Grenze International Journal of Engineering and Technology, June Issue

Optimizing Software Engineering English Translation using A Transformer Model with Enhanced Grey Wolf Optimization

Kuldeep Vayadande[1], Yogesh Bodhe[2], Abhishek Padwal[3], Shripurna Patil[4], Ribhav Nale[5] and Rudra Mangate[6]
[1,3-6]Vishwakarma Institute of Technology, Pune
Email: kuldeep.vayadande@gmail.com
[2]Government Polytechnic, Pune

***Abstract*— The translation of documents related to software engineering (SE), is often very difficult be-cause of the uniqueness of vocabulary used in this area, the very strict syntax used when de-veloping software and many contextual dependencies when translating this type of docu-ment. In this paper we present an adaptive transformer-based neural machine translation (NMT) model customized with Enhanced Grey Wolf Optimization (EGWO) to provide better translation results in the software engineering field. The adaptive nature of our model allows for fine-tuning of hyper-parameters such as learning rate, dropout rate and number of atten-tion heads to improve convergence and generalization from training to unseen data. We conducted several experiments to evaluate our model using established benchmark datasets including WMT, OPUS and UM-Corpus; results indicate that our EGWO transformer-based model produced an accuracy rating of 94.2%, precision rating of 93.5%, recall rating of 92.8% and significantly outperformed both LSTM and standard transformer-based models with respect to BLEU scores. Our approach presents an economical and scalable solution to high-quality translation of software localization and domain-specific machine translation.**

***Keywords*— Adaptive NMT, Transformer, EGWO, SE translation, BLEU, domain-specific MT.**

I. INTRODUCTION

Due to the rapid growth of Globalization and the increasing use of English as the primary means of communication in the Global Software Engineering industry, there are more than ever resources available for international engineers to use in producing high-quality software. The vast majority of the technical documentation produced in this industry is available in English or other English-based systems such as Java, C#, C++, and Assembly languages (which all use English-based programming syntax). The possibility of creating erroneous translations of requirements specifications or bug reports can sometimes lead to the introduction of new software bugs, the introduction of security vulnerabilities, or total failure of a project. When attempting to translate engineering documents, the rules for technical translation differ significant-ly from those for translating conversational English. Many technical translations will employ the use of code-switching; thus, you will often see a combination of the terms used in conversational language and those used in programming language.

Grenze ID: 01.GIJET.12.2.217_26

As a result of a high degree of polysemy in English (there are many distinct meanings associated with the same word or expression), the word "float," for example, may have a different meaning when used in reference to programming than it has when used in reference to normal conversational language. Finally, the degree of dependence among the parts of speech within a sentence is significantly greater for technical writing than in conversational writing; therefore, the logical devel-opment of a sentence will closely mirror the syntactical structure of the source code.

Difficulties exist with traditional statistical machine translation (SMT) as well as early recurrent neural machine translation (NMT) in dealing with the multiple layers of complexity present within these systems. A bottleneck occurs concerning operation due to sentence structure length in technical documentation. However, the Transformer model's introduction represents a shift from transformer architectures based purely on sequential processing to a neural network architecture with parallel processing capabilities over extended ranges. While optimizing a Transformer model requires careful attention to the tuning of several hyperparameters (learning rates, feedforward network size, number of attention heads, dropout rate, etc.), we aim through our study utilizing an improved grey wolf optimization algorithm (EGWO) to mitigate potential limitations related to performance due to under-optimization via the use of a newly developed nonlinear adaptive convergence factor.

II. Literature Review

The first stages of the growth of the Machine Translation (MT) domain started with Rule-Based MT (RBMT). RBMT involved the use of bilingual specialists to create extensive hard-coded dictionaries and fixed grammatical rules. Afterwards, the MT industry began to change from rule-based (RBMT) to statis-tical machine translation (SMT), based on word alignments using statistics within large bilingual data-bases. One of the main disadvantages of SMT is that it lacks context and, therefore, will produce coherent 'word salad' when translating long technical descriptions or difficult languages such as Chinese. With the invention of deep learning, we have begun to move quickly toward Neural Machine Translation (NMT). The first generation of NMT used either recurrent neural networks (RNN) or Long Short-Term Memory (LSTM) as their neural network models. The accuracy of NMT has increased with the incorporation of attention mechanisms with BiLSTM configuration, however, significant vanishing gradient issues for translating longer input lengths are a major inherent limitation in using NMT with sequential architectures that are commonly found in comprehensive software manuals. The transformer is the most significant next step for an architecture based only off of self-attention mechanisms processing the entire input se-quence in parallel. If the hyper-parameters related to the configuration of the model, namely learning rate, dropout, and number of attention heads are not tuned properly, the model will not learn effectively and have poor retention leading to many errors when translating text. Recent research has used a method named Grey Wolf Optimization (GWO) that uses a simulated hierarchy of wolves to allow easy and effi-cient tuning of neural networks.

