https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

A Reproducible Research Framework for Wind Energy

Daragh O'Connor[1], Hoa Xuan Nguyen[1], Juan Manuel González Sopeña[1], Vikram Pakrashi[2], and Bidisha Ghosh[1]

[1]QUANT Group, Department of Civil, Structural and Environmental Engineering, Trinity College Dublin, Dublin, Ireland.
[2]UCD Centre for Mechanics, Dynamical Systems and Risk Laboratory, School of Mechanical and Materials Engineering, University College Dublin, Dublin, Ireland.

Correspondence: Bidisha Ghosh (bghosh@tcd.ie)

Abstract. With the rapid growth of wind energy installations and the corresponding increase in data-driven research outputs, establishing robust reproducible research (RR) practices has become essential to ensure reliability, transparency, and long-term scientific value. Wind energy research is characterised by high data volumes, complex computational workflows, and increasing reliance on advanced modelling and machine learning techniques, all of which amplify reproducibility challenges. This paper examines the current state of reproducible research practices within the wind energy domain and identifies key gaps that limit computational reproducibility and replicability. In response, it proposes a structured, sector-specific reproducible research workflow designed to improve transparency, reliability, and ease of replication. The proposed workflow spans three stages; Conceptualisation and Planning, Implementation and Execution, and Dissemination. These emphasise essential components such as systematic data management, code sharing, platform selection, version control, and comprehensive documentation. In addition, the paper introduces a set of Python-based tools and best practices that support reproducibility at each stage of the workflow. A practical case study in wind power forecasting using an open-access dataset and a publicly available GitHub repository is presented to demonstrate the application of the workflow and to highlight common reproducibility challenges, particularly those related to data sharing, preprocessing, and documentation. The results show that adopting structured reproducible research practices enables transparent verification, facilitates independent replication, and enhances the reusability of computational wind energy studies. Collectively, the proposed framework and case study provide actionable guidance to support more reliable, verifiable, and collaborative wind energy research.

1 Introduction

There is a surge in wind energy research, with around 22,200 publications in the field of Engineering Science under the category of "Wind Energy" from the years 2022 to 2025, including 5,658 records in 2025, 6,359 in 2024, 5,084 in 2023, and 5,099 in 2022 Analytics (2026). Given the substantial volume of research in wind energy, it is crucial to establish and adhere to reproducibility standards to maintain the reliability and validity of the findings. Defining and implementing clear practices for reproducibility in wind energy research is essential to advance the field and to ensure robust and trustworthy scientific results.

Reproducibility has been consistently identified as a major challenge across scientific domains, with empirical studies demonstrating substantial variability in researchers' ability to reproduce published results—variability that is often driven by

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

lack of access to data, missing metadata, undocumented preprocessing steps, or inconsistent computational environments Peng (2011); Ioannidis et al. (2009). These challenges are not unique to wind engineering research; similar issues have been widely documented in other data-intensive disciplines. For example, in biomedical and biological sciences, reproducibility failures arising from undocumented workflows, batch effects, and analytical variability have been extensively highlighted Ioannidis (2005). In numerical simulation research, even small differences in software controls and solver implementations can lead to diverging results, underscoring the need for stringent computational controls Hocquet and Wieber (2021). Likewise, the machine-learning community has identified major reproducibility barriers related to dependency management, versioning, random initialisation, and insufficient documentation of training pipelines Semmelrock et al. (2025). Importantly, these challenges highlight a distinction between *experimental reproducibility* (recreating physical conditions or measurements) and *computational reproducibility* (recreating analytical results), a distinction that is particularly relevant to wind engineering.

In wind engineering, reproducibility issues become even more pronounced due to the field's reliance on large, heterogeneous datasets, complex modelling workflows, and algorithms sensitive to factors such as software versioning, preprocessing decisions, hyperparameter tuning, and stochasticity in model training. Without transparent, fully documented, and environment-captured workflows, even small implementation differences can lead to substantial divergence in results. Given the increasing operational and scientific importance of wind-energy forecasting, resource assessment, and control strategies, establishing rigorous, field-specific reproducibility standards—spanning data management, code organisation, environment capture, and end-to-end workflow documentation—is essential for ensuring robust and trustworthy scientific outcomes and enabling cumulative progress.

The concept of reproducible research (RR) differs across disciplines, with each field adopting specific methods and tools to overcome reproducibility challenges. In the transportation sector, general guidelines emphasise the importance of code sharing Zheng (2021). Figure 1 presents a simplified recreation of a workflow proposed by Zheng (2021) consisting of five stages: study design, data collection and input, optional data processing, data analysis and modelling, and documentation and dissemination. To ensure complete reproducibility, every stage of the workflow must be programmed using a computational language for analysis and a documentation language for narratives, then combined into a dynamic document. In numerical simulation research, issues related to computational reproducibility arise from differences in hardware and software environments Lenhard and Küster (2019). Institutional libraries play a key role in supporting RR by offering infrastructure and services Schmidt et al. (2024). In medical education, RR is increasingly integrated into teaching through standardised workflows and best practices in statistical analysis Meid (2021). In fields such as ecology and biology, structured RR workflows emphasise open-source software and detailed documentation Alston and Rick (2021). Therefore, addressing reproducibility challenges requires a customised framework for each field, using the right tools and approaches Conrad et al. (2024).

