comments, we introduced the attention mechanism into our model. At time step *t* of the decoder in Section 3.3.4, the weight matrix of the input at each time step of the encoder, which is denoted as α_t, is computed by

$$\alpha_t = softmax(\widetilde{\alpha}_t) \quad (9)$$

$$\widetilde{\alpha}_t = vE_t \quad (10)$$

$$E_t = \tanh(W_s S_{t-1} + W_h H) \quad (11)$$

where H is the hidden state obtained from the feature representation layer. v, W_s and W_h are weight matrixes. S_{t-1} is the hidden state at time step $t-1$ of the decoder. In particular, S_0 is the initial hidden state of the

Table 9
Comparison results of contrastive penalty using different rules.

Model	overall	single indicator					multiple indicators		
	BLEU	A%	P%	R%	F%	TA%	A%	F%	CA%
BiGRU-ENC	58.40	62.31	60.41	59.64	60.25	70.69	47.45	48.24	67.41
-att0	57.86	62.73	60.33	60.72	60.41	75.76	47.79	50.74	69.25
-att	58.41	73.40	75.11	69.43	73.90	84.11	69.93	73.64	78.62

decoder, which the output of the encoder above.

Then, the weighted contextual semantic matrix, which is denoted as C_t, is computed by

$$C_t = \alpha_t H \tag{12}$$

we fed C_t into the decoder together with the text encoding vectors to help the decoder to select proper contents.

3.3.4. Text generation

We used a LSTM network as the decoder to generate comments. The generated feature word at time step t of the decoder is computed by

$$\widehat{y_t} = softmax(align(emb(y_t), C_t, S_t)) \tag{13}$$

$$S_t = LSTM(emb(y_t), C_t, S_{t-1}) \tag{14}$$

where y_t is the feature word of the annotated comments. $emb(y_t)$ is the encoding vector of y_t. C_t is the weighted contextual semantic matrix.S_t, S_{t-1} are the hidden states at time step t and $t-1$ of the decoder, respectively. *align* is the linear layer.

We used the negative log-likelihood loss function (Uehara et al., 2020) to calculate the loss of our model for parameter optimization, which is represented as follows:

$$Loss = \frac{1}{M}\sum_{t=1}^{M}\ln(1 + e^{-\hat{y}_t\omega^\top y_t}) \tag{15}$$

where M is the length of the generated comments. ω is the assigned weight. y_t, $\widehat{y_t}$ are the annotated and generated feature words, respectively.

3.3.5. Contrastive penalty

In anomaly analysis comments of DHI interpretation report, some words must be correct to ensure that the generated sentence is highly reliable for anomaly diagnosis, we called these words keywords. In order to improve the correctness of our model in generating keywords, we introduced a contrastive penalty mechanism (Noji and Takamura, 2020) into our model. The mechanism generated contrastive examples by replacing keywords in the annotated comments firstly, and then calculated the loss which is denoted as *Margin_Loss* between the outputs of our model training with the annotated comments and the contrastive examples, separately. We used the *Margin_Loss* together with the *Loss* to optimize the parameters of our model.

Specifically, we used pre-established keyword substitution rules, which is shown in Table3, to generate contrastive examples. We extracted an keyword w_v randomly from the input annotated comments denoted as $Y_{correct}$, and found the set of substitution keywords according to w_v from Table 3, then we substitute w_v with a keyword randomly selected from the substitution set, thus we obtained the contrastive example denoted as $Y_{contrastive}$. Note that rules 1 to 6 and rules 7 to 8 are used to generate dynamic trend comments for a single indicator and multiple indicators, respectively.

Taking $Y_{correct}$ and $Y_{contrastive}$ as the input of the decoder, respectively, and we obtained their corresponding generated comments $\widehat{Y}_{correct}$ and $\widehat{Y}_{contrastive}$, and the loss between their output probabilities is computed by

$$Margin\_Loss = mean\left(\Sigma_0^b max(0, \omega(margin - logP(\widehat{Y}_{correct}) + logP(\widehat{Y}_{contrastive})))\right) \tag{16}$$

where $b \in \{0, \cdots, batch\_size - 1\}$. ω is the assigned weight; *margin* is the pre-established threshold between the log-likelihoods of $\widehat{Y}_{correct}$ and $\widehat{Y}_{contrastive}$.

Finally, we obtained the total loss of the model as follows:

$$Total\_Loss = Margin\_Loss + Loss \tag{17}$$

which is computed to participate in the gradient descent of the model, thus optimizing the model parameters.