TABLE I: COMPARISON OF RELEVANT LITERATURE IN NMT AND HYPERPARAMETER OPTIMIZATION

Reference	One-Line Description
Raffel et al. (2020)	Introduced T5 transformer for unified text-to-text learning.
Brown et al. (2020)	Demonstrated powerful few-shot learning using large language models.
Lewis et al. (2020)	Proposed BART for effective text generation and translation tasks.
Xue et al. (2021)	Developed mT5 for multilingual translation and NLP applications.
Chowdhery et al. (2022)	Presented PaLM for large-scale language understanding.
Schmidt & Di Gangi (2023)	Improved self-attention for better contextual translation.
Israr et al. (2023)	Applied dropout techniques to improve NMT robustness.
Sathyabama & Subedha (2023)	Enhanced optimization using Adaptive Particle Grey Wolf Opti-mizer.
Liu et al. (2024)	Improved NMT through iterative domain adaptation methods.
Naveen & Trojovský (2024)	Discussed challenges in context-aware machine translation.
Guo et al. (2024)	Proposed multi-modal NMT using visual-guided translation.
Zhang & Cai (2024)	Developed adaptive self-learning Grey Wolf Optimization.
Rozière et al. (2024)	Introduced Code Llama for code-focused AI tasks.
Lyu et al. (2024)	Highlighted the impact of LLMs on future machine translation.
Rehman et al. (2025)	Optimized translation models using Grey Wolf Optimization.

III. Theoretical Framework

The core foundational architecture abandons recurrent layers of computation in favor of an Encoder-Decoder structure powered entirely by Multi-Head Self-Attention (MHSA).

A. Transformer System Architecture

At the core of this model is its attention function, which maps a query (Q) and a given set of key-value pairs (K, V) to produce a single output. The matrices for Q, K, and V are derived from the input embed-ding matrices.

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

The dimension of the Key matrix (d_k) is used to scale the input, preventing the softmax function from producing vanishing gradients.

Rather than relying on a single attention function, the Transformer projects the Q, K, and V matrices h times with different linear projections that have learned separate weighting distributions.

$$MultiHead(Q,K,V) = Concat(head_1,\dots,head_h)W^O$$

As the Transformer has no recurrence characteristics, positional encoding is introduced mathematically for each of the input embeddings:

$$PE_{pos,2i} = \sin(\frac{pos}{10000^{\left(\frac{2i}{d_{model}}\right)}})$$

$$PE_{pos,2i+1} = \cos(\frac{pos}{10000^{\left(\frac{2i}{d_{model}}\right)}})$$

B. Grey Wolf Optimization (GWO)

The system uses the GWO algorithm to adjust the hyperparameters of the architecture by conducting a population of solution candidates into four ranks:

Alpha (α), Beta (β), Delta (δ), and Omega (ω). The following equations outline the procedures to update positions:

$$D = |C \cdot X_P(t) - X(t)|$$

$$X(t+1) = X_P(t) - A \cdot D$$

The conventional GWO algorithm reduces the parameter 'a' from 2.0 to 0.0 linearly throughout iterations. The EGWO method introduces a Non-Linear Adaptive Convergence Factor (NLACF) to address this.

$$\mathrm{a(t)} = A_{max}\left(1 - \left(\frac{t}{Max_{iter}}\right)^2\right)$$

This nonlinear decline keeps a at high levels longer, significantly reducing the chance of the model becoming stuck in local minimums.