Although many publications in wind energy research mention terms such as "reproduced" and "reproducible", they often focus on recreating environmental conditions, experimental setups, and wind loads rather than ensuring the reproducibility of research findings. For example, Perez-Becker et al. (2021) emphasise reproducible aeroelastic load calculations for the design and control of modern wind turbines. Similarly, innovative wind tunnel experiments, such as those described by Frederik et al. (2018), use an open-jet wind tunnel with an active grid to create reproducible turbulent wind conditions. However, these uses

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

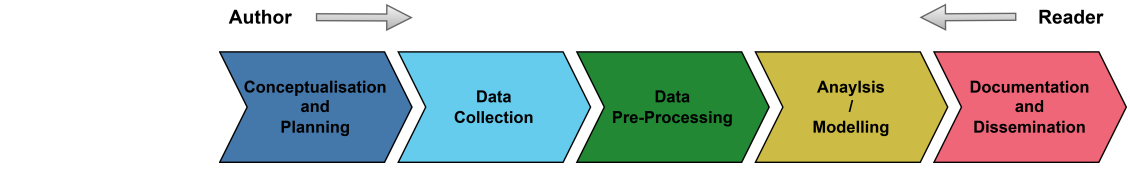

Figure 1. Generalised workflow for reproducible research Zheng (2021).

of "reproducible" primarily address physical consistency rather than the computational reproducibility needed for consistent analytical results.

According to the National Academy of Sciences, Engineering, and Medicine National Academies of Sciences et al. (2019), "reproducibility means computational reproducibility, obtaining consistent computational results using the same input data, computational steps, methods, code, and conditions of analysis", whereas "replicability means obtaining consistent results across studies aimed at answering the same scientific question, each of which has obtained its own data." In the context of wind energy research, this distinction is critical, as confirmed by Pelser et al. (2023), who highlight that the lack of open data and code sharing significantly hampers computational reproducibility in Wind Resource Assessments (WRAs), despite their role in informing energy policy.

These examples illustrate a gap in current wind energy research practices: while physical conditions are often recreated to ensure consistency, computational reproducibility of research findings remains limited. This highlights the need for a more standardised, computationally focused approach to reproducibility in wind energy research, emphasising the sharing of data and code to enable true reproducibility and replicability.

To improve reproducible research practices in wind energy, efforts must be tailored to the specific needs of the field, ensuring that openness and reusability are present throughout the entire research process. The evolving Open Science mandates of the EU European Commission (2025) further reinforce the requirement for transparent data, workflows, and computational environments in publicly funded research. Since openness alone does not guarantee reproducibility, it is crucial to provide detailed documentation that includes data, software, workflows, and computing environments Chen et al. (2019); Conrad et al. (2024) to foster the reproducibility of research.

This paper has three primary contributions. First, it proposes the first sector-specific RR workflow tailored to the unique challenges of wind energy research. Second, it provides a practical, fully documented example of wind energy prediction using a real open-access dataset and a publicly shared GitHub repository, allowing end-to-end replication. Third, it translates foundational RR concepts—such as those from Peng, the National Academies, and Knuth's literate programming—into actionable, domain-specific guidance for wind energy researchers.

A framework and tools for applying reproducible research practices at each stage of the research process in wind energy in presented in Section 2 and Section 3. Section 4 provides practical examples of reproducible research in wind energy and offers a template to facilitate sharing and collaboration using GitHub.

