

Improving AI English Translation Systems By Using Multilingual Models

Ke Li*

School of Foreign Languages, Wuhan Business University, Hubei, Wuhan, 430056, China

* Corresponding author. E-mail: nicoletrans17@163.com, keli978787@outlook.com

Received: Mar. 23, 2026; Accepted: July. 06, 2026

Artificial intelligence (AI)-based English translation systems using large language models enhance translation accuracy but rely heavily on large datasets, making them prone to bias and challenges in handling subtle linguistic contexts. This study incorporates advanced AI techniques to improve the precision, accuracy, and overall performance of English translation systems in addressing linguistic issues. A Scalable Sailfish-tuned Adaptive Gated Recurrent Unit (SS-AGRU) model is proposed to maximize translation accuracy while improving adaptability and efficiency. The dataset consists of multilingual parallel data containing diverse sentence structures, idiomatic expressions, and domain-specific content to enable robust model training. Preprocessing includes handling missing data, stopword removal, and tokenization to normalize text. Word2Vec is employed to capture semantic relationships, while TF-IDF highlights term importance, enhancing input quality and model performance. The proposed approach integrates AGRU with an optimized Scalable Sailfish (SS) algorithm, improving sequence processing, contextual understanding, parameter optimization, and computational efficiency. Experimental results demonstrate strong performance, achieving a BLEU score of 50.12, translation time of 38.7 minutes, low TER of 0.337, and WER of 8.89. The SS-AGRU model outperforms baseline methods in translation accuracy, contextual comprehension, and computational efficiency. Overall, combining SS-AGRU with the optimized SS algorithm significantly enhances accuracy, flexibility, and computational performance in AI-based language translation systems.

Keywords: English translation, AI, Large language models, Scalable Sailfish Algorithm-tuned Adaptable Gated Recurrent Unit (SS-AGRU)

© The Author(s). This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are cited.

http://dx.doi.org/10.6180/jase.202612_35.044

1. Introduction

Neural network-based machine translation models are increasingly replacing human translators by effectively handling complex, high-dimensional multilingual data [1]. Conventional translation methods struggle with low-resource, on-demand tasks due to linguistic complexity, including varied sentence structures and idiomatic expressions, resulting in reduced accuracy [2]. Recent models like GPT-3 leverage large datasets and self-attention to capture long-range dependencies, enabling accurate, human-like text generation and translation [3].

Multilingual translation systems of large scale are being

developed to develop universal models to run several languages with one model [4]. This saves on engineering work and improves performance over conventional bilingual models [5]. Bilingual terminology alignment and cross-lingual representation remain challenging, addressed by multi-level feature fusion using Transformer and BiLSTM to improve semantic alignment and translation accuracy [6]. AI-enhanced translation tools enhance coherence and are able to reduce errors which are also making translation overall better [7]. This research uniquely integrates SSA with AGRU, where SSA enhances hyperparameter optimization through improved exploration-exploitation

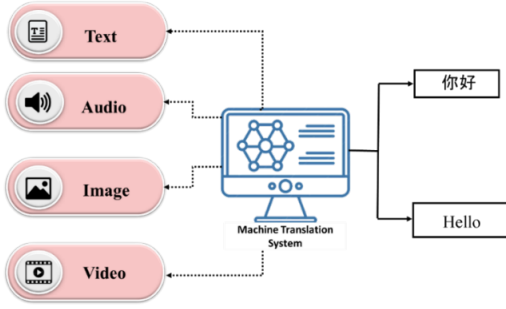


Fig. 1. AI-based English translation workflow with multi-modal inputs and processing stages

balance, outperforming traditional methods like GA and PSO to achieve better translation accuracy and efficiency. Fig. 1 illustrates an AI-powered English translation system that accepts text, audio, image and video inputs.

1.1. Study Objective

Objective of this research is to improve accuracy and fluency of English translation system by applying advanced AI techniques. It leverages LLM models to improve performance and address linguistic complexities. Scalable Sailfish-tuned Adaptable Gated Recurrent Unit method is employed to successfully improve translation accuracy. This research is highly effective because they can learn complex patterns in datasets and improve translation by understanding context, grammar and nuances in the source language.