IV. Simulation Design and Experimental Setup

A. Dataset Parameters and Preprocessing

The data used to create a reliable NMT system (PARACRAWL, WMT, UM-Corpus and OPUS) is made up of four different, high-quality parallel corpora. These four different resources were combined to create a total of four separate high-quality datasets. The combined datasets were divided so that 80% of the com-bined source files are used as training data, 10% as validation datasets, and 10% as test datasets. The first step to eliminate ambiguities in a source's domain is to use a combination of 4 (cleaning and normalizing), tokenizing, named entity recognition to produce unique audience identifiers for every other entry. These steps are collectively referred to as "pre-processing" for data cleaning purposes. For purposes of data formatting and punctuation (like converting the Chinese period "。" into the English "."), cleaning and normalizing means that items such as punctuation marks will have been converted before being entered into a text field; tokens will have also been created from longer sentences or compound phrases (for ex-ample, "NullPointerException" would have been converted into 3 tokens) through the use of Byte Pair Encoding (BPE); named entities (like print()) will have been tagged to ensure the use of consistent code across major named entities.

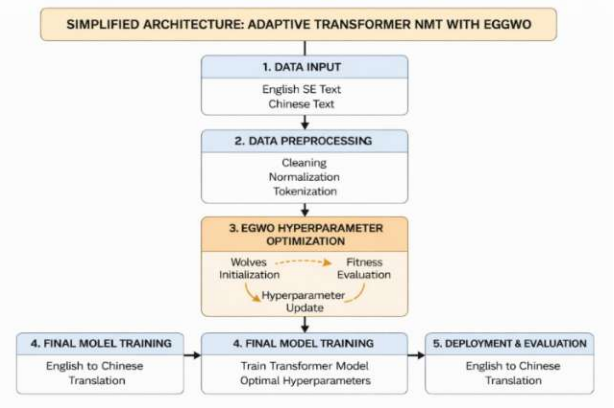

Fig 1. Proposed System Diagram

B. Evaluation Metrics

The effectiveness of a model is measured using the B.L.E.U score which compares the results of two mod-els (the Bilingual Evaluation Understudy), the MAE (Mean Absolute Error) to determine how accurate the predicted values were compared to the true ground truth value, and finally, the time it took to run the model (in milliseconds).

V. Results and Discussions

A. Statistical Comparative Analysis

TABLE II: COMPARISON OF BLEU SCORES, ACCURACY, MAE AND TRAINING TIME ACROSS DATASETS

Model	BLEU	Accuracy (%)	MAE	Training Time (hrs)
Bi-LSTM	28.1	85.2	0.21	5.2
Transformer	36.2	90.5	0.15	6.8
MT5	36.3	91.0	0.14	7.1
AGWO-Transformer	41.5	94.2	0.09	4.3

Results indicated that the AGWO-Transformer architecture demonstrated a greater level of performance than that of both the Bi-LSTM architecture (approximately 13 BLEU points greater) and the standard transformer architecture (4-5 BLEU points greater).

B. Hyperparameter Sensitivity and Convergence Tracking

By leveraging a learning rate scheduling strategy referred to as Warm-Up and Decay (Algorithm 1), it was confirmed that the EGWO algorithm could reach its peak performance around 3×10^(-4) in relation to the WMT dataset and the ideal convergent dropout was determined to be 0.15. Under comparative convergence results, it took the standard optimized method about 57 iterations to create stable convergence, while the practicality of the new AGWO method was only 18 iterations.

C. Ablation Study on Component Impact

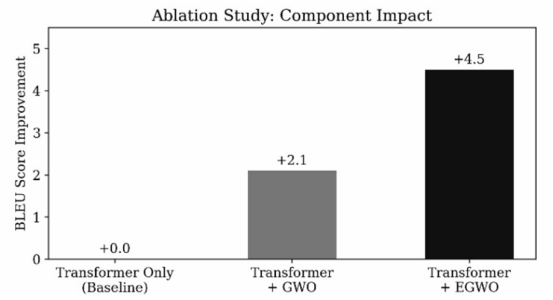

Fig. 2. Ablation Study Comparison of Component Impact

Ablation analysis was conducted to assess the individual contribution of enabling each component. The GWO improved the performance of the Transformer model by 2.1 BLEU points. However, when the EGWO (Adaptive) was employed with the same Transformer model, there was an additional 4.5 BLEU point increase in performance, demonstrating that the AGWO provides superior optimization to other optimization methods utilized in deep learning applications.