3.4. Experimental setup

Since we can directly generate comments of static anomaly according to our pre-established template, the fluency and correctness of the generated comments can be guaranteed as long as the anomaly type of an indicator is located accurately. To this end, we conducted automatic evaluation on the 336 raw records and human evaluation on the 3024 constructed records, respectively, where we compared the consistency between the generated sentence and the annotated comment in automatic evaluation and artificially judged the correctness of the generated sentence in human evaluation. Automatic evaluation on raw records showed that the comments generated by template are all same with the annotated comments, while human evaluation results on constructed records showed that the comments generated by template are all correct, which indicated the effectiveness of our template-based strategy in fluently and correctly generate the comments of static anomaly. Therefore, we mainly conducted experiments to evaluate the performance of our DHICG model on generating dynamic trend comments in this paper.

3.4.1. Parameters

We divided our dataset into three parts: 2016 for training, 672 for validation, and 672 for testing. We set $N = 5$, which is the number of time steps for indicator values, and used Adam (Kingma and Ba, 2014) for optimization with a learning rate of 0.001 and a batch size of 256. The dropout was set to 0.3. The dimensions of word embedding and hidden states for both the encoder and decoder were set to 128, 256 and 256, respectively. The *margin* was set to 0.01. For CNN, we used a single convolutional layer and set the filter size to 2.

3.4.2. Models

To evaluate the effectiveness of the techniques we introduced, we conducted experiments with 12 models. Table 4 shows an overview of the compared models. We compared four types of models: baselines, our model, related works and ablated models. Models *MLP-ENC*, *CNN-ENC* and *BiGRU-ENC*, which did not take attention and contrastive penalty into account, were regarded as baselines. We also evaluated three models of related works to validate the superiority of our model in fluently and correctly generating anomaly analysis comments from time-series data of DHI indicators. To determine whether each component contributes to the results, we conducted 5 ablated experiments (e.g., *-global*), for example, *-global* is a model that does not use the global moving reference.

3.4.3. Evaluation metrics

Since annotated comments generally mention important semantic information, such as increase gradually and decrease sharply, which is critical to understand the comments. we used BLEU-4 (Papineni et al., 2002) to measure the matching degree between the comments generated by our model and the annotated comments written by humans. However, just mention the important information is not enough, the mentioned information should be as correct as possible, that is, consistent with the facts reflected by the numerical values of indicators. For example, a generated comment states that *the somatic cell count increases gradually* while the annotated comments states that *the somatic cell count decrease sharply*, although the generated sentence is fluent, its semantic information is wrong because the generated keywords is not correct, which will lead to false diagnosis in locating the management problems of dairy farms in the future.

Therefore, to furtherly assess the correctness of the generated comments, we calculated the accuracy, precision, recall, F-measure and *trend_accuracy* of keywords for description of single indicator, and calculated the accuracy, F-measure and *consitency_accuracy* of keywords for description of multiple indicators. Note that the *trend_accuracy* is the proportion of samples that correctly describe the change trend but may not correctly describe the change degree in the total samples, such as *increase sharply* is regarded as positive sample even we annotated *increase gradually*, this metrics we proposed is used to evaluate the ability to capture the change trend of our model. The *consitency_accuracy* is the proportion of samples whose joint trend description of multiple indicators is consistent with the trend of each of the multiple indicators in the total samples, such as *the cell score increases with the increase of the somatic cell count* is regarded as positive sample if the cell score and the somatic cell count are all tend to increase in actually, otherwise, the comment is regarded as negative sample if one of them is tend to decrease. We use this metric to evaluate our model's ability to maintain the consistency of the generated comments.

4. Results and discussions

4.1. Comparisons with different models

The evaluation results of baseline models, our model and related works are listed in Table 5. We calculated the BLEU score to evaluate the fluency of our model in generating comments, and other metrics for description of single indicator and multiple indicators, respectively, to detail asses the correctness of our model in generating keywords in the comments.

Firstly, our model outperforms the baselines which all do not take into account the attention and contrastive penalty mechanisms. In terms of BLEU, the performance of our model is 61.28, up by 4.93 % compared with the best baseline *BiGRU-ENC*. In terms of F-measures, our model has improved by 23.84 % and 53.07 % in comparison to the best *MLP-ENC* in single indicator and the best *BiGRU-ENC* in multiple indicators, respectively. This suggests that incorporating the attention and contrastive penalty enables the model to more fluently and correctly generate comments. In addition, the performance in *trend_accuracy* and *consitency_accuracy* demonstrate that our model can better capture the change trend and maintain the consistency of generated comments compared with baselines.