1.2. Key contributions

Key insights of this research are as follows:

- To collect data based on multilingual parallel corpora, featuring diverse sentence structures, idiomatic expressions and domain-specific content.
- To apply a preprocessing method that involved handling missing data, stop word removal, and tokenization to standardize text.
- To determine which are the important features of the received data by using Word2Vec and TF-IDF.
- To perform the recognition method utilizing GRU (SS-AGRU) to improve translation accuracy and increase translation task performance.

1.3. Paper Organization

The relevant studies are outlined in Section 2. Section 3 then discusses methodologies, research design and data collection. Section 4 reveals the results. Section 5 discusses

the conclusion, including its limits and potential future applications.

2. Related work

This work evaluates and compares NMT and LLM translation performance using metrics like BLEU, CHRF, and TER across different token sizes, directions, and language pairs [8]. Translation systems have evolved from rule-based to neural approaches, leading to LLM-based online English translation methods [9].

The language training model, which improved the translation performance's capacity for semantic understanding, was introduced [10] using the random shielding technique. Transfer learning enhances machine translation by improving transformer-based word alignment using corpus optimization, enabling flexible and accurate sentence conversion [11]. On-demand translation faced word choice accuracy issues, addressed by combining fuzzy decision algorithms with deep learning, while recent advances in LLMs further enhance translation quality and comprehension [12] to handle translation tasks more effectively than NMT. According to the results, all of the chosen systems generally translate non-literary texts better, and the translation quality of these LLMs was not appreciably better than that of the NMT system. LLMs improve translation quality through better context understanding and multilingual support, but face scalability challenges due to high computational and data requirements. GRUs handle sequential data more efficiently than LSTMs on smaller datasets and are widely used in industry-level machine translation models [13] using the English-German data-set of WMT2021, which initially consisted of more strings from German news with parallel English translations. Using the quaternion cluster in the updated systems technique, phrase antecedent and postscript probability were computed. Integration of parallel English-Chinese corpora into the semantic recognition system using semantic analysis was performed [14]. Grammar, superficial and deep semantic labeling were used in a Chinese-English translation system, while an AI-based framework using ML models like Gaussian Naive Bayes, Decision Tree, and Random Forest was developed to improve investigative prediction accuracy [15].

2.1. Research Gaps

Significant progress has been made in AI-based English translation models, but challenges remain in adapting large language models to real-time, in-context learning, particularly for complex languages like English and Chinese. LLMs are great for adapting to domain-specific terms, but

often struggle with specialized translation needs for idiomatic or nonliterary texts. While combining AI with human expertise can boost efficiency, finding the right balance between automation and human input is still unclear. These issues highlight the need for more flexible models to handle diverse linguistic challenges and deliver high-quality translations. To address these challenges, proposed SS-AGRU model aims to optimize translation accuracy and improve model adaptability and efficiency.

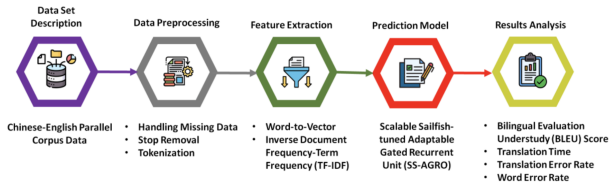


Fig. 2. Proposed SS-AGRU translation system with preprocessing, feature extraction, and SSA optimization

3. Methodology

A brief explanation of processes comprised in this research is provided in this section to compute the overall efficiency of English translation system using AI. By employing SS-AGRU, enhanced model flexibility, translation accuracy, and translation task performance are provided. Fig. 2 represents the steps involved in the proposed approach.

3.1. Dataset Description

The dataset known as Chinese-English Parallel Corpus Data is utilized for analysis of English translation performance. It was collected from a publicly available platform called Kaggle [16]. This dataset provides parallel sentence pairs for Chinese and English, enabling effective training and evaluation of translation models. Language translation studies, cultural research, and natural language processing (NLP) tasks widely utilize such parallel corpora.