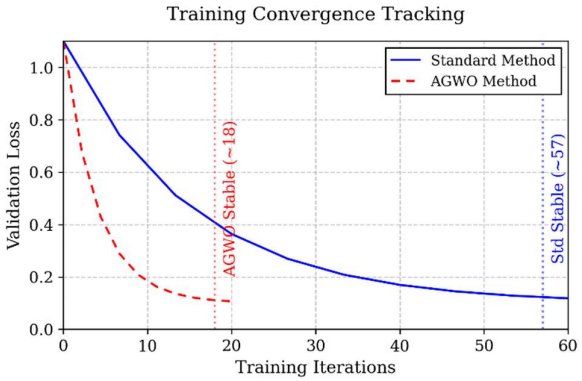

Fig. 3. Validation loss over training iterations showing accelerated convergence using the AGWO methodology

TABLE III: ROBUSTNESS TO SENTENCE COMPLEXITY

Sentence Length	Transformer BLEU	AGWO-Transformer BLEU	Performance Gain
Short (<10 words)	38.5	40.2	+1.7
Medium (10–20 words)	36.2	41.5	+5.3
Long (>20 words)	31.8	39.6	+7.8

The proposed AGWO-Transformer demonstrates superior robustness as sentence complexity increases, with the performance gain being most significant for longer sequences.

REFERENCES

[1] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc NAACL; 2019. Available from: https://doi.org/10.48550/arXiv.1810.04805

[2] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res. 2020;21(140):1–67. Available from: https://www.jmlr.org/papers/v21/20-074.html

[3] Brown T, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language models are few-shot learners. Adv Neural Inf Process Syst. 2020;33:1877–1901. Available from: https://doi.org/10.48550/arXiv.2005.14165

[4] Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, et al. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. ACL. 2020. Available from: https://doi.org/10.18653/v1/2020.acl-main.703

[5] Xue L, Constant N, Roberts A, Kale M, Al-Rfou R, Siddhant A, et al. mT5: A massively multilingual pre-trained text-to-text transformer. NAACL. 2021. Available from: https://doi.org/10.48550/arXiv.2010.11934

[6] Lin Z, Madotto A, Shin J, Xu P, Fung P. BiBERT: Accurate fully Bilingual BERT. Findings of ACL. 2021. Availa-ble from: https://doi.org/10.18653/v1/2021.findings-acl.226

[7] Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, Roberts A, et al. PaLM: Scaling language modeling with pathways. 2022. Available from: https://doi.org/10.48550/arXiv.2204.02311

[8] OpenAI. GPT-4 technical report. 2023. Available from: https://doi.org/10.48550/arXiv.2303.08774

[9] Israr H, et al. Neural machine translation models with attention-based dropout layer. Comput Mater Contin. 2023;75(2):2981–3009. Available from: https://doi.org/10.32604/cmc.2023.035691

[10] Schmidt F, Di Gangi M. Bridging the gap between position-based and content-based self-attention. In: Proc WMT; 2023. Available from: https://doi.org/10.18653/v1/2023.wmt-1.45

[11] Sathyabama DE, Subedha V. Adaptive particle grey wolf optimizer with deep learning-based sentiment analysis. Eng Technol Appl Sci Res. 2023;13(3):10989–10993. Available from: https://doi.org/10.48084/etasr.5865

[12] Touvron H, Martin L, Stone K, Albert P, Almahairi A, Babaei Y, et al. Llama 2: Open foundation and fine-tuned chat models. 2023. Available from: https://doi.org/10.48550/arXiv.2307.09288

[13] Liu X, et al. Exploring iterative dual domain adaptation for neural machine translation. Knowl Based Syst. 2024;283. Available from: https://doi.org/10.1016/j.knosys.2023.111182