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

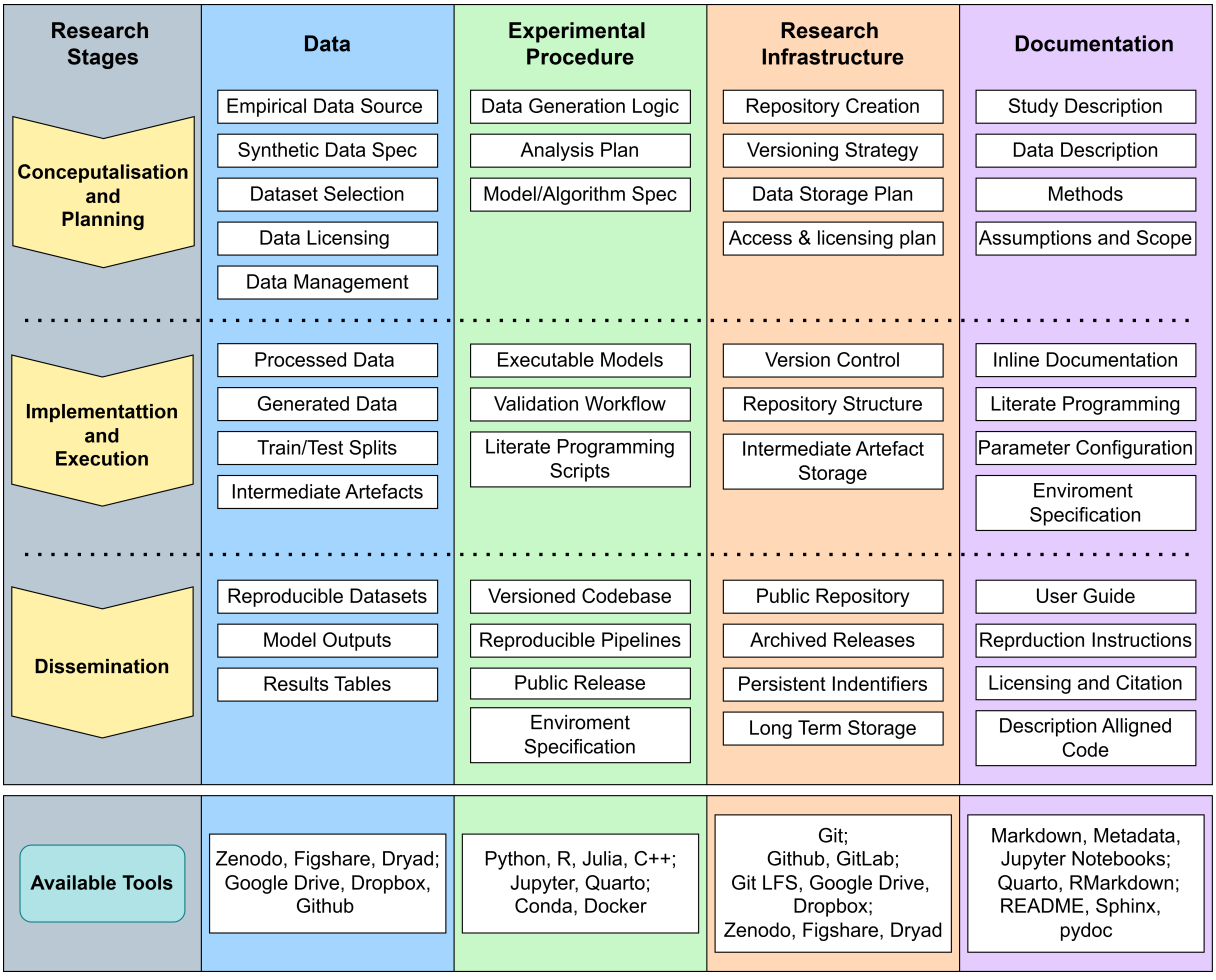

Figure 2. A recommended workflow for research outputs in wind energy research.

2 Proposed workflow

The workflow that follows offers a methodical way to apply repeatable procedures at every phase of wind energy research. The RR workflow, depicted in Figure 2, demonstrates the application of reproducible practices in three crucial phases of wind energy research: conceptualisation and planning, implementation and execution, and dissemination.

The first stage of a reproducible workflow, conceptualisation and planning, aims to guarantee transparency and accessibility through accurate documentation. Depending on the type of research being undertaken, a data management plan adapted to either synthetic or real data should be used. For projects involving the generation or collection of real data, data imputation, manipulation, and cleaning procedures must be documented in a systematic manner. In cases where synthetic data is used, the data generation process, including model parameters, should be recorded. Projects using open access datasets require proper attribution and a clear description of the dataset features used in the study. These steps should be documented in accompanying

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

README or metadata files. Once the data management strategy is established, platforms for data sharing and version control should be selected to ensure that all project materials are securely stored and easily accessible for future use.

As the research moves to the Implementation and Execution phase, the focus shifts to ensuring that all computational steps are clearly documented as they evolve. If raw data is cleaned or validated at this stage, the associated documentation, models, and code should be organised in the same manner as in the Conceptualisation and Planning stage. The development of numerical models should be documented in detail, along with the calibration and validation methods used for analysis. Effective documentation relies on good programming practices such as modularity, where code is divided into self contained units, and literate programming techniques that combine code with explanatory text in a single document Knuth (1984). These practices are particularly important when complex numerical models, validation methods, and result processing steps are involved, as transparency in model parameters and hyperparameters is essential for reproducibility. In addition, detailed records of the computational environment, including software versions, dependencies, and numerical parameters, should be maintained to allow the analysis to be repeated under the same conditions. Continuous documentation at this stage ensures that challenges encountered during the project are traceable and can be replicated.

In the Dissemination phase, research outputs are shared with the wider scientific community. This includes making datasets, code, models, and results available through reliable sharing platforms that remain accessible after the research is completed. Alongside executable code, the core principles of the algorithms and methods used in the study should be clearly described in the published paper, with this description aligning closely with the code implementation in an understandable manner. This alignment supports transparency and enables readers to follow the logic of the methods without relying solely on the code. All shared outputs must be well organised and documented to establish a clear link between the raw data, the analysis code, and the research conclusions. This traceability allows future researchers to reproduce the results and understand the full research process. Providing access to the final versions of code and data on a structured sharing platform, together with clear instructions for running the analysis under the same settings and environment, supports verification, replication, and reuse of the work.

3 Tools for Reproducible Research in Wind Energy

To establish a RR workflow, it is crucial to adopt open-source programming languages National Academies of Sciences et al. (2019), such as Python, R, and C++ that foster accessibility and provide researchers with essential tools. Over the past two decades, RR tools have been developed, both for specific domains and general use. The tools discussed in this section are primarily related to Python-based wind energy research, providing accessible and transparent solutions.