3.2. Data preprocessing

Preprocessing is an essential step that cleans, transforms, and integrates raw data using techniques like missing data handling, stop word removal, and tokenization to improve DL-based translation quality.

- **Handling Missing Data:** Missing data patterns represent the values that are absent in an input data set. In this method, any rows or columns that contain missing values are removed from the dataset.
- **Stop Words Removal:** Common words like conjunctions, and articles, are eliminated from the list of tokens in the subsequent text pre-processing step. The

following are the outcome of using the stop word removal procedure on the aforementioned equation (1).

$$\text{filtered}_{\text{tokens}} \equiv (\text{example, tokenization}) \quad (1)$$

where token lists eliminated from the stop words. Eliminating stop words makes the vocabulary smaller, which speeds up computation. Despite the potential influence of the stop words elimination process on sequence-related information in case of machine translations, this step is conducted sparingly. Its influence on the resulting translation output is limited due to the preservation of context-relevant terms.

- **Tokenization:** Tokenization is a key preprocessing step in text classification that directly impacts model performance by converting text into tokens while removing blank or whitespace-only data before processing. The term frequency (TF) of a term t in a document d is calculated as, equation (2) is utilized.

$$TF(t, d) = \frac{f_{t,d}}{\max(f_{t',d})} \quad (2)$$

where $f_{t,d}$ denotes the frequency of term t in document d , and $\max(f_{t',d})$ represents the maximum frequency of any term in the same document.

Tokenization can be done in various ways, such as word-level tokenization, subword tokenization, and multilingual techniques like BPE (Byte-Pair Encoding) and Sentence Piece. Subword tokenization is especially useful for multilingual translation because it deals effectively with rare words.

3.3. Feature Extraction

Feature extraction in an English translation model is applied to the process of identifying and selecting relevant information from the input data that the model uses to generate accurate translations in the target language. This research involves word2vec and TF-IDF.

- **Word2Vec:** Word2Vec uses Skip-gram to represent words as vectors capturing semantic and syntactic relationships, with parameters like window size 5 and embedding dimension 100 to balance efficiency and translation accuracy.
- **Inverse Document Frequency-Term Frequency:** TF-IDF measures word importance by combining term frequency and inverse document frequency, giving lower weight to common words like "the" while assigning higher weight to rare, informative terms, though it struggles to capture semantic relationships in multilingual contexts.

Word2Vec and TF-IDF features are integrated to leverage both semantic and statistical representations of text. Specifically, TF-IDF weights are applied to the corresponding Word2Vec embeddings to emphasize important terms.

3.4. Proposed Scalable Sailfish Algorithm-tuned Adaptable Gated Recurrent Unit (SS-AGRU)

SSA, inspired by sailfish hunting behavior, is used for hyperparameter optimization in ML models like AGRU, complementing AGRU's role in sequence-based tasks such as language translation by efficiently exploring and selecting optimal parameter sets.

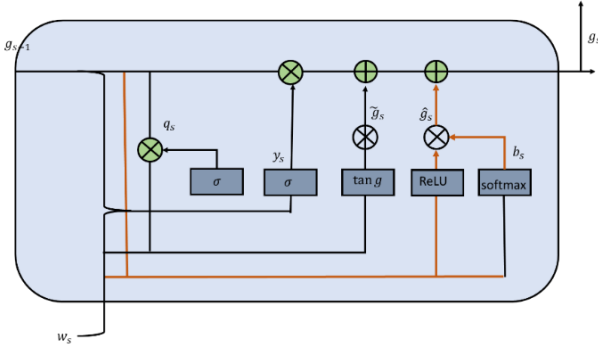


Fig. 3. Enlarged and clearly formatted structure of the AGRU model showing update and reset gates with hidden state transitions

3.4.1. Adaptable Gated Recurrent Unit (AGRU)

AGRU is a sequential learning model that integrates update and reset gates with an attention mechanism, where attention weights are computed using the previous input and current hidden state to improve context understanding in language processing tasks. The AGRU model's structure is shown in Fig. 3. In particular, it is possible to calculate an attention weight by using equation (4).