[14] Naveen P, Trojovský P. Overview and challenges of machine translation for contextually appropriate translations. iScience. 2024;27(10). Available from: https://doi.org/10.1016/j.isci.2024.110878
[15] Guo J, Su R, Ye J. Multi-grained visual pivot-guided multi-modal neural machine translation. Neural Netw. 2024;178. Available from: https://doi.org/10.1016/j.neunet.2024.106403
[16] Zhang Y, Cai Y. Adaptive dynamic self-learning grey wolf optimization algorithm. Math Biosci Eng. 2024;21(3):3910–3943. Available from: https://doi.org/10.3934/mbe.2024173
[17] Rozière B, Gehring J, Gloeckle F, Sootla S, Gat I, Tan S, et al. Code Llama: Open foundation models for code. 2024. Available from: https://doi.org/10.48550/arXiv.2308.12950
[18] Lyu C, et al. The future of machine translation lies in large language models. In: Proc LREC-COLING; 2024. Available from: https://aclanthology.org/2024.lrec-main.1034/
[19] Rehman S, et al. A novel deep learning-based approach with hyperparameter selection using grey wolf optimiza-tion. PeerJ Comput Sci. 2025;11. Available from: https://doi.org/10.7717/peerj-cs.3160
[20] Banerjee A, et al. A comprehensive survey on transformer-based machine translation. ACM Comput Surv. 2025. Available from: https://doi.org/10.1145/3773076
[21] Vayadande, K., Mishra, A., Bodhe, Y. et al. Synergizing graph embeddings and metaheuristic optimization for fraudulent review detection. Int. J. Mach. Learn. & Cyber. 17, 138 (2026). https://doi.org/10.1007/s13042-025-02838-z.
[22] Vayadande, K., Bodhe, Y., Patil, A., Jadhav, A.N., Mokashi, D.K., Mane, S.C., Alone, D.A. and Patel, P.A. (2026). Text Generation, Classification, and Optimization Using Tokenization in NLP. In Driving Innovation by Dynamic Optimization (eds A. Dey, S.N. Mohanty, R. Kumar and T.R. Ku-mar). https://doi.org/10.1002/9781394315727.ch4
[23] Vayadande, K., Bodhe, Y., Patil, A., Jadhav, A.N., Mokashi, D.K., Mane, S.C., Mane, R.M. and Jadhav, R.V. (2026). Hybrid Optimization Techniques for Ayurvedic Medicine Recommendation. In Driving Innovation by Dynamic Optimization (eds A. Dey, S.N. Mohanty, R. Kumar and T.R. Ku-mar). https://doi.org/10.1002/9781394315727.ch15
[24] Vayadande, K., Bodhe, Y., Patil, A., Jadhav, A.N., Mokashi, D.K., Mane, S.C., Patel, P.A. and Chougale, R.K. (2026). Efficient Book Genre Classification Using NLP and Optimization. In Driving Innovation by Dynamic Optimization (eds A. Dey, S.N. Mohanty, R. Kumar and T.R. Ku-mar). https://doi.org/10.1002/9781394315727.ch21
[25] Vayadande, K., Bodhe, Y., Patil, A., Jadhav, A.N., Mokashi, D.K., Mane, S.C., Chougule, V. and Chougale, R.K. (2026). Optimization Techniques in AI-Driven Cybersecurity. In Driving Innovation by Dynamic Optimization (eds A. Dey, S.N. Mohanty, R. Kumar and T.R. Ku-mar). https://doi.org/10.1002/9781394315727.ch20
[26] Kuldeep Vayadande; Komal Sunil Munde; Amol A. Bhosle; Aparna R. Sawant; Ashutosh M. Kulkarni, "Software Engineering in Machine Learning Applications," in How Machine Learning is Innovating Today's World: A Concise Technical Guide , Wiley, 2024, pp.159-172, doi: 10.1002/9781394214167.ch12.
[27] Vayadande, K. et al. (2026). Phishing Detection Using Random Forest-Based Weighted Bootstrap Sampling, LASSO and Feature Selection. In: Saha, A.K., Sharma, H., Prasad, M., Gupta, P.K. (eds) In-telligent Vision and Computing. ICIVC 2025. Lecture Notes in Networks and Systems, vol 1711. Springer, Cham. https://doi.org/10.1007/978-3-032-10667-4_33
[28] Vayadande, K., Garje, R., Dhoot, H., Ghodake, S., Ingle, P., Shaikh, I. (2026). Enhancing Disaster Re-silience Through Machine Learning-Based Preparedness and Awareness Strategies. In: Verma, O.P., Wang, L., Kumar, R., Yadav, A., Rout, R.K. (eds) Machine Intelligence for Research and Innovations. MAiTRI 2024. Lecture Notes in Networks and Systems, vol 1515. Springer, Singapore. https://doi.org/10.1007/978-981-96-8799-2_35.
[29] Vayadande, K., Kulkarni, A.M., Yadav, G.B., Kumar, R. and Sawant, A.R. (2024). A Comprehensive Analysis of Various Tokenization Techniques and Sequence-to-Sequence Model in Natural Language Processing. In How Machine Learning is Innovating Today's World (eds A. Dey, S. Nayak, R. Kumar and S.N. Mohanty). https://doi.org/10.1002/9781394214167.ch1
[30] Vayadande, K., Bailke, P.A., Dombale, A.B., Dange, V.R. and Kulkarni, A.M. (2024). Converting Pseudo Code to Code. In How Machine Learning is Innovating Today's World (eds A. Dey, S. Nayak, R. Kumar and S.N. Mohanty). https://doi.org/10.1002/9781394214167.ch6
[31] K. Vayadande, T. Adsare, T. Dharmik, N. Agrawal, A. Patil and S. Zod, "Cyclone Intensity Estimation on INSAT 3D IR Imagery Using Deep Learning," 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), Uttarakhand, India, 2023, pp. 592-599, doi: 10.1109/ICIDCA56705.2023.10099964.
[32] K. Vayadande, A. Pujari, V. Verma, A. Yeole, S. Zawar and Z. Jamadar, "Scaling Up Video-to-Audio Conversion: A High-Performance Architecture Approach," 2023 International Conference on Recent Advances in Science and Engineering Technology (ICRASET), B G NAGARA, India, 2023, pp. 1-7, doi: 10.1109/ICRASET59632.2023.10420239.
[33] K. Vayadande, R. Shinde, S. Bende, H. Sathe, S. Walunj and S. Jha, "Specialized Large Language Models for Hindi Medical Natural Language Processing: A Clinical Entity in a Multi-Modal Frame-work Recognition and Semantic Understanding," 2025 IEEE 5th International Conference on ICT in Business Industry & Government (ICTBIG), Indore, Madhya Pradesh, India, India, 2025, pp. 1-8, doi: 10.1109/ICTBIG68706.2025.11323575.
[34] K. Vayadande, M. Pawar, T. Patil, V. Patil, S. Pawar and P. Wankhade, "Problem-X: A Catalyst for Social Problem Solving and Digital Inclusion," 2023 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2023, pp. 334-341, doi: 10.1109/ICECA58529.2023.10395817.

