README

August 20, 2024

Contents

1 Introduction

Welcome to the README file for the code for *A framework for timely and accessible long-term forecasting of shale gas production based on time series pattern matching* submitted to the *International Journal of Forecasting*. The rest of this file is composed of three sections. Section 2 will provide an overview of the reproduction process. Section 3 will describe each file in detail. Finally, Section 4 lists the dependencies and discusses known issues that could harm the reproduction process and our ways to deal with them. Please feel free to contact us if there are any problems.

2 Computational times

Although the original data is not publicly available, the computational time is a good reference for users who want to try the code. With the original data, the complete reproduction of our work requires around a month on a regular desk computer. In specific, our work includes 10 data splits and 6 methods. The execution time for the first data split is discussed in Section 3.7 of the manuscript and a summary table is quoted here:

Table 1: Computational times.

Device	Proposed	Proposed-NC	ANN	ARIMA-R	ARIMA-P	Naive
A	54min	1897min	59min	58min	72s	15s
B	41min	1350min	19min	37min	49s	10s

In the table above, Device A refers to a desk computer with an Intel i5-10500 CPU and 16GB RAM and Device B refers to a desk computer with an Intel i7-11700F CPU, 64GB RAM, and an NVIDIA RTX 3080 Ti GPU. The other nine splits would require longer execution times since all splits are ordered by the average length of the test wells in ascending order. Therefore, we estimate the execution time for each part of the article is as follows:

(1) (a) Target: The comparison of the proposed method with default hyperparameters and other methods (Section 3.2).

   (b) Content: Executing all 6 methods on 10 splits.

   (c) Estimated time: 15-20 days.

(2) (a) Target: The sensitivity analysis of the proposed method: Part I (Section 3.3).

   (b) Content: Executing the proposed method with 12 new hyperparameter settings on 10 splits.

   (c) Estimated time: 4-6 days, given the results from (1).

(3) (a) Target: The sensitivity analysis of the proposed method: Part II (Section 3.4).

   (b) Content: Executing the proposed method with 13 new hyperparameter settings and the naive method with 1 new hyperparameter setting on 10 splits.

   (c) Estimated time: 4-6 days.

(4) (a) Target: Constructing the prediction interval of the proposed method (Section 3.5).

   (b) Content: Executing the proposed method with 9 new clustering random seeds on 10 splits.

   (c) Estimated time: 3-5 days, given the results from (1).

(5) (a) Target: Result summarizing and statistical inference.

   (b) Estimated time: Each part requires at most 1 hour, given the results from (1) to (4).

The detailed process for the reproduction of each part of the article is introduced in the next section.

3 Data and code description

3.1 Overview

All result figures (Figures 5 to 15 in the manuscript and Figures 1 to 15 in the supplementary file) can be directly produced by the code files. Table 2 in the manuscript can be directly read from the output inside TestDataStat.ipynb. Supp. Table 1 can be directly read from the output inside ConfirmANNStructure.ipynb. Other result tables (Tables 3 and 5) and in-text results can also be read from the output inside Jupyter Notebooks or accessed through raw results. Detailed instructions are given in the corresponding Jupyter Notebooks. The following figure depicts the relationships between each file. The arrows inside this figure indicate the order of execution. In this figure, red rectangles stand for method files, blue rectangles stand for post-processing files, green rectangles stand for extra files, purple rectangles stand for final results, and black rectangles stand for data folders and extra folders. In the descriptions below, "Reproduction" is the first-layer directory for all files for this reproduction, including the original files provided and the intermediate and result files to be produced during reproduction.