AGRU's attention mechanism helps improve contextual awareness by placing importance scores on relevant input sequences. This feature allows the network to emphasize critical words in the process of translating texts into their desired form.

The ability of the AGRU network to learn long-range dependencies is facilitated by the gating process in the recurrent cells, which controls the movement of information between different timesteps. Different from the transformer-based networks that depend on positional encoding to learn sequence information, the AGRU network maintains the order of information via its hidden state in equation (3).

$$y_s = \sigma(X_y[w_s, g_{s-1}] + a_y) \quad (3)$$

The attention weights are calculated for each time step using a softmax calculation based on the alignment scores that represent the relevance between the input being processed and the previous hidden states. This gives the ability to pay more attention to relevant tokens during the translation process.

Where, X_y denotes a_y is the update gate bias and weight of the update gate bias where σ represents the sigmoid function in equation (4).

$$\tilde{g}_s = \tan g(X_{\tilde{g}}[w_s, q_s \odot g_{s-1}] + a_{\tilde{g}}) \quad (4)$$

Here, $\tan \odot$ is the function of the hyperbolic tangent, $X_{\tilde{g}}$ indicates the weights of the update candidates, $a_{\tilde{g}}$ symbolizes the bias of the update candidate value in equation (5).

$$g_s = y_s \odot g_{s-1} + (1 - y_s) \odot \tilde{g}_s \quad (5)$$

Here, y_s has the ability to regulate how much information is $\tilde{g}_s$ based on the state that is hidden g_s symbolizes and g_{s-1} reset gate of the prior time and the candidate state at time s . The hidden state update in the AGRU model is computed as, ReLU explain in equation (6).

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \quad (6)$$

where h_t is the current hidden state, h_{t-1} is the previous hidden state, z_t is the update gate, $\tilde{h}_t$ is the candidate hidden state, and $\odot$ denotes element-wise multiplication.

3.4.2. Scalable Sailfish Algorithm (SSA)

SSA is a nature-inspired metaheuristic optimization algorithm that enhances global search and exploration diversity, improves LLM-based English translation accuracy, and avoids local minima issues compared to PSO and Genetic Algorithms.

- Initialization of the location of the sardine and sailfish populations: Finding the sardines and sailfish in the search space is crucial for the population is created at random in the solution space.
- Elite selection method: Elites are known to attack sardines and cause significant harm to them, making them the least fit members of the sardine population.
- Alternate strategy for attack: Sailfish behavior of hunting involves adjusting one's posture based on where other sailfish are in relation to sardines. The sailfish position update formula is shown by equation (7).

$$W_j^{(s)} = W_{\text{elite}}^{(s-1)} - \lambda_j^{(s-1)} \times \left(\text{rand} \times \left(\frac{W_{\text{elite}}^{(s-1)} - W_{\text{injured}}^{(s-1)}}{2} \right) - W_j^{(s-1)} \right) \quad (7)$$

Where, $W_{\text{injured}}^{(s-1)}$ represents the sardine that suffered severe injuries in the $s - 1$ iteration, $W_j^{(s-1)}$ is the j_{th} sailfish in the $s - 1$ iteration. $W_{\text{elite}}^{(s-1)}$ is the top-tier sailfish in the $s - 1$ iteration.

- Evasion Method: In each iteration, sardines will evaluate the position and attack power of the elite sailfish and modify their positioning to evade the sailfish's pursuit. It is expressed by equation (8).

$$\begin{cases} E_j^{(s)} = \text{rand} \times (W_{\text{elite}}^{(s-1)} - W_j^{(s-1)} + R^{(s-1)}) \\ R^{(s-1)} = B \times (1 - (2 \times (s - 1) \times \zeta)) \end{cases} \quad (8)$$

Here, $E_j^{(s)}$ is the i_{th} sardine of $s - 1$ iteration, $E_j^{(s)}$ is the repetition of sardine and $R^{(s-1)}$ shows how strong a sailfish's assault is in the $s - 1$ iteration, B and ζ are measures of assault strength.