3.1 Data sharing platforms

In accordance with the principles of Open Data European Commission (2024) and FAIR Jones and Grootveld (2017), data used in research should be accessible to other researchers to support reproducibility and transparency. Several cloud-based platforms offer free or low-cost data storage solutions that address common issues, such as accessibility, version control, and long-term storage.

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

For small datasets, GitHub Github (2020) is ideal due to its version control capabilities and its suitability for organising project files. However, GitHub enforces a file size limit, generally 100 MB per file, so large datasets require alternative storage solutions. In cases where datasets are less than 2GB (GitHub free account, maximum 5GB for GitHub Enterprise Cloud), Git Large File Storage (LFS) GitHub (2024) can be used to manage files beyond the 100 MB limit, allowing GitHub to track changes in large files efficiently. Platforms such as Dropbox Dropbox (2024), Google Drive Google (2024), and Kaggle Kaggle (2024) offer convenient options for storing files that exceed GitHub's storage limits. In case where datasets are particularly large, segmenting files into smaller parts before uploading is recommended to ensure easy access and download by other researchers.

For long-term data storage and citation, platforms such as Zenodo European Organization For Nuclear Research and OpenAIRE (2013), Dryad White et al. (2008) and Figshare Singh (2011) are recommended. Zenodo accommodates larger files and assigns each dataset a unique Digital Object Identifier (DOI), simplifying data referencing in published research. Figshare offers similar long-term storage and supports various file types and sizes. Using reliable data-sharing platforms ensures that datasets are accessible, organised, and citable, which is essential for maintaining research integrity and reproducibility across projects. For further details on reproducible research platforms, including strengths, weaknesses, dependency management, and peer review, see Colom et al. (2019).

3.2 Computational code

Tools that allow users to create easy to follow documents that combine code, data analysis, and narrative text include Jupyter Notebooks Kluyver et al. (2016), R Markdown in RStudio Xie et al. (2018), and Quarto Allaire et al. (2024). A key component of the strategy behind these tools is literate programming, a technique developed by Knuth (1984). Code, narrative, and outputs are combined in literate programming to create a comprehensive, independent record of the code process. Because each code section is clearly documented, other researchers can comprehend the context, reasoning, and goal of each analytical step.

Regarding programming languages, open-source options such as Python and R are preferred, as they offer unrestricted access to source code, which facilitates implementation inspection and collaborative development. Further, they integrate well with version control and continuous integration systems. By contrast, MATLAB is a fundamentally proprietary software with closed-source code and licenses required for full use while it remains widely used. Recent developments have allowed MATLAB's basic features to be used via MATLAB Online without a license and also provide better compatibility with GitHub.

Good quality code and commonly understood development techniques are crucial to ensuring reproducibility. Consistent formatting and readability are encouraged by style guidelines like the Python PEP8 style guide van Rossum et al. (2001). Many tools are available to aid in this process. Stylistic rules, correction of mistakes, bug finding, and automatically formatting code sections can be done automatically with formatters (autopep8, black, etc) and linters (flake8, pylint, etc). By ensuring that function inputs and outputs adhere to expected data types, type checking tools like mypy reduce ambiguity in intricate analytical workflows and add an extra degree of dependability. A further step to increase reusability is to enforce a modular code structure, ie. dividing code into clear and intuitive modular sections. Clarity is enhanced by breaking up large portions of the code and including logical directories and clear documentation. Best-practice resources like Wilson et al. (2017), which

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

emphasise modularity and documentation, provide guidance on code organisation for research projects. Literate programming, coding standards, modular structure, linters, formatters, and type checking all work together to provide a strong basis for computational reproducibility in wind energy research.

3.3 Documentation

Documenting each step thoroughly is essential for ensuring that research can be replicated and understood by others, which includes detailing the computational environment, setting up reporting tools, and using structured documentation.

A well-detailed README file is fundamental to project documentation, providing setup instructions, code execution guidelines, and a workflow overview to help users navigate, set up, and replicate the research effectively. The README should provide clear instructions on setting up the computational environment, running the code, and understanding the overall workflow. It can also include descriptions of the directory structure and brief summaries of each file's purpose, guiding users through the replication process step by step Crome (2024); Singers (2024).

Additionally, precise software and dependency management is crucial for reproducibility National Academies of Sciences et al. (2019). The repository should include an environment file that documents the computational environment required to run the code, ensuring consistent results. Python's `pip freeze` or `conda` Anaconda Software Distribution (2016) can generate environment files, such as `requirements.txt` or `environment.yml`, listing all required packages so that others can replicate the analysis under identical conditions. This is especially helpful for complex workflows involving machine learning models or numerical simulations, where minor changes can significantly impact results. Environment files specify the exact versions of software libraries and dependencies, providing a stable setup essential for machine learning algorithms that rely on specific configurations for model training and prediction consistency. For studies involving complex statistical models or machine learning applications, capturing details of the numerical model—such as hyperparameters, model architectures, and overall model structure—is essential for reproducing results. Comprehensive documentation of these components within the environment files and supporting materials ensures that the analysis can be accurately reproduced and validated.