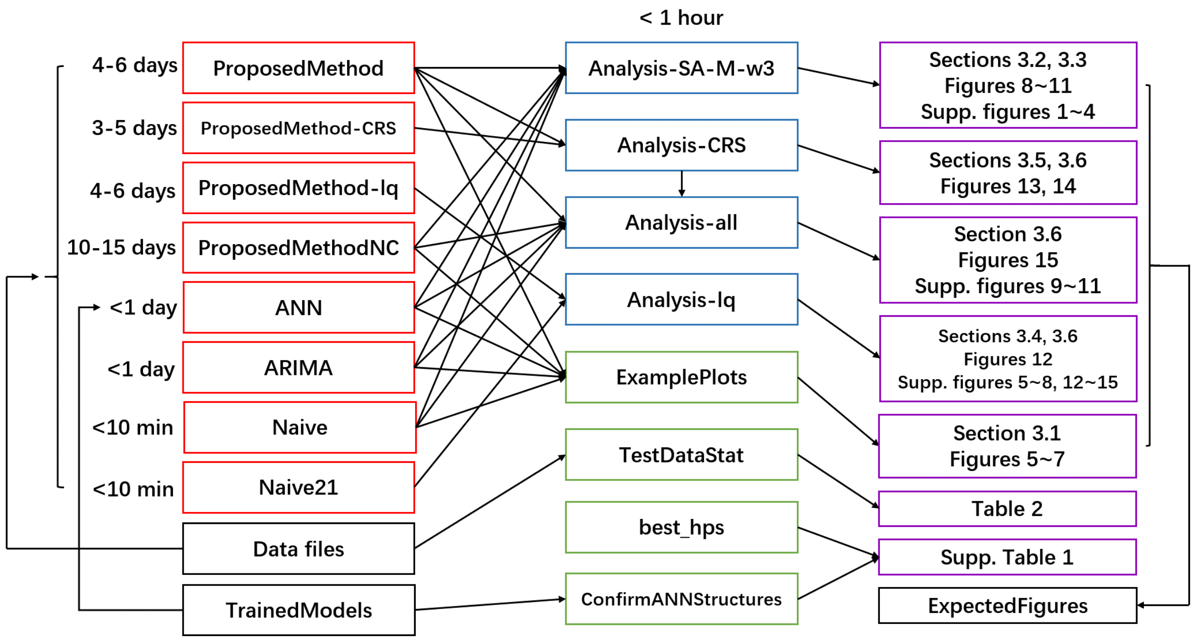

Figure 1: Summary of the code files.

3.2 Data folders

There are 3 data folders. Inside each folder, the data files have names in the form of "PXXWX.xlsx". For example, "P01W1.xlsx" stands for the data of the first well in the first pad. Each data file contains 3 columns. t is the daily production time; Q is the daily gas yield; Mark is a string with the value "reopening" (the first day of a production stage) or "production" (the other days of a production stage). The contents of each folder are:

(1) Data21: Production stages longer than 21 days.

(2) Data14: Production stages longer than 14 days.

(3) Data7: Production stages longer than 7 days.

Please note that the data folders are not publicly available. You can transform your own data to the form described above to try the code. The file names must be in the same format. The column names must be the same.

3.3 Method files

There are 8 method files in .ipynb format. They are:

(1) (a) Name: ProposedMethod.ipynb

(b) Content: This file produces the raw data for the proposed method with different M and w_3 values. $M = 200$ and $w_3 = 1.1$ corresponds to the default setting. The other values correspond to the sensitivity analysis regarding M and w_3.

(c) Result directory: /Reproduction/Results/SA/

(d) Estimated time: This file requires executing the proposed method with 14 different parameter settings on 10 splits, hence 4 to 6 days.

(e) Related sections in the manuscript: Sections 3.2, 3.3, and 3.5.

(2) (a) Name: ProposedMethod-CRS.ipynb

(b) Content: This file produces the raw data for the proposed method with 9 new clustering random seeds.

(c) Result directory: /Reproduction/Results/CRS/

(d) Estimated time: This file requires executing the proposed method with 9 different clustering seeds on 10 splits, hence 3 to 5 days.

(e) Related sections in the manuscript: Section 3.5.

(3) (a) Name: ProposedMethod-lq.ipynb

(b) Content: This file produces the raw data for the proposed method with different l_q settings.

(c) Result directory: /Reproduction/Results/lq/

(d) Estimated time: This file requires executing the proposed method with 13 different parameter settings on 10 splits, hence 4 to 6 days.