- Hunting Method: The position of the captured sardine is replaced with the sailfish when the sardine's fitness is lower than the sailfishes, as expressed in equation (9).

$$W_j^{(s)} = W_j'^{(s)} \text{ if } E(W_j^{(s)}) > E'(W_j'^{(s)}) \quad (9)$$

Where $W_j^{(s)}$ is the sailfish's location within the s iteration and $W_j'^{(s)}$ represents the sardine's location in the s iteration.

- Weight inertia: SSA's alternating attack and pursue procedure incorporates weight inertia, which enhances the ability of sardines and sailfish to hunt locally.
- Global search formula: To improve the algorithm's global search performance and prevent the SSA from drifting toward local optimization.
- Levy flight Method: The Levy flight method, a random search strategy that makes use of the Levy distribution, broadens the diversity of the algorithm population.

SS-AGRU improves translation accuracy and efficiency by using SSFO/SSA to optimize AGRU hyperparameters such as learning rate, sequence length, and dropout rate

through iterative evaluation, enabling adaptive real-time learning and reduced translation time. The optimized values of the hyperparameters are used during each iteration of the optimization process and updated by SSA before AGRU training. These parameters are introduced into the training process before testing the model, as shown in Table 1.

Table 1. Hyperparameter ranges used in the proposed SS-AGRU model

Hyperparameter	Range
Learning Rate	0.001-0.01
Sequence Length	100-1000
Number of Epochs	20-50
Iteration	1000-10000
Dropout Rate	0.2-0.5
Hidden State Size	64-256
Embedding Dimension	50-300

SSA optimization also tunes AGRU hyperparameters such as hidden units and embedding dimensionality to expand the search space and improve accuracy, while keeping the learning rate between 0.001 and 0.01 to ensure stable convergence, faster training, and efficient translation performance.

The pseudocode is structured to clearly represent initialization, fitness evaluation, iterative optimization, and convergence steps to ensure consistency between theoretical formulation and implementation. The procedures and techniques used in the suggested SS-AGRU for improving the English translation systems using AI is deliberated in Algorithm 1.

4. Results and discussion

This section delivers the outcomes procured by the proposed SS-AGRU model implemented in improving English translation performance. Certain performance metrics are involved to evaluate the performance and are also discussed. Further, comparative analysis is provided to compute the effectiveness of the proposed with other state-of-art methods.

4.1. Experimental Setup

The system described is equipped with an Intel i7 CPU, offering strong multi-core performance and making it well-suited for demanding computation tasks with 16 GB of RAM, and NVIDIA GTX 1050i GPU are the components of a computer system environment that uses Python to perform its operations. The system utilizes CUDA support (version 11.x) for GPU acceleration to improve training effi-

Algorithm 1. SS-AGRU for Improving the English Translation Systems**Step 1: Define AGRU Model**

```
def AGRU(input_sequence, hidden_state, weights, biases):
    update_gate = sigmoid(weights['update'] @ [input_sequence, hidden_state] + biases['update'])
    candidate_state = tanh(weights['candidate'] @ [input_sequence, update_gate * hidden_state]
                          + biases['candidate'])
    hidden_state = update_gate * hidden_state + (1 - update_gate) * candidate_state
    return hidden_state
```

Step 2: Initialize SSA parameters

```
def initialize_SSA_population(n_sailfish, n_sardines, dims):
    sailfish_population = random_initialize(n_sailfish, dims) # Random initialization for sailfish
    sardine_population = random_initialize(n_sardines, dims) # Random initialization for sardines
    return sailfish_population, sardine_population
```