For complex workflows, Docker Boettiger (2015); Alston and Rick (2021) is a useful tool, as it allows researchers to create a portable and isolated environment containing the code and its dependencies. By packaging code and dependencies within Docker containers, researchers can ensure that their analyses are executed in a controlled and consistent environment, regardless of individual system configurations. Docker images, which store the complete setup, can be shared and reused, allowing others to build and run identical containers, thus enabling reproducibility and minimising environment-related discrepancies.

For literate programming and generating dynamic documentation directly within R, RMarkdown Xie et al. (2018) offers a powerful solution. RMarkdown combines text, code, and output in a single document, making it suitable for documenting detailed workflows, generating reports, and maintaining reproducibility across analyses. For Python projects, tools such as Jupyter Notebook, Quarto, Sphinx Brandl (2021), and `pydoc` create formal documentation directly from code, enhancing the accessibility and readability of project documentation.

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

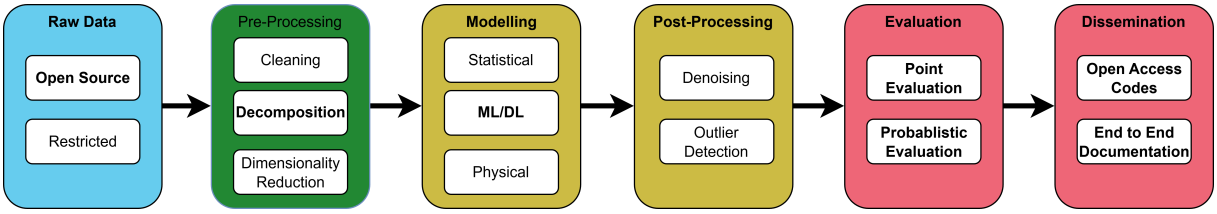

Figure 3. High-level workflow for short-term wind power forecasting used in the case study.

3.4 Version control

Version control tools are essential for collaboration among researchers, research dissemination, and ongoing reader engagement. Git and GitHub Github (2020) are widely used for this purpose. Git allows researchers to track and manage file changes locally, enabling systematic edits, reversions, and updates without compromising previous versions.

Importantly, version control should be adopted from the very beginning of a project, ideally from version 0, so that the full development history is preserved. Early experiments, intermediate decisions, and discarded approaches often contain valuable contextual information that helps future researchers understand why specific methodological choices were made. Documenting the code development reduces the likelihood of repeating ineffective approaches, provides insight into the reasoning behind refinements, and contributes to transparency in the scientific process.

Github is the platform that compliments Git, it allows users to store their updates and publish changes as they go. Gihutb supports clear version tracking, feedback from other users and real time documentation updates. Furtehr more it can be set up in a collaborative structure so that co-authors can easily comment and adjust parts of the code base. GitHub branches also enable researchers to work independently on specific project elements, merging contributions smoothly upon completion while preserving the integrity of the main files. For further information, a comprehensive introductory to version control in collaborative research projects has been publishedChacon and Straub (2014).

4 Reproducible Research: Case Study in Wind Energy

This section presents a practical case study demonstrating how RR principles can be applied to a real-world wind energy application. The case study replicates and extends the methodology of Sopeña et al. (2023) using an openly available dataset from the Kelmarsh wind farm Plumley (2022), replacing the proprietary data used in the original study. The objective is not to re-evaluate the forecasting methodology itself, but to illustrate how transparent data management, structured code organisation, and comprehensive documentation can enable computational reproducibility and facilitate reuse by other researchers.

Figure 3 provides a high-level overview of a reproducible wind power forecasting workflow. The components adopted in this study are highlighted in bold. The figure serves as a conceptual reference for the subsequent planning and implementation stages.

