The International Innovations Journal of Applied Science (IIJAS) Vol. 2, No.2, 15-07-2025
https://doi.org/10.61856/c1232843

Scientific Events Gate

The International Innovations Journal of Applied Science

Journal homepage: https://iijas.eventsgate.org/iijas

ISSN: 3009-1853 (Online)

IIJAS

Classification Users' Sentiments About the Threads Application Using Machine and Deep Learning

Shaima Mahdi Orebi [1], Asmaa Mohsin Naser [2]

[1] Department of Information Technology, University of Thi Qar, Iraq
[2] Department of Computer Science, University of Thi Qar, Iraq

ARTICLE INFO

Article history:
Received 10 Aug. 2025
Revised 4 Sep. 2025,
Accepted 8 Sep. 2025,
Available online 15 Sep. 2025

Keywords:
Natural Language Processing
Sentiment Analysis
Threads Apps
Deep Learning
Recurrent Neural Networks

ABSTRACT

This study focused on textual sentiment analysis on social media platforms, which is an important tool for understanding public opinion trends, user behavior, and their comments about a specific product or topic. These sentiments or opinions can be analyzed into positive, negative, and neutral. The aim of the research was to classify the opinions and sentiments of Twitter users regarding a Meta application, Thread, which was launched in 2023. Thread is a rapidly growing application that generates massive amounts of textual comments. Therefore, analyzing these comments provides valuable insights into users' emotional engagement with the content. The study applied Recurrent neural networks (LSTM ,BILSTM ,GRU), leveraging their superior ability to handle unstructured text and recognize subtle patterns in natural language. To achieve this, a dataset was collected from data warehouse largest (Kaggle) , which contains user reviews of the application. The results showed acceptable classification accuracy compared to machine learning algorithms, according to the evaluation metrics used (confusion matrix, F1 score, precision, accuracy, and recall). This study contributes to the field of natural language processing by providing insights into users' experiences with the Thread app. It also opens the paves the way to future applications, including predicting user trends and improving the digital content experience.

1. Introduction

Recent years have witnessed rapid development in the field of social media applications, as companies seek to innovate or produce new tools to enhance user engagement and meet their needs. Meta launched the Thread app in 2023 (Zhang et al., 2024).

Meta launched it as an attempt to offer a more focused text-based communication experience alongside its popular Instagram platform (Kambayo et al., 2025). Threads is a new platform that allows users to post and share their thoughts in text threads, similar to the tweets on Twitter, but with integration with Instagram accounts, allowing users to easily navigate and share without having to create a separate account(Khairunnas et al., 2025).

The widespread use of the internet has led to an increase in information. Social media plays a key role in the exchange of information and topics (Lestari & Ambarwati, 2024).

One such product is the recently launched Thread app, which has gained widespread popularity among users. which is considered a small blog from the main Instagram application. You must have an Instagram account in order to install Thread, is primarily concerned with political news, unlike Instagram, which is concerned with daily life(Alam & Abdi, 2025). Using the Thread app generates comments and responses from users of the app. These opinions and comments can be used as an indicator to measure user acceptance levels of the app, as the information may include criticism, suggestions, comments, or other information (Suhaimi & Lestari,

Corresponding author.
E-mail address: shiama_mahdi@utq.edu.iq

https://doi.org/10.61856/c1232843

2024). Given the advancements in technology in our current era, users have been able to express their feelings and opinions on social media sites on various topics. The challenge lies in developing effective tools capable of classifying and analyzing these feelings and opinions (Varadharajan et al., 2025).

This research aims to collect opinions and feedback from Thread users. This data was collected through data warehouses. Three types of deep learning models were then compared to determine which yielded the best results. User sentiment toward Thread was studied and analyzed using deep learning techniques, by analyzing text comments. The project aims to build a model capable of accurately classifying sentiment and explore the impact of using deep models compared to traditional text analysis methods. Used in this study three types of deep learning models(Dang et al., 2021), especially recurrent neural networks, which are LSTM (Long Short-Term Memory) ((Srinivas et al., 2021). BILSTM(Bidirectional Long Short-Term Memory) (Dong et al., 2020) ,GRU(Gated Recurrent Unit) (Sachin et al., 2020), where the models are trained on text data of Thread app reviews compared to traditional machine learning models, which are (Decision Tree, Random Forest) (Guia et al., 2019; Madyatmadja et al., 2024). In this study, we chose recurrent neural networks (RNN), a specialized type of neural network designed to handle sequential data. Unlike traditional neural networks, which assume that inputs are independent of each other, RNNs possess a memory that enables them to use information from previous time steps to process current inputs. Their simplicity lies in the recurrent loop principle, where the output of a hidden layer is passed to the same layer as part of the inputs in the next time step. This gives them the ability to understand context and dependencies between sequential elements, such as words in a sentence, making them ideal for natural language processing tasks(Sunagar et al., 2024).