Secondly, our model also outperforms the models of related works, they only used global moving reference and standardization except the study of Uehara et al. (2020) which employed contrastive penalty additionally. The performance of our model has improved by 13.17 % in BLEU compared with the best previous model proposed by Uehara et al. (2020), and the F-measures in single indicator and multiple indicators achieve an improvement of 10.10 % and 4.10 %, respectively, meanwhile, the *trend_accuracy* and *consitency_accuracy* also perform the best, this is because we used local moving reference to capture changes between two adjacent months, which is conform to the real scenarios, and used attention mechanism to select proper contents. Specifically, the model of Murakami et al. (2017) is incapable to properly generate comments of our task, as it was proposed to deal with single indicator, namely, stock prices. Aoki et al., 2018 extended the model by introducing external resources but removing the multi-level representations of input data, which made the model be capable to deal with multiple indicators. To conduct experiments on our dataset, we adjusted the model of Murakami et al. (2017) in order to take into account multiple indicators as same as the model of Aoki et al., 2018. Comparison results in Table 5 shows that the latter performs better no matter in fluency or correctness, which indicates that complicated but insignificant features may not necessarily improve the model performance. In addition, compared to the model of Aoki et al., 2018, the model of Uehara et al. (2020) performs almost same in BLEU, but there is a significant improvement in F-measures, which also proves the effectiveness of contrastive penalty.

Interestingly, the performance of Uehara et al. (2020) is inferior to *MLP-ENC* in BLEU, but it significantly improves the correctness of generated comments both in descriptions of single indicator and multiple indicators. The same issue appears between *BiGRU-ENC* and the model proposed by Uehara et al. (2020). This implies that it is difficult to evaluate the correctness of generated comments by relying on BLEU score only.

4.2. Contributes of each component

Next, we compared the models to investigate how each component contributes to their performance. Firstly, we discussed which kind of encoder is more suitable for time-series numerical data in baseline models. Secondly, we explored the effect of different data formats on model performance according to the best encoder. Then, we conducted ablated experiments to see the contributes of pre-processing, attention and contrastive penalty modules, and the experimental results are shown in Table 7.

Encoder for time-series numerical values. As shown in Table 5, BiGRU encoder performs best among the three baselines in term of BLEU, which has improved by 5.30 % and 52.64 % compared with MLP encoder and CNN encoder. In term of correctness, BiGRU encoder performs much better than CNN encoder both in single indicator and multiple indicators, although CNN encoder achieves higher accuracy in multiple indicators, the *consitency_accuracy* is 0 %, which infers that CNN encoder completely loses the ability to identify the joint change trend between multiple indicators because of its simplicity. In comparison to MLP encoder, BiGRU encoder performs almost same in accuracy and F-measure, but it achieves an improvement of 6.54 % and 17.71 % in *trend_accuracy* and *consitency_accuracy*, respectively, which means BiGRU encoder is superior in capturing more correct trends and maintaining the consistency of the generated comments. Therefore, we employed BiGRU as the encoder of our model.

Effect of data format. As we mentioned above, complicated but insignificant features may be not benefit to the performance of data-to-text models. To this end, we used three types of data format as input of the *BiGRU-ENC* to determine the suitable format for our model. We illustrate the three data formats in Fig. 4.

Using different formats of data as the input of the BiGRU-based model, we obtained three models: *BiGRU-ENC*, *BiGRU-6* and *BiGRU-1*, which employed format (a), (b) and (c), respectively, and the comparison results of them are shown in Table 6. We found that it is more effective to input data one by one than to aggregate data into vectors, the greater the degree of aggregation, the more difficult it is for the model to learn the implicit knowledge hidden among the numerical values. To be specific, *BiGRU-1* performs the worst no matter in fluency or correctness, in particularly, both F-measure for multiple indicators and *consitency_accuracy* is 0 % which reflect that it is unable to deal with the descriptions of multiple indicators like *CNN-ENC*. *BiGRU-6* performs better than *BiGRU-1* with a BLEU of 54.74, but still decreased by 6.27 % compared with *BiGRU-ENC*. Note that the data format has a greater influence on *trend_accuracy* and *consitency_accuracy*, because the model could not explicitly find the data segment corresponding to aggregated vectors to generate comments. Therefore, if the computing resources is sufficient, it is helpful to improve the model performance by diving the input time-series data into pieces as much as possible.