Step 3: Evaluate fitness of each population

```
def evaluate_fitness(population, AGRU_model, data):
    fitness_scores = []
    for individual in population:
        # Use the individual's position (hyperparameters) to tune the AGRU model
        hyperparameters = map_to_hyperparameters(individual)
        AGRU_model.set_hyperparameters(hyperparameters)
        performance = AGRU_model.evaluate(data) # Evaluate the model's performance
        fitness_scores.append(performance)
    return fitness_scores
```

Step 4: SSA Main Loop

```
def SSA_AGRU_optimization(data, n_sailfish, n_sardines, dims, max_iterations):
    Initialize populations
    sailfish_population, sardine_population = initialize_SSA_population(n_sailfish, n_sardines, dims)
    Evaluate initial fitness
    sailfish_fitness = evaluate_fitness(sailfish_population, AGRU, data)
    sardine_fitness = evaluate_fitness(sardine_population, AGRU, data)
    Return the best hyperparameters from the elite sailfish after optimization
    best_hyperparameters = elite_sailfish
    return best_hyperparameters
Set the AGRU model with the best hyperparameters found by SSA
AGRU_model.set_hyperparameters(best_hyperparameters)
AGRU_model.train(data)
End
```

ciency. A batch size of 32 is used to balance memory usage and computational performance during model training.

The current study utilizes a Chinese-English parallel corpus, future work will incorporate multiple multilingual datasets to further validate the generalization capability of the proposed model across diverse language pairs.

4.2. Experimental Outcomes

Performance is evaluated using BLEU for translation quality (higher is better), TER and WER for error measurement (lower is better), and translation time, collectively assessing the efficiency and accuracy of machine translation outputs. These metrics are tabulated in Table 2.

As inferred from the table, proposed SS-AGRU produces BLEU (50.12), translation time (38.7 min), TER (0.337) and WER (8.89). This demonstrates that the proposed model

procures high performance and improves the English translation accuracy.

The proposed SS-AGRU framework is also compared theoretically to the architecture that uses transformers. Transformers make use of self-attention and position embeddings to perform sequence modeling tasks. However, despite their high level of effectiveness, transformer models have some limitations in terms of the required amount of computational power and time to train. Comparative analysis based on existing studies such as bidirectional recurrent neural networks (Bi-RNN) [17], LSTM [17], convolutional neural network (CNN) [17] and GRU [17] and the proposed SS-AGRU is implemented. Thus, the following discussion is held regarding the outcomes generated by each approach depending on evaluation variables. Table 3 indicates the BLEU and translation time outcomes

Table 2. Outcomes of Proposed Model

Metrics	BLEU	Translation Time (min)	TER	WER
SS-AGRU [Proposed]	50.12	38.7	0.337	8.89

Table 3. Outcomes of BLEU and Translation Time

Method	GRU [17]	CNN [17]	LSTM [17]	Bi-RNN [17]	SSA-AGRU [Proposed]
BLEU	0.77	21.8	47.14	49.67	50.12
Translation Time (min)	39	205	43	40	38.7

generated by the proposed and existing model.

Both CNN [17] and GRU [17] models achieve TER = 0.826 and WER = 21.00. While LSTM [17] model performs better than the GRU and CNN models, with TER (0.361) and WER (10.00). Whereas, the TER (0.342) and WER (9.00) was produced by Bi-RNN [17]. The proposed SS-AGRU method achieves the best results with the lowest TER of 0.337 and WER of 8.89, demonstrating superior translation performance with fewer errors.

4.3. Ablation Study

Ablation studies show that GRU performs poorly due to limited contextual learning, while AGRU improves results via attention mechanisms, and combining SSA with GRU enhances accuracy by effectively optimizing model parameters. In all aspects, the combination of the SS-AGRU model gives the best results having the BLEU score of 50.12, TER of 0.337 and WER of 8.89, as shown in Table 4.