The significance of this study lies in its use of deep learning techniques to classify sentiment in texts, which is crucial in an era of increasing digital content on social media platforms and websites. Understanding user sentiment helps organizations make more accurate marketing and development decisions, and enhances the user experience in customer service applications, recommendations, and sentiment analysis. It contributes to enhancing our understanding of how artificial intelligence models work with natural language and emotional expressions on rapidly evolving digital platforms.

2. Methodology

To classify or analyze the opinions and reviews of Thread app users by using deep learning tools , we follow the following steps :

Step1 : Datasets Collection

In this step, which is considered the first step in the sentiment analysis phase, which is collecting the data or data set that we will use in the study, we used a text data set for the opinions and feelings of tweeters on Twitter about the Thread application, and the name of the data set was (37000_reviews_of_thread_app) .

This data set contains the following columns(,source,review_id,user_name,review_title,review_description,rating,thumbs_up,review_date,developer_response,developer_response_date,appVersion,laguage_code,country_code) . Only two columns were used in the study, namely (review_description , rating) .The number of data was (36943) distributed according to the following percentages (30.5%) very negative and (5.7%) negative and (8.0%) Neutral and (9.7%) positive and (46.0%) very positive . the language is tweeters' comments is (English). The first column is the review description or the tweeter's opinion on Twitter. The second column is the rating, where the ratings range from 1 to 5. Opinion Ratings: Reviews are tagged with accompanying star ratings and sentiment ratings (e.g., positive, very positive, neutral, negative, very negative) to facilitate sentiment analysis and polarity classification tasks. By observing the ratios, we notice that the percentage of neutral comments is small compared to the positive and negative

https://doi.org/10.61856/c1232843

ones, so this category was neglected and the positive was combined with the very positive ones, in addition to the negative and very negative ones. Thus, two categories of data were obtained, which are (positive (1) and negative (0)).

Step2: Pre-processing

This step begins by taking data and analyzing it using a Python library. First, we remove duplicate data or rows. Data preprocessing is an important step in all fields, particularly in sentiment analysis, due to its impact on the results and accuracy of models and algorithms. If data preprocessing is performed correctly and accurately, it will increase the accuracy of the model and produce good, satisfactory results. This stage begins by converting the raw data in the first step into a form that is easy for algorithms to understand, which helps reduce time and makes it easier and faster for algorithms. The pre-processing process in this study includes several steps, including :

- Clean data by deleting duplicate rows.
- Convert capital letters to lowercase.
- Remove stop words.
- Text Tokenization
- Lemmatization of words by returning the word to its origin.

In addition to Remove special characters, question marks, and hyphens and removing unnecessary data such as emojis, links, punctuation marks and numbers. All steps are implemented using the Python library(NLTK). Figure 1 Preprocessing steps. .