Effect of pre-processing methods. As shown in Table 7, in term of BLEU, *–std* without using the standardization method performs worst with a decrease of 17.33 % compared with our *DHICG* model, while *–global* and *–local* have decreased by 5.17 % and 3.79 %, respectively. For description of single indicator, *–std* also performs worst with a decrease of 10.43 %, 27.61 % and 22.23 % in accuracy, F-measure and *trend_accuracy*, respectively, meanwhile, *–global* which did not use global moving reference and *–local* which did not use local moving reference have almost the same influences. For description of multiple indicators, *–local* and *–std* performs the similar in accuracy and F-measure with a decrease around 8 % and 7 %, respectively, worse than *–global* model, however, in term of *consitency_accurac*, *–std* model performs the worst with a decrease of 77.82 %, followed by *–global*, and *–local* performs the best. According to the results, we can see that all of the three pre-processing methods have influences no matter on fluency or correctness of our model, in which *–std* has the most significant influence on the performance of our model because the numerical values of our dataset have different dimensions.

Effect of attention mechanism. Our *DHICG* model has an improvement of 4.91 % in BLEU, 1.30 % and 0.27 % in F-measures, 4.47 % and 2.34 % in *trend_accuracy* and *consitency_accuracy*, respectively, compared with *–att* model which did not use attention mechanism. The results show that the attention mechanism improves the fluency of the model better than the correctness of the model, which demonstrate that attention mechanism is useful to make our model focus on important information which is used to generate proper comments. For example, *–att* may generate a comment that includes 4 indicators, but the annotated comment involves 7 indicators in fact. This problem has been greatly alleviated after employing the attention mechanism to our model.

Effect of contrastive penalty mechanism. In terms of BLEU, there is no significant difference between our *DHICG* model and *–con* model which did not use contrastive penalty mechanism. However, the accuracy, F-measure and *trend_accuracy* of *DHICG* are up by 17.78 %, 23.57 % and 18.98 % in description of single indicator, 43.32 %, 46.16 % and 9.38 % in description of multiple indicators compared with *–con*, respectively. The results show that the contrastive penalty mechanism improves the correctness of the model better than the fluency of the model, which demonstrate the effectiveness of the contrastive penalty mechanism for generating correct keywords of the comments.

To further validate the effectiveness of the rules for generating contrastive examples we set, we conducted additional experiments and compared 3 models: *BiGRU-ENC*, *-att0* and *-att*, where *BiGRU-ENC* was the baseline using neither of the attention and contrastive penalty, *-att0* and *-att* did not use attention but used rules shown in Table 8 and Table 3 for generating contrastive examples, respectively, to realize contrastive penalty. The experimental results are shown in Table 9. We can see that there is no significant difference between *BiGRU-ENC* and *–att0*, although *–att0* performs a little better, it still did not achieve our expected results. However, *–att* achieved significant improvement in accuracy, F-measure, *trend_accuracy* and *consitency_accuracy* compared with *BiGRU-ENC*. This implies that contrastive penalty is absolutely beneficial to the model, but how to properly set the rules for generating contrastive examples is critical to our model. The reason why we set the rules shown in Table 3 is because that we compared the numbers of different types of wrong keywords generated by *BiGRU-ENC* according to an original keyword, then we chose the words that appeared frequently as the substitution keywords. For example, the model was expected to generate *increase gradually*, but it generated some other words, among which the proportion of *decrease gradually* is relatively high while proportions of other words are low, therefore, we mainly used *decrease gradually* to substitute *increase gradually* in generating contrastive examples.

5. Conclusion

In this paper, we proposed a framework to automatically generate anomaly analysis comments of DHI interpretation reports from time-series data of DHI indicators. We used a series of pre-set anomaly degree matrixes to detect static anomaly of indicators and generate comments according to templates, and proposed an improved end-to-end data-to-text model to jointly detect dynamic trends of indicators as well as generate comments. Experimental results show that the BLEU score of our proposed model achieves 61.28 with the accuracy and F-measure are 77.49 %, 74.86 % in descriptions of single indicator and 70.83 %, 73.84 % in descriptions of multiple indicators, respectively. Our model outperforms the baselines and models of related works, which is suitable for the preliminary description of DHI indicators, and can effectively promote the efficiency and quality of writing DHI interpretation reports. However, our model still has problems as follows: (1) the rules for generating contrastive examples are relied on dataset, (2) our model performs not well in describing *remain steady* and *fluctuate slightly*, (3) the proposed framework tackles our two subtasks of describing static anomaly and dynamic trend separately. For problems (1) and (2), we need to collect more real-world DHI reports to extend our dataset, so as to formulate more general rules and improve the generalization and robustness of our model. For problem (3), we aim to build an integrated model to deal with it incorporating with knowledge graph in the future.

CRediT authorship contribution statement

Xiao Han: Methodology, Data curation, Software, Validation, Writing – original draft, Writing – review & editing. **Meng Gao:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Formal analysis, Supervision, Funding acquisition. **Weizheng Shen:** Conceptualization, Supervision, Funding acquisition. **Huihuan Liu:** Resources, Data curation, Validation. **Baisheng Dai:** Software, Funding acquisition. **Yongqiang He:** Resources, Validation. **Huixin Liu:** Software.