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

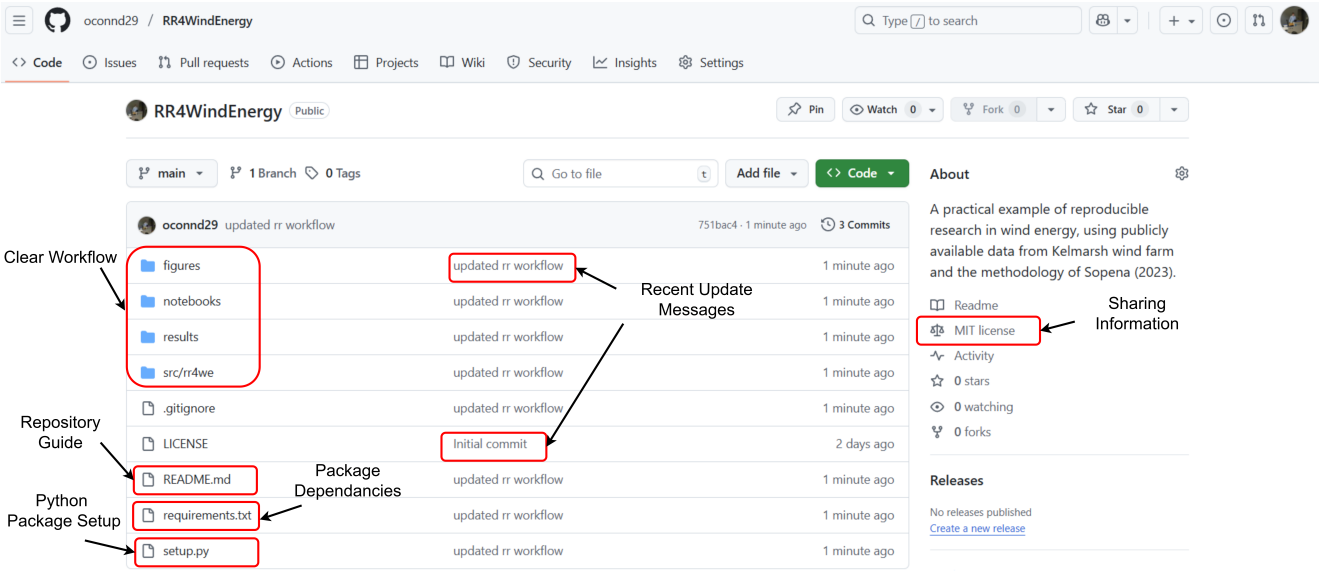

Figure 4. Structure of the RR4WindEnergy GitHub repository used in the case study.O'Connor (2023)

All materials required to reproduce the case study—including code, documentation, and links to the data—are made publicly available via a GitHub repository (https://github.com/oconnd29/RR4WindEnergy). The case study is structured according to the reproducible research workflow introduced in Sections 2 and 3, progressing from conceptualisation and planning, through implementation and execution, to dissemination of results. Each subsection highlights reproducibility considerations specific to that stage, avoiding duplication while maintaining traceability across the full research lifecycle.

4.1 Conceptualisation and planning

The case study is based on an open-access dataset comprising 10-minute resolution SCADA and event data from six Senvion MM92 wind turbines located at the Kelmarsh wind farm Plumley (2022). The dataset spans turbine operation between 2016 and 2023 and includes key variables such as power output, rotor diameter, hub height, and fiscal/grid meter measurements. The temporal resolution and richness of the dataset support detailed analysis of turbine-level and wind farm–level behaviour, making it a suitable open alternative to the proprietary dataset used in Sopeña et al. (2023). The raw data are hosted on Zenodo, which provides persistent identifiers, long-term availability, and versioning, all of which are essential for reproducible research.

At the conceptualisation stage, a clear data management and project organisation strategy is defined to ensure transparency and traceability throughout the study. Data, code, and documentation are organised within a GitHub repository following the structured workflow described in Section 2 and illustrated in Figure 4. The repository is divided into dedicated directories for `data`, `notebooks`, `src`, `figures`, and `results`, enforcing a clear separation between raw inputs, executable analysis, reusable source code, and generated outputs.

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

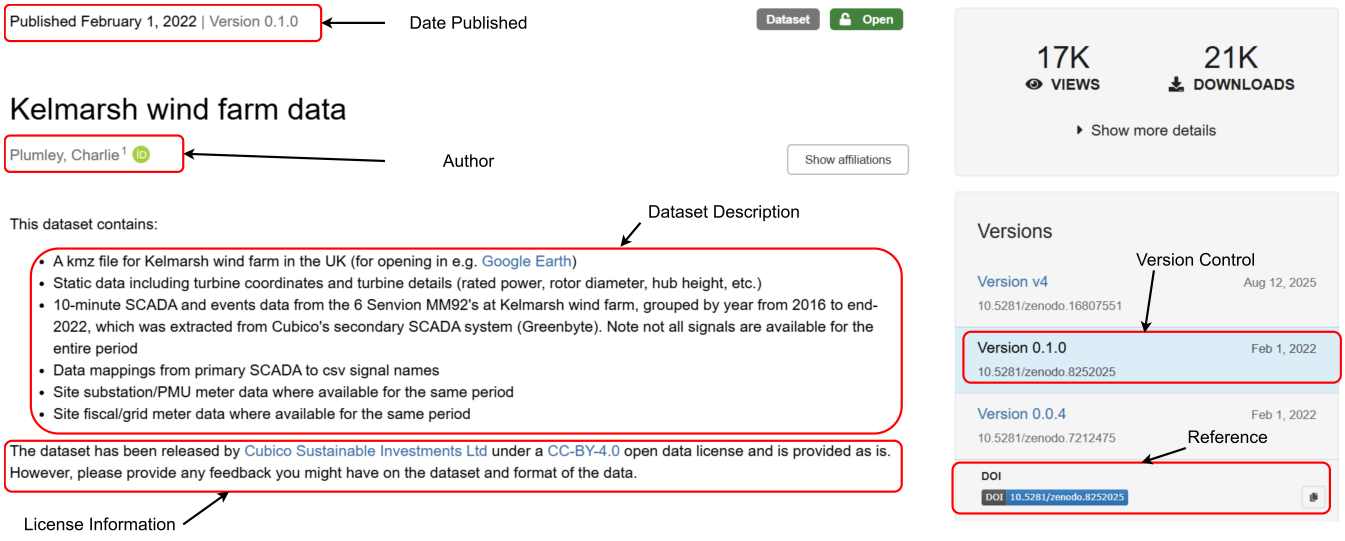

Figure 5. Zenodo landing page for the Kelmarsh wind farm dataset, showing persistent access and metadata.Plumley (2022)

Due to GitHub file-size constraints, raw SCADA data are not stored directly in the repository. Instead, the `data/` directory contains scripts and documentation describing how the dataset is obtained from Zenodo and processed locally, see Figure 5. A `.gitignore` file is used to explicitly exclude raw data, temporary files and environment-specific files from version control, ensuring that the repository remains lightweight.