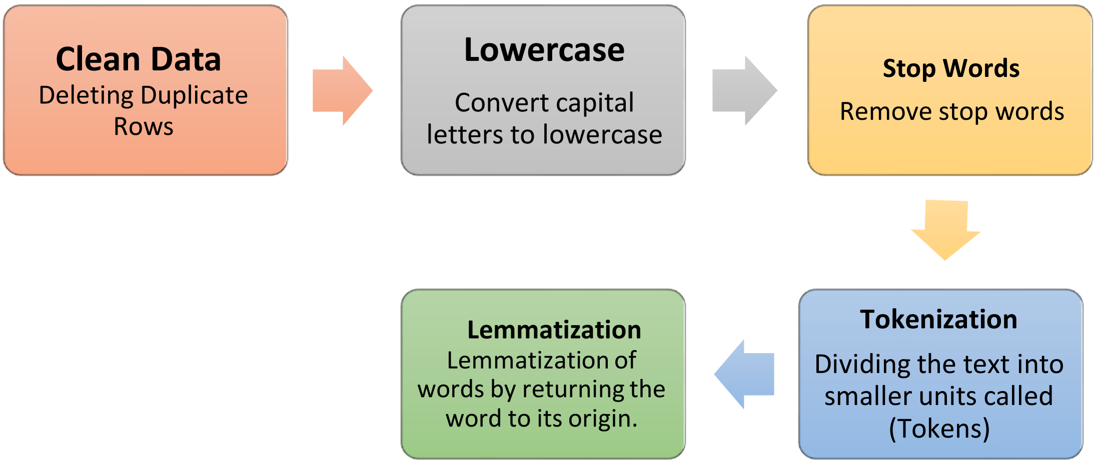

Figure 1. Preprocessing Steps

Step 3: Data Partitioning

In this step, the data is divided into two sets, namely training and testing (80:20). Using (train_test_split), the model is trained on the training set only, while the testing set is used to evaluate the accuracy of the model on previously unseen data, to ensure that the model is not biased and to demonstrate its ability to generalize.

Step 4: Text Tokenization and Sequence Padding .

After the pre-processing stage and converting the texts into an easier format, in order to deal with this data, the texts must be converted into numbers so that the classifier can understand and analyze the words. The texts are converted into numbers using the Keras library in Python, where they are converted into vectors to create a matrix. Word Embedding This method is the most effective in deep learning. Each word is converted into a vector of numbers, where each vector represents the meaning and context of the word. Words with similar meanings are close together in the vector space. This step is important, as it processes raw texts to convert

https://doi.org/10.61856/c1232843

them into digital or numerical representations suitable for use and input into deep learning models. It involves several steps, including segmenting or dividing the text into words or smaller units, with each word extracted as an independent element. Numerical encoding replaces each word with an Numerical representing it in a specific dictionary. Sequence length standardization (Sequence Padding) ensures that all sequences are of the same length to provide uniform input for neural networks. These sequences of numbers are then fed into the embedding layer, which takes the numbers and converts them into vectors that carry semantic information and relationships between words. The embedding layer uses the Keras library, which is able to make the model learn during training the numeric representations of words, taking into account the context and meaning, so that words that are similar in meaning have close numeric representations. Parameters that are adjusted when converting texts to numbers:

1- Maximum sentence length
   Because neural networks need a constant length.
2- Vocabulary size
   Number of words to be kept in the dictionary
3- Digital representation type
   Word embedded diminutions
   Each word is represented by a vector.
4- Dealing with unknown words.
   Where these words are given a special symbol.
5- Use of padding before or after a sentence

Step 5: Model Architecture

A sequential model consisting of multiple layers was built for text processing. This model was chosen for its ability to efficiently process sequential data.The embedding Layer: This is the first layer of the model and its function is to convert each word in the text into a dense digital vector. This representation enables the model to understand the semantic relationships between words. This layer consists of the following parameters:(Vocabulary size contains the total number of unique words recognized by the model), (Vector dimensions contains the size or length of the vector that will be used to represent each word), (Input sequence length contains the maximum number of words that each text example in the input can contain). This layer is the core of the model, handling the numeric sequences generated by the embedding layer. Three different models were tested in this study, each representing a different recurrent structure in the second layer:GRU Model: This model used the GRU layer to process the sequence. GRU is known for its computational efficiency and ability to capture long-term dependencies using gating mechanisms that control the flow of information.LSTM Model: This model used the LSTM layer. LSTM differs from GRU by having an additional state cell, making it highly effective at solving the vanishing gradient problem and retaining information from the beginning of the sequence.BiLSTM Model: This model used the Bidirectional (LSTM) layer. This is the most advanced architecture, processing the sequence twice: once from front to back and once from back to front. This enables the model to understand the full context of words, which can lead to improved performance on text classification tasks.. This is the last layer in the model and is responsible for producing the final prediction. The activation function (sigmoid) is used because it is suitable for binary classification tasks.