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.

Data availability

The authors do not have permission to share data.

Acknowledgements

This work was financially supported by the Natural Science Foundation of Heilongjiang Province of China (Grant No. LH2022C031), the Key Research and Development Program of Heilongjiang Province of China (Grant No. 2022ZX01A24) and the earmarked fund for CARS36.

References

Uehara, Y., Ishigaki, T., Aoki, K., Noji, H., Goshima, K., Kobayashi, I., Takamura, H., Miyao, Y., 2020. Learning with contrastive examples for data-to-text generation. In: Proceedings of the 28th International Conference on Computational Linguistics.2352–2362.https://doi.org/10.18653/v1/2020.coling-main.213.

Murakami, S., Tanaka, S., Hangyo, M., Kamigaito, H., Funakoshi, K., Takamura, H., Okumura, M., 2021. Generating weather comments from meteorological simulations. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics(Main Volume).1462–1473.https://doi.org/10.18653/v1/2021.eacl-main.125.

Aoki, T., Miyazawa, A., Ishigaki, T., Goshima, K., Aoki, K., Kobayashi, I., Takamura, H., Miyao, Y., 2018. Generating market comments referring to external resources. In: Proceedings of the 11th International Conference on Natural Language Generation.135–139.https://doi.org/10.18653/v1/W18-6515.

Banaee, H., Ahmed, M.U., Loutfi, A., 2013. A Framework for Automatic Text Generation of Trends in Physiological Time Series Data. 2013 IEEE International Conference on Systems, Man, and Cybernetics.3876–3881.https://doi.org/10.1109/SMC.2013.661.

Banaee, H., Loutfi, A., 2015. Data-driven rule mining and representation of temporal patterns in physiological sensor data. IEEE journal of biomedical and health informatics.19 (5), 1557–1566.https://doi.org/10.1109/JBHI.2015.2438645.

Cascallar-Fuentes, A., Gallego-Fernández, J., Ramos-Soto, A., Saunders-Estévez, A., Bugarín-Diz, A., 2022. Automatic generation of textual descriptions in data-to-text systems using a fuzzy temporal ontology: Application in air quality index data series. Applied Soft Computing.119, 108612.https://doi.org/10.1016/j.asoc.2022.108612.

Hunter, J., Freer, Y., Gatt, A., Reiter, E., Sripada, S., Sykes, C., 2012. Automatic generation of natural language nursing shift summaries in neonatal intensive care: BT-Nurse. Artificial intelligence in medicine.56 (3), 157–172.https://doi.org/10.1016/j.artmed.2012.09.002.

Konstas, I., Lapata, M., 2013. Inducing document plans for concept-to-text generation. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing.1503–1514.

Liang, P., Jordan, M.I., Klein, D., 2009. Learning semantic correspondences with less supervision. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. 91-99.

Murakami, S., Watanabe, A., Miyazawa, A., Goshima, K., Yanase, T., Takamura, H., Miyao, Y., 2017. Learning to generate market comments from stock prices. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics.1374–1384.https://doi.org/10.18653/v1/P17-1126.

Papineni, K., Roukos, S., Ward, T., Zhu, W. J., 2002. Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 311-318.

Riza, L.S., Putra, B., Wihardi, Y., Paramita, B., 2019. Data to text for generating information of weather and air quality in the R programming language. Journal of Engineering Science and Technology.14 (1),498-508.

Huang, J., Li, Y., Ping, W., & Huang, L., 2018. Large margin neural language model. arXiv preprint arXiv:1808.08987.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Kingma, D. P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Noji, H., Takamura, H., 2020. An analysis of the utility of explicit negative examples to improve the syntactic abilities of neural language models. arXiv preprint arXiv:2004.02451.

Gong, H., Feng, X., Qin, B., & Liu, T., 2019. Table-to-text generation with effective hierarchical encoder on three dimensions (row, column and time). arXiv preprint arXiv:1909.02304.

Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., Weston, J., 2019. Neural text generation with unlikelihood training. arXiv preprint arXiv:1908.04319.

Wiseman, S., Shieber, S. M., & Rush, A. M., 2017. Challenges in data-to-document generation. arXiv preprint arXiv:1707.08052.

Yan, A., He, Z., Lu, X., Du, J., Chang, E., Gentili, A., McAuley, J., Hsu, C. N., 2021. Weakly supervised contrastive learning for chest x-ray report generation. arXiv preprint arXiv:2109.12242.