Reusable utility functions are organised within a Python package in the `src/` directory, structured by workflow stage. This modular design supports code reuse, clarity, and extension, and ensures that analytical logic is separated from execution. Jupyter notebooks corresponding to each major workflow stage are stored in the `notebooks/` directory, providing an executable and narrative-driven interface for the analysis. The role of these notebooks in operationalising the workflow is described in detail in the subsequent implementation and execution subsections.

Project-level documentation is established at this stage through a `README.md` file, which provides an overview of the study objectives, repository structure, data sources, and setup instructions. Computational dependencies are captured in a `requirements.txt` file, which is updated as the project evolves to ensure that the computational environment required for reproduction is explicitly defined. Licensing information is also included to clarify reuse conditions.

By defining repository structure, data provenance, and documentation practices at the outset, this conceptualisation and planning phase establishes the foundation for reproducible implementation, analysis, and dissemination, which are addressed in the following sections.

4.2 Implementation and Execution

This phase implements the reproducible workflow defined during conceptualisation by executing a structured pipeline that transforms raw SCADA data into model-ready inputs and applies a VMD-based wind power forecasting methodology. Figure

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

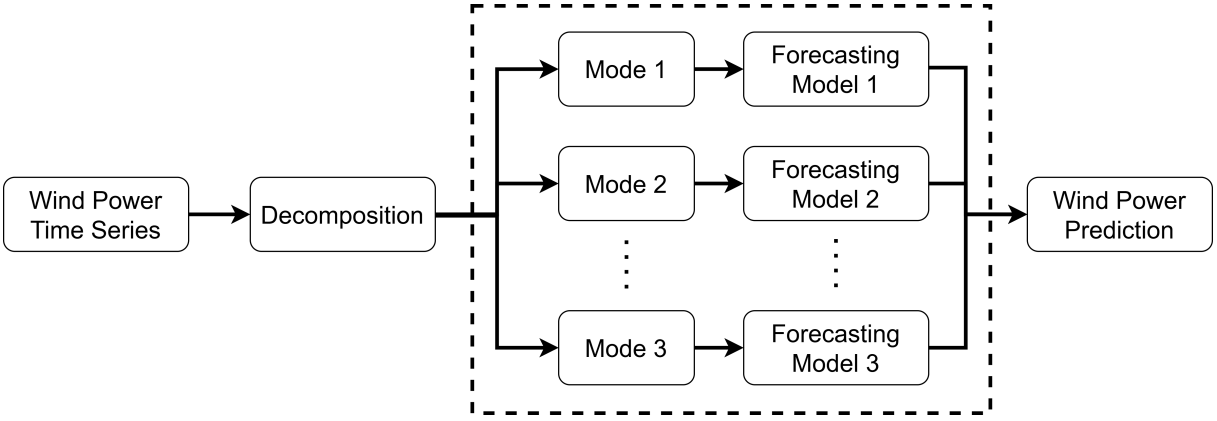

Figure 6. Flowchart for wind power forecasting using decomposition-based hybrid models.

6 illustrates the topology of the prediction procedure. Execution is organised through a sequence of Jupyter notebooks located in the `notebooks/` directory, supported by reusable and modular Python code contained in the installable package `src/rr4we/`. Intermediate files, trained models, and prediction outputs are written to well-defined subdirectories under `data/` and `results/`, ensuring traceability between inputs, processing steps, and outputs.

4.2.1 Data preprocessing

Data preprocessing is implemented in the notebook `notebooks/01_data_preprocessing.ipynb`, which is fully parametrised to ensure transparent and repeatable execution. Key configuration parameters include a global analysis period (`2016-01-01` to `2023-01-01`), the set of turbines considered (`T1-T6`), and the root directory containing the locally stored Kelmarsh SCADA export obtained from Zenodo. These parameters allow the preprocessing workflow to be rerun or adapted without modifying the core logic.

The notebook performs an automated data ingestion step in which all turbine CSV files are discovered by recursively scanning the configured directory for yearly data files. Each file is loaded using a dedicated utility that handles vendor-specific formatting, including skipping non-data header rows, parsing the timestamp column, and assigning it as a datetime index. Turbine identifiers are extracted directly from filenames, and the resulting data are grouped by turbine before being concatenated into continuous time series.

For each turbine, duplicate timestamps are removed and the time index is rounded to a uniform 10-minute resolution. The data are then reindexed onto a complete 10-minute grid spanning the full analysis period, ensuring that all turbines share a common time axis and that missing observations are explicitly represented as `NaN`. The aligned multi-turbine dataset is stored as a dictionary keyed by turbine identifier and serialised to disk as `data/raw/Kelmarsh_SCADA.pkl`, together with basic coverage and completeness statistics. This intermediate artefact provides a reproducible reference point between raw data ingestion and downstream analysis.