(e) Related sections in the manuscript: Section 3.4.

(4) (a) Name: ProposedMethodNC.ipynb

(b) Content: This file produces the raw data for ProposedNC (the proposed method without clustering).

(c) Result directory: /Reproduction/Results/ProposedNC/

(d) Estimated time: This file requires executing ProposedNC on 10 splits, hence 10 to 15 days.

(e) Related sections in the manuscript: Section 3.2.

(5) (a) Name: ANN.ipynb

(b) Content: This file produces the raw data for the ANN method.

(c) Result directory: /Reproduction/Results/ANN/

(d) Estimated time: This file requires executing ANN on 10 splits, hence within 1 day.

(e) Related sections in the manuscript: Section 3.2

(6) (a) Name: ARIMA.ipynb

(b) Content: This file produces the raw data for ARIMA-R and ARIMA-P.

(c) Result directory: /Reproduction/Results/ARIMAR/ and
/Reproduction/Results/ARIMAP/

(d) Estimated time: This file requires executing ARIMA-R and ARIMA-P on 10 splits, hence within 1 day.

(e) Related sections in the manuscript: Section 3.2.

(7) (a) Name: Naive.ipynb

(b) Content: This file produces the raw data for the naive method with $l_q = 14$.

(c) Result directory: /Reproduction/Results/Naive/

(d) Estimated time: This file requires executing the naive method on 10 splits, hence within 10 minutes.

(e) Related sections in the manuscript: Section 3.2.

(8) (a) Name: Naive21.ipynb

(b) Content: This file produces the raw data for the naive method with $l_q = 21$.

(c) Result directory: /Reproduction/Results/lq/

(d) Estimated time: This file requires executing the naive method on 10 splits, hence within 10 minutes.

(e) Related sections in the manuscript: Section 3.4.

3.4 Post-processing files

There are 4 post-processing files in .ipynb format with all outputs not cleared. They are:

(1) (a) Name: Analysis-SA-M-w3.ipynb

(b) Content: This file produces i) the raw post-processing results for the proposed method with different M and w_3 values and the baseline methods; ii) Figures 8 to 11 in the manuscript and Figures 1 to 4 in the supplementary file.

(c) Result directory: /Reproduction/Analysis-SA/ and /Reproduction/Figures/

(d) Requirements: This file requires results from ProposedMethod.ipynb, ProposedMethodNC.ipynb, ANN.ipynb, ARIMA.ipynb, and Naive.ipynb to execute.

(e) Related sections in the manuscript: Sections 3.2 and 3.3.

(2) (a) Name: Analysis-CRS.ipynb

(b) Content: This file produces i) the raw post-processing results for the proposed method with different clustering random seeds; ii) the ensemble of the proposed method with different clustering seeds; iii) Figures 13 and 14 in the manuscript; iv) statistical inference regarding this part of the study.

(c) Result directory: /Reproduction/Analysis-CRS/ and /Reproduction/Figures/

(d) Requirements: This file requires results from ProposedMethod.ipynb, ProposedMethod-CRS.ipynb, ProposedMethodNC.ipynb, ANN.ipynb, ARIMA.ipynb, and Naive.ipynb to execute.

(e) Related sections in the manuscript: Sections 3.5 and 3.6.

(3) (a) Name: Analysis-all.ipynb

(b) Content: This file produces i) the raw post-processing results for the proposed method with different M and w_3 values, the baseline methods, and the ensemble of the proposed method with different clustering seeds; ii) Figure 15 in the manuscript and Figures 9 to 11 in the supplementary file; iii) statistical inference regarding this part of the study.

(c) Result directory: /Reproduction/Analysis-all/ and /Reproduction/Figures/

(d) Requirements: This file requires results from ProposedMethod.ipynb, ProposedMethod-CRS.ipynb, ProposedMethodNC.ipynb, ANN.ipynb, ARIMA.ipynb, Naive.ipynb, and Analysis-CRS.ipynb to execute.