Step 6: Model Evaluation

Measuring the model's performance on new data (test data) to determine the model's ability to generalize and make correct predictions. The following metrics were used to evaluate the performance of each model:Accuracy: Expresses the percentage of correct predictions out of the total predictions.Precision: Measures the quality of the model's positive predictions.Recall: Measures the model's

https://doi.org/10.61856/c1232843

ability to find all positive instances in the data.F1-Score: Represents the harmonic mean of precision and recall and is used as a balanced measure of performance, especially in cases of data imbalance.In addition to these metrics, a confusion matrix was used to visualize the classification results. It clearly shows the number of correctly classified instances versus those misclassified, helping to identify the type of errors (false positives or false negatives).

3. Results and discussion

3.1 Data Distribution:

By analysing the data set used for user reviews of the Thread application, you notice its distribution as in the Figure 2 where the highest percentage of positive reviews and the lowest percentage of neutral reviews. The data were balanced to prevent models from being biased towards one category over another.

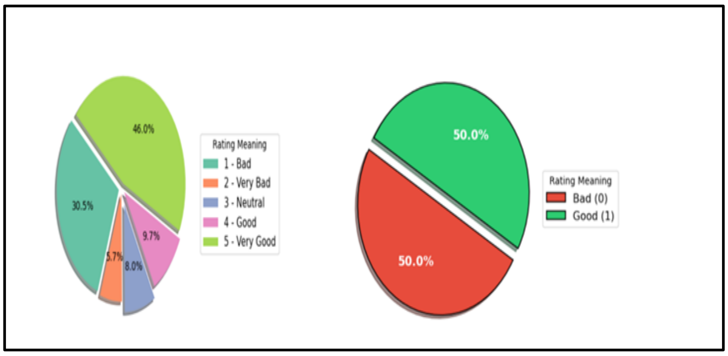

Figure 2. Dataset Distributed

3.2 Word Cloud Analysis:

By analyzing and drawing the word cloud, we see words such as (love, good, great, wonderful) in the positive word cloud, and conversely, we see words (bad, not good, poor, etc.) in the negative word cloud.

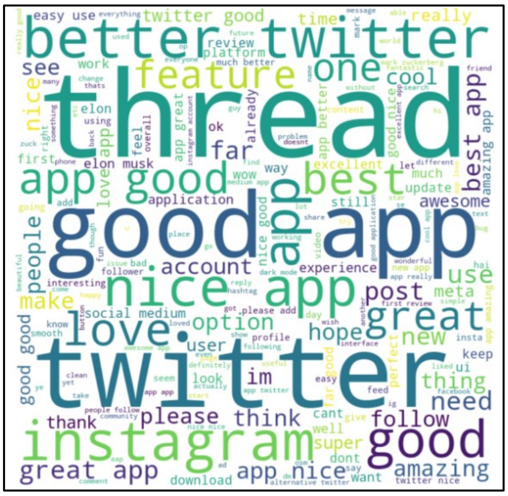

Figure 3. Word Cloud Good Sentiment

Figure 4. Word Cloud Bad Sentiment

3.3 Results Analysis and Performance Evaluation:

This section provides a detailed analysis of the performance of the models trained on text classification tasks around Thread reviews using performance metrics .Table 1 showing the performance metrics for each of the recurrent networks. Where the values of each measure (Accuracy, Precision, Recall, F1-Score) are displayed for each of the models used in this study.

Table 1: Performance Metrics

Model	Accuracy	Precision	Recall	F1-Score
LSTM	0.8860	0.8589	**0.9237**	0.8901
BILSTM	0.8890	0.8666	0.9196	0.8923
GRU	0.8945	0.8749	0.9206	0.8972

Analysis of the table results shows that model (GRU) outperforms other models in all main metrics. The model achieved the highest value in (0.8972) for (F1-Score). Indicates that the model achieves an ideal balance between precision and recall. In addition, the model (LSTM) achieved the highest value (0.9237) for (Recall). This indicates its better ability to identify actual positive cases.

The Figure 5 shows the confusion matrix for a recurrent neural network (RGU), where the rows represent the actual values and the columns represent the values predicted by the model. The upper left part contains (3576) the number of instances that actually belong to class (0) and were correctly classified as negative. The upper right part contains (542) cases that actually belong to the negative class (0), but the model incorrectly classified them as positive class (1).