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

To prepare the univariate signal used in the forecasting experiments, the serialized dataset is reloaded and a single turbine (by default `T1`) is selected. The active power signal is extracted over the same analysis period and inspected to quantify missing, zero, and negative values. Remaining missing observations are conservatively filled with zeros, yielding a continuous 10-minute power time series suitable for model input. The resulting cleaned signal is saved as `data/processed/power_data_filled.csv`, which serves as the sole input to subsequent modelling notebooks.

The preprocessing notebook is designed to run end-to-end using a single "Run All" execution, ensuring that data ingestion, alignment, cleaning, and persistence are executed consistently across environments. All subsequent forecasting and analysis stages operate exclusively on the processed dataset, enabling a clear separation between data preparation and model execution and supporting traceable, repeatable experiments.

4.2.2 Analysis

The analysis stage implements a Variational Mode Decomposition (VMD) and Feedforward Neural Network (FFNN) forecasting pipeline, following the methodological framework of Sopeña et al. (2023). Core algorithmic components, including VMD routines, neural network definitions, evaluation metrics, and utility functions, are implemented in a modular manner within the `src/rr4we/` package. This separation of reusable source code from executable notebooks supports clarity, reuse, and controlled modification.

VMD is first applied to decompose the wind power time series into intrinsic mode functions (IMFs), which are then used as inputs to the forecasting model. The forecasting task focuses on short-term horizons of 10, 20, and 30 minutes using 10-minute resolution data. Each forecast uses the previous 72 time steps (equivalent to 12 hours) as input, and multi-step prediction is performed using a Multiple-Input Multiple-Output (MIMO) strategy:

$$[\hat{y}_{t+H}, \ldots, \hat{y}_{t+1}] = f(y_t, \ldots, y_{t-d+1}), \tag{1}$$

where H denotes the forecast horizon and d is the number of historical observations used as input. This formulation enables the model to jointly predict the full forecast horizon while preserving temporal dependencies.

The notebook `notebooks/02_generate_vmd_datasets.ipynb` applies VMD to the processed dataset and stores the resulting mode decompositions as structured artefacts under `results/modes/vmd/`. The forecasting notebook `notebooks/03_wind_power_forecast.ipynb` then trains FFNN models using the decomposed signals and saves trained model files (`.h5`) and prediction outputs (`.pickle`) to clearly defined subdirectories under `results/models/` and `results/predictions/`. This explicit organisation enables selective re-execution of individual stages (e.g. retraining models without recomputing VMD), which is a key aspect of computational reproducibility.

All notebooks are written using a literate programming approach, combining executable code with inline documentation and explanatory text. Hyperparameters, architectural choices, and processing assumptions are explicitly recorded within the notebooks and source code, ensuring that model behaviour can be inspected, reproduced, and modified by other researchers. Version control is maintained throughout using Git and GitHub, allowing changes to code and notebooks to be tracked across the development lifecycle.

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

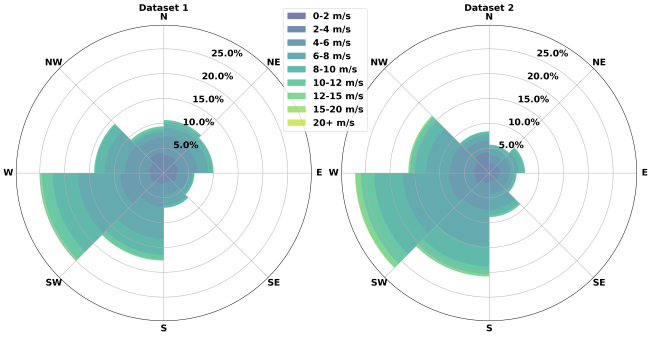

Figure 7. Wind roses for Dataset 1 and Dataset 2 from the Kelmarsh wind farm (Turbine 1).

4.3 Dissemination

4.3.1 Documentation

All materials required to reproduce this case study, including code, documentation, and links to the underlying data, are publicly available through the project's GitHub repository. The final processed dataset used for all experiments is stored in the `data/processed/` directory as `power_data_filled.csv`, providing a stable and well-defined input for downstream analysis.

The repository contains the complete and final set of scripts and notebooks used in this study, organised to ensure that file paths, dependencies, and execution order are consistent across environments. Computational dependencies are specified explicitly in `requirements.txt`, enabling users to recreate the software environment required to run the analysis.

User-facing documentation is provided through the repository's `README.md` file, which outlines the project structure, environment setup, and execution sequence for the notebooks. Each Jupyter notebook is designed to run end-to-end using a single "Run All" execution, allowing users to reproduce the full workflow—from data preprocessing through model training and prediction—without manual intervention. Inline comments and explanatory text are included to support interpretability and reuse.

4.3.2 Results and Comparability

To support qualitative comparison and interpretation, wind roses for Dataset 1 and Dataset 2 are shown in Figure 7, providing contextual information on wind direction and speed distributions associated with the evaluation periods.

Forecasting results obtained using the VMD–FFNN model are illustrated in Figure 8, which shows observed power output alongside point predictions and associated prediction intervals for both datasets at a 10-minute forecasting horizon. The results demonstrate that the reproduced workflow yields predictive behaviour consistent with that reported in Sopeña et al. (2023), despite the use of a different, openly available dataset.

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