(e) Related sections in the manuscript: Section 3.6.

(4) (a) Name: Analysis-lq.ipynb

(b) Content: This file produces i) the raw post-processing results for the proposed method with different l_q values and the navie method with $l_q = 21$; ii) Figure 12 in the manuscript, Figures 5 to 8 and Figures 12 to 15 in the supplementary file; iii) statistical inference regarding this part of the study.

(c) Result directory: /Reproduction/Analysis-lq/ and /Reproduction/Figures/

(d) Requirements: This file requires results from ProposedMethod-lq.ipynb and Naive21.ipynb to execute.

(e) Related sections in the manuscript: Sections 3.4 and 3.6.

3.5 Extra files

There are 3 extra files in .ipynb format with all outputs not cleared and 1 extra file in .csv format. They are:

(1) (a) Name: ExamplePlots.ipynb

(b) Content: This file produces Figures 5 to 7 in the manuscript.

(c) Result directory: /Reproduction/Figures/

(d) Requirements: This file requires results from ProposedMethod.ipynb, ProposedMethodNC.ipynb, ANN.ipynb, ARIMA.ipynb, and Naive.ipynb to execute.

(e) Related sections in the manuscript: Section 3.1.

(2) (a) Name: TestDataStat.ipynb

(b) Content: This file produces data for Table 2 in the manuscript.

(c) Result directory: The results can be directly read from the output inside the Jupyter Notebook.

(3) (a) Name: best_hps.csv

(b) Content: This file contains the raw data for Table 1 in the supplementary file. These data are connected to the trained ANN models in /Reproduction/TrainedModels/. Please have a look at the description of TrainedModels in Section 3.6 for details.

(4) (a) Name: ConfirmANNStructure.ipynb

(b) Content: This file contains the confirmation of the content in /Reproduction/best_hps.csv using trained models from /Reproduction/TrainedModels/.

(c) Result directory: The results can be directly read from the output inside the Jupyter Notebook.

3.6 Extra folders

There are 3 extra folders. They are:

(1) (a) Name: Figures

(b) Content: This is an empty folder to store graphical results generated by code files.

(2) (a) Name: TrainedModels

(b) Content: This folder contains trained ANN models. Please have a look at Section 4 in README for details.

(3) (a) Name: ExpectedFigures

(b) Content: This folder contains expected graphical results. Please note that the following figures are in their revised form: Figure 12, Supplementary Figures 5 to 8 and 12 to 15.

3.7 Further notes on file directory

As mentioned before, ”Reproduction” is the first-layer directory for all files for this reproduction. This includes the original files provided and the intermediate and result files to be produced during reproduction. Specifically, in the unzipped reproduction file, you will find the following content under the folder ”Reproduction”: data folders, method files, post-processing files, extra files, and extra folders. Then, during execution, the following folders will be created under the folder ”Reproduction”: Results, Analysis-SA, Analysis-CRS, Analysis-all, and Analysis-lq.

With the original data, more than 10,000 intermediate files would be created during the reproduction process. Normally, all new directories and files will be automatically created during reproduction. However, since the reproduction process is expected to be very long, unexpected interruptions may happen during code execution (system update, power loss, ...). This makes the manual management of the directories and files also important. In the case of interruption, please take the following steps to reset the progress for the interrupted file:

(1) Locate the interruption point in the Jupyter Notebook.

(2) Search for all strings called `os.makedirs` that appears before that interruption point. This can be done by pressing ctrl+F in VSCode.

(3) Delete all folders created by `os.makedirs` found in the previous step.

(4) Restart the notebook and execute again.

4 Dependencies and known issues

4.1 Dependencies

Our code is purely based on Python 3.8.13. We advise you to use Windows 10 or 11 as the code has not been checked on other operating systems. The following two packages must be installed in the specified version for reproduction:

Table 2: Dependencies 1.

Name	Version
`seaborn`	0.13.0
`tensorflow` (GPU version)	2.9.0

The versions of the following packages do not have known impacts on the results. We are just listing the versions we used.