Table 4. Performance Comparison for Ablation Study

Model Configuration	BLEU	TER	WER
GRU (Baseline)	47.10	0.362	10.20
AGRU (without SSA)	48.85	0.350	9.75
SSA + GRU	49.20	0.345	9.40
SS-AGRU (Proposed)	50.12	0.337	8.89

5. Discussion

SS-AGRU outperforms GRU, CNN, LSTM, and Bi-RNN in English translation by achieving the highest BLEU score (50.12) and fastest translation time (38.7 minutes). It effectively balances translation quality and computational efficiency through optimized AGRU and SSA-based parameter tuning. GRU and CNN show poor BLEU scores and lower performance, with CNN also being unsuitable for real-time use due to higher processing time. LSTM and Bi-RNN perform better than basic models but still fall short of SS-AGRU. Overall, SS-AGRU delivers superior accuracy, speed, and contextual translation quality.

6. Conclusion and future scope

The proposed SS-AGRU-based English translation system enhances both accuracy and fluency by integrating LLM techniques with effective preprocessing, feature extraction, and optimization strategies. The model utilizes Word2Vec for semantic representation and TF-IDF for feature weighting, along with SS-AGRU for improved sequence learning. It achieves strong performance with a BLEU score of 50.12, TER of 0.337, WER of 8.89, and a reduced translation time of 38.7 minutes. These results clearly indicate that the proposed approach provides a better balance between translation quality and computational efficiency when compared to existing models such as GRU, CNN, and LSTM.

The study is limited by the use of a single dataset, which may restrict generalization and lead to potential overfitting in specific translation domains. This can reduce the model's adaptability when applied to diverse linguistic contexts. Future work can address these limitations by incorporating larger and more diverse multilingual datasets to improve robustness and generalization capability. Additionally, further research can focus on integrating more advanced deep learning architectures and optimization techniques to enhance translation accuracy, contextual understanding, and real-world applicability of the model.

Declaration:

Funding statement:

University-Industry Collaboration Talent Cultivation Project of Ministry of Education : On the Innovation Model Construction of BTI Internship Practice Base for Smart Translation Education (No. 230907205285510)

Data availability:

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Data availability statement:

No datasets were generated or analyzed during the current study

Conflict of interest

There is no conflict of interests between the authors.

Declaration of interests:

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.

Ethics approval:

Not applicable.

Patient consent statement:

We have not harmed any human person with our research data collection, which was gathered from an already published article.

Permission to reproduce material from other sources:

Yes, you can reproduce.

Clinical trial registration:

We have not harmed any human person with our research data collection, which was gathered from an already published article

Authors' contributions

All authors have made equal contributions to this article.

Availability of data and materials

The data sets used and analyzed during this study are available from the corresponding author on request.