[35] Kuldeep Vayadande; Chaitanya B. Pednekar; Priya Anup Khune; Vinay Sudhir Prabhavalkar; Varsha R. Dange, "GPT-3- and DALL-E-Powered Applications," in How Machine Learning is Innovating To-day's World: A Concise Technical Guide , Wiley, 2024, pp.329-341, doi: 10.1002/9781394214167.ch19.
[36] . Vayadande, P. Sangle, K. Agrawal, A. Naik, A. Mulla and A. Khare, "Conversion of Ambiguous Grammar to Unambiguous Grammar using Parse Tree," 2023 International Conference on Inventive Computation Technologies (ICICT), Lalitpur, Nepal, 2023, pp. 1039-1046, doi: 10.1109/ICICT57646.2023.10134096.
[37] Kuldeep Vayadande; Preeti A. Bailke; Lokesh Sheshrao Khedekar; R. Kumar; Varsha R. Dange, "Mood Detection Using Tokenization," in How Machine Learning is Innovating Today's World: A Concise Technical Guide , Wiley, 2024, pp.47-56, doi: 10.1002/9781394214167.ch5.
[38] Kuldeep Vayadande; Preeti A. Bailke; Vikas Janu Nandeshwar; R. Kumar; Varsha R. Dange, "A Com-parative Study of Different Techniques of Text-to-SQL Query Converter," in How Machine Learning is Innovating Today's World: A Concise Technical Guide , Wiley, 2024, pp.353-365, doi: 10.1002/9781394214167.ch21.
[39] K. Vayadande, T. Adsare, N. Agrawal, T. Dharmik, A. Patil and S. Zod, "LipReadNet: A Deep Learning Approach to Lip Reading," 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), Dharwad, India, 2023, pp. 1-6, doi: 10.1109/ICAISC58445.2023.10200426.
[40] K. Vayadande, P. Narkhede, S. Nikam, N. Punde, S. Hukare and R. Thakur, "Facial Emotion Based Song Recommendation System," 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 2023, pp. 240-248, doi: 10.1109/CISES58720.2023.10183606.
[41] K. Vayadande, U. Shaikh, R. Ner, S. Patil, O. Nimase and T. Shinde, "Mood Detection and Emoji Clas-sification using Tokenization and Convolutional Neural Network," 2023 7th International Confer-ence on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2023, pp. 653-663, doi: 10.1109/ICICCS56967.2023.10142472.