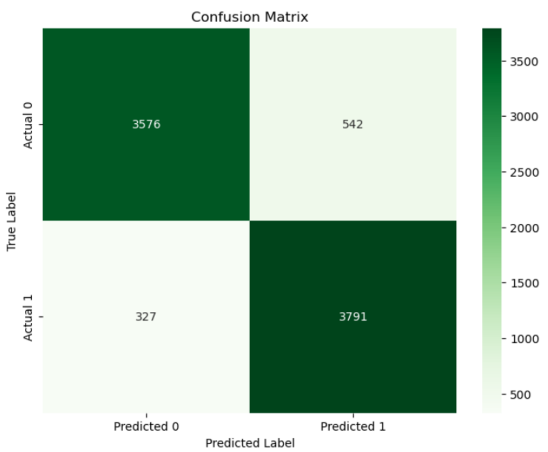

Figure 5. Confusion Matrix for GRU

As for the lower part of the matrix, the left part contains (327), which is the number of cases that actually belong to the positive category (1), but the model incorrectly classified them as negative (0). As for the lower right part, it contains (3791), the number of cases that actually belong to the positive category (1) and were correctly classified as positive.

3.4 Compare Between Models:

In this section, the performance of recurrent neural networks such as (LSTM ,BI LSTM ,GRU) is compared with traditional machine learning models such as (Decision Tree, Random Forest) for the purpose of classifying data on Thread app reviews. The scheme shown in Figure 6.Recurrent neural networks are able to capture contextual sequential patterns in data, making them ideal for text classification tasks. While machine learning models often rely on flat representations of data such as bag of words or (TF-IDF) and fail to capture word order. This superior performance is attributed to the ability of recurrent neural networks to process sequential data efficiently. Unlike traditional machine learning algorithms that lose the contextual order of words, the (GRU) model can understand the context of texts, which leads to more accurate representation and more accurate predictions. Although some previous studies on sentiment analysis of text data from the Thread app relied on traditional machine learning models and achieved results similar to this research in terms of sentiment classification accuracy, the methodology of this research relies on deep learning, specifically recurrent neural networks. Most previous studies used data other than English. Therefore, the use of deep learning not only aims to improve numerical performance, but also to build more generalizable and flexible models, especially when moving to new datasets or different application environments. Therefore, even if the numerical results are similar, the approach adopted in this research provides a more robust and sustainable technical foundation compared to studies that relied on traditional methods.

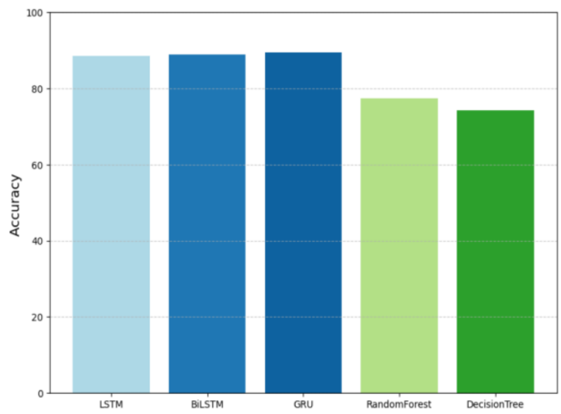

Figure 6. Model Accuracy Comparison

4. Conclusions

This study represents a serious attempt to explore the broad potential of deep learning techniques in the field of sentiment analysis. In this research, we review theoretical frameworks that explain the nature of emotions. Recurrent neural networks (LSTM, BILSTM, GRU) are ideal for complex tasks such as text classification and generally outperform traditional machine learning models in capturing complex patterns and contexts. We also discuss how advanced computational models can understand and analyze this complex phenomenon from digital data.Our findings demonstrate the significant effectiveness of deep learning in building models capable of accurately and efficiently classifying emotions. A deeper understanding of the emotions underlying our digital interactions opens the door to innovative applications that enhance communication, improve services, and provide unprecedented support in fields such as mental health and marketing. Despite these successes, this study acknowledges ongoing challenges, particularly in dealing with the complexities of emotional expression such as sarcasm or cultural nuances. This study confirms that sentiment analysis using deep learning is not just a passing research trend, but a fundamental field with tremendous potential to shape our understanding of the digital world around us and create smarter, more empathetic systems that truly respond to our emotional needs.