Author disclosure statement

The authors declare that they have no competing interests

References

- [1] N. A. Mohamed, M. A. Zulkifley, A. A. Ibrahim, and M. Aouache, (2021) "Optimal Training Configurations of a CNN-LSTM-Based Tracker for a Fall Frame Detection System" *Sensors* **21**(19): 6485. DOI: [10.3390/s21196485](https://doi.org/10.3390/s21196485).
- [2] A. K. Mandal, R. Sen, S. Goswami, and B. Chakraborty, (2021) "Comparative Study of Univariate and Multivariate Long Short-Term Memory for Very Short-Term Forecasting of Global Horizontal Irradiance" *Symmetry* **13**(8): 1544. DOI: [10.3390/sym13081544](https://doi.org/10.3390/sym13081544).
- [3] E. Kasneci, K. Sessler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günemann, E. Hüllermeier, and S. Krusche, (2023) "Chat-GPT for Good? On Opportunities and Challenges of Large Language Models for Education" *Learning and Individual Differences* **103**: 102274. DOI: [10.1016/j.lindif.2023.102274](https://doi.org/10.1016/j.lindif.2023.102274).
- [4] C. Su, H. Huang, S. Shi, P. Jian, and X. Shi, (2020) "Neural Machine Translation with Gumbel Tree-LSTM Based Encoder" *Journal of Visual Communication and Image Representation* **71**: 102811. DOI: [10.1016/j.jvcir.2020.102811](https://doi.org/10.1016/j.jvcir.2020.102811).
- [5] J. Wang, J. Zhu, and Z. Yu, (2021) "Person Re-Identification Based on Attention Clustering and Long Short-Term Memory Network" *Journal of Electronic Imaging* **30**(3): 033014. DOI: [10.1117/1.JEI.30.3.033014](https://doi.org/10.1117/1.JEI.30.3.033014).
- [6] Q. Fang, (2025) "Terminology Alignment Based on Multi-level Feature Fusion for Japanese Scientific and Technological Literature Terminology Translation" *Journal of Applied Science and Engineering* **29**(2): 465–473. DOI: [10.6180/jase.202602_29\(2\).0021](https://doi.org/10.6180/jase.202602_29(2).0021).
- [7] P. K. Nagaraj, K. S. Ravikumar, M. S. Kasyap, M. H. S. Murthy, and J. Paul, (2021) "Kannada to English Machine Translation Using Deep Neural Network" *Ingénierie des Systèmes d'Information* **26**(1): 123–127. DOI: [10.18280/isi.260113](https://doi.org/10.18280/isi.260113).
- [8] J. Son and B. Kim, (2023) "Translation Performance from the User's Perspective of Large Language Models and Neural Machine Translation Systems" *Information* **14**(10): 574. DOI: [10.3390/info14100574](https://doi.org/10.3390/info14100574).
- [9] C. Wang, (2022) "AI-Based Heterogenous Large-Scale English Translation Strategy" *Mobile Information Systems* **2022**(1): 8344814. DOI: [10.1155/2022/8344814](https://doi.org/10.1155/2022/8344814).
- [10] X. Guo, (2022) "Optimization of English Machine Translation by Deep Neural Network under Artificial Intelligence" *Computational Intelligence and Neuroscience* **2022**(1): 2003411. DOI: [10.1155/2022/2003411](https://doi.org/10.1155/2022/2003411).
- [11] X. Han, (2024) "Analyzing the Impact of Deep Learning Algorithms and Fuzzy Logic Approach for Remote English Translation" *Scientific Reports* **14**(1): 14556. DOI: [10.1038/s41598-024-64831-w](https://doi.org/10.1038/s41598-024-64831-w).

- [12] X. Li, (2024) "Comparison of Translation Quality between Large Language Models and Neural Machine Translation Systems: A Case Study of Chinese-English Language Pair" **International Journal of Education and Humanities** 4(2): 121–128. DOI: [10.58557/ijeh.v4i2.213](https://doi.org/10.58557/ijeh.v4i2.213).
- [13] K. C. Lion, Y. H. Lin, and T. Kim, (2024) "Artificial Intelligence for Language Translation: The Equity Is in the Details" **JAMA** 332(17): 1427–1428. DOI: [10.1001/jama.2024.15296](https://doi.org/10.1001/jama.2024.15296).
- [14] L. Zhou, (2023) "English-Chinese Corpus Collection and Artificial Intelligence Translation Based on Dynamic Clustering Model" **International Journal of Science and Engineering Applications** 12(2): 63–69. DOI: [10.7753/IJSEA1202.1029](https://doi.org/10.7753/IJSEA1202.1029).
- [15] P. Alagarsundaram, (2023) "AI-Powered Data Processing for Advanced Case Investigation Technology" **Journal of Science and Technology** 8(8): 18–34. DOI: [10.46243/jst.2023.v8.i08.pp18-34](https://doi.org/10.46243/jst.2023.v8.i08.pp18-34).
- [16] Nexdata. *Chinese-English Parallel Corpus Data*. 2023. URL: <https://www.kaggle.com/datasets/nexdatafrank/chinese-english-parallel-corpus-data>.
- [17] H. Israr, S. A. Khan, M. A. Tahir, M. K. Shahzad, M. Ahmad, and J. M. Zain, (2023) "Neural Machine Translation Models with Attention-Based Dropout Layer" **CMC-Computers, Materials & Continua** 75(2): 2982. DOI: [10.32604/cmc.2023.035814](https://doi.org/10.32604/cmc.2023.035814).