Table 3: Dependencies 2.

Name	Version
`os`	N/A
`datetime`	N/A
`pandas`	1.4.2
`csv`	1.0
`matplotlib`	3.5.1
`numpy`	1.22.3
`scipy`	1.7.3
`sklearn`	1.1.1
`kneed`	0.8.3
`pmdarima`	2.0.3
`keras-tuner`	1.4.7

4.2 Known issues

Three devices have participated in our work. The first two are Device A and Device B mentioned in Section 2 of README. The third one, Device C, is a desk computer with an Intel i7-13700F CPU, 64GB RAM, and a NVIDIA RTX 4070 Ti GPU. The majority of our work was completed on Device B, and we have partly tested the reproducibility of our work on all three devices. During the tests, we have found three issues that could affect the reproduction of our work. The issues and our countermeasures are discussed below.

Firstly, `pmdarima` may produce non-reproducible results on different machines. Please have a look at the GitHub issue different results across machines #52 for details. We found that enlarging the argument `maxiter` mitigates, but cannot solve this issue. This corresponds to `maxiter=100000` in ARIMA.ipynb. Nevertheless, this issue happens in only a few production stages and cannot bring notable differences in the result.

Secondly, `numpy` may produce non-reproducible results on different machines due to a variety of implicit reasons. There are many discussions on this topic, such as StackOverflow questions Floating point math in python / numpy not reproducible across machines and Same Python code, same data, different results on different machines. For our code and data, we found Device B and C would give identical results. However, Device A may give different results for the proposed method and the proposed method without clustering. Similar to the first issue, this issue also happens in only a few production stages and cannot bring notable differences in the result.

Thirdly, `tensorflow` is expected to produce different results on different machines. Therefore, we have directly provided the trained ANN models for the reproduction of the results. We have tested that, with `tensorflow 2.9.0`, trained models would produce identical results on Device B and C (both have a dedicated GPU), but not on Device A (does not have a dedicated GPU). Therefore, we believe the ANN results are reproducible using trained models on a computer with a dedicated GPU and `tensorflow 2.9.0`.

Finally, although this is not an issue, it should be noted that the data dots contained in strip plots may have different positions for the same plotting data. This is just a matter of displaying rather than differences in the results. The following two figures provide an example:

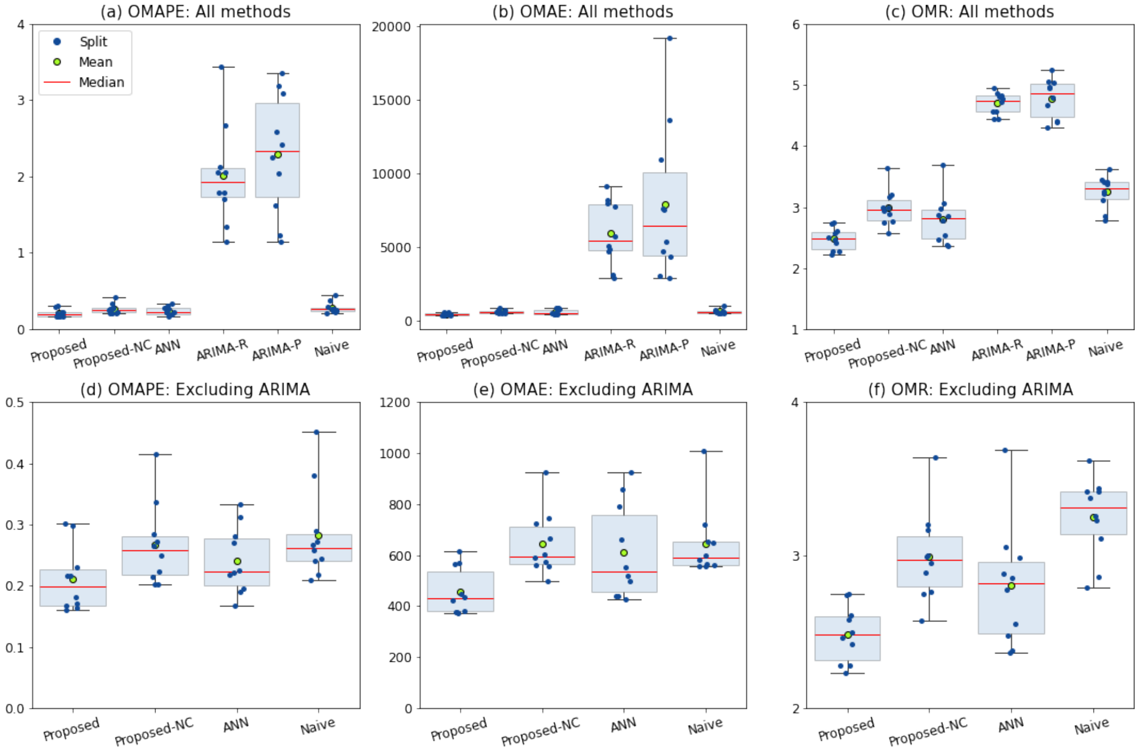

Figure 2: Figure 8.

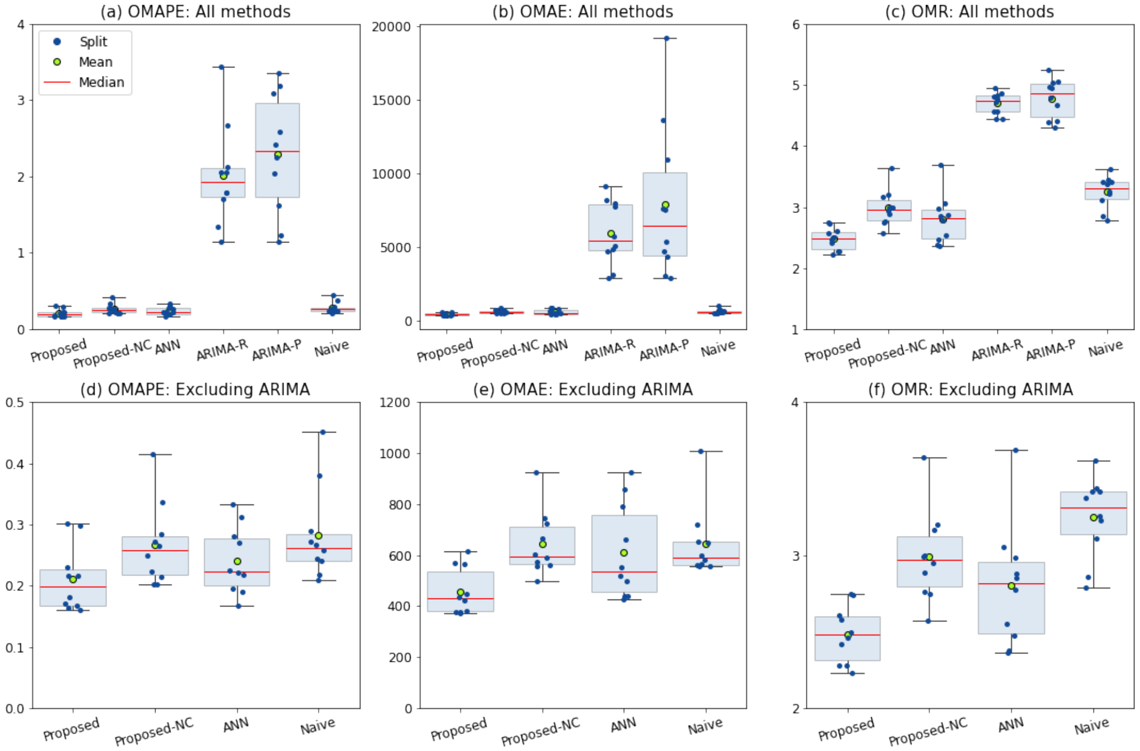

Figure 3: Another Figure 8 with the same data.