References

Alam, N., & Abdi, M. (n.d.). *Sentiment Analysis of Threads Application Users Using Support Vector Machine Method.*

Dang, C. N., Moreno-García, M. N., & De La Prieta, F. (2021). Hybrid Deep Learning Models for Sentiment Analysis. *Complexity*, *2021*(1), 9986920. https://doi.org/10.1155/2021/9986920

Dong, Y., Fu, Y., Wang, L., Chen, Y., Dong, Y., & Li, J. (2020). A Sentiment Analysis Method of Capsule Network Based on BiLSTM. *IEEE Access*, *8*, 37014–37020. https://doi.org/10.1109/ACCESS.2020.2973711

Guia, M., Silva, R., & Bernardino, J. (2019). Comparison of Naïve Bayes, Support Vector Machine, Decision Trees and Random Forest on Sentiment Analysis: *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 525–531. https://doi.org/10.5220/0008364105250531

Kambayo, A. N., Berutu, S. S., & Nshimiyimana, A. (n.d.). *Aspect-Based Sentiment Analysis Using Latent Dirichlet Allocation (LDA) and DistilBERT on Threads App Reviews*.

Khairunnas, R., Pagua, J. A., Fitriya, G., & Ruldeviyani, Y. (2025). User sentiment dynamics in social media: A comparative analysis of X and Threads. *IAES International Journal of Artificial Intelligence (IJ-AI)*, *14*(1), 447. https://doi.org/10.11591/ijai.v14.i1.pp447-456

Lestari, D. W., & Ambarwati, R. (2024). Social Network Analysis: Understanding User Behavior in Threads. *Kontigensi : Jurnal Ilmiah Manajemen*, *12*(1), 115–124. https://doi.org/10.56457/jimk.v12i1.505

Madyatmadja, E. D., Candra, H., Nathaniel, J., Jonathan, M. R., & Rudy. (2024). Sentiment Analysis on

User Reviews of Threads Applications in Indonesia. *Journal Européen Des Systèmes Automatisés*, *57*(4), 1165–1171. https://doi.org/10.18280/jesa.570423

Sachin, S., Tripathi, A., Mahajan, N., Aggarwal, S., & Nagrath, P. (2020). Sentiment Analysis Using Gated Recurrent Neural Networks. *SN Computer Science*, *1*(2), 74. https://doi.org/10.1007/s42979-020-0076-y

Srinivas, A. C. M. V., Satyanarayana, Ch., Divakar, Ch., & Sirisha, K. P. (2021). Sentiment Analysis using Neural Network and LSTM. *IOP Conference Series: Materials Science and Engineering*, *1074*(1), 012007. https://doi.org/10.1088/1757-899X/1074/1/012007

Student, Department of Computer Engineering, JSPM's Rajarshi Shahu College of Engineering. Tathawade (Maharashtra), India., Ghumade*, Ms. T. G., Deshmukh, Mrs. R., & Department of Computer Engineering, JSPM's Rajarshi Shahu College of Engineering. Tathawade (Maharashtra), India. (2019). A Document Classification using NLP and Recurrent Neural Network. *International Journal of Engineering and Advanced Technology*, *8*(6), 633–636. https://doi.org/10.35940/ijeat.F8087.088619

Suhaimi, N. B. A., & Lestari, M. (2024). Sentiment Analysis of Tiktok App Reviews on Google Play using Several Machine Learning Methods. *International Journal of Global Operations Research*, *5*(4), 275–287. https://doi.org/10.47194/ijgor.v5i4.343

Sunagar, P., Sowmya, B. J., Pruthviraja, D., Supreeth, S., Mathew, J., Rohith, S., & Shruthi, G. (2024). Hybrid RNN Based Text Classification Model for Unstructured Data. *SN Computer Science*, *5*(6), 726. https://doi.org/10.1007/s42979-024-03091-x

Varadharajan, V., Smith, N., Kalla, D., Samaah, F., & Mandala, V. (2025). Deep Learning-Based Sentiment Analysis: Enhancing IMDb Review Classification with LSTM Models. *Universal Journal of Computer Sciences and Communications*, *4*(1), 1–14. https://doi.org/10.31586/ujcsc.2025.1249

Zhang, P., He, Y., Haq, E.-U., He, J., & Tyson, G. (2024). *The Emergence of Threads: The Birth of a New Social Network* (No. arXiv:2406.19277). arXiv. https://doi.org/10.48550/arXiv.2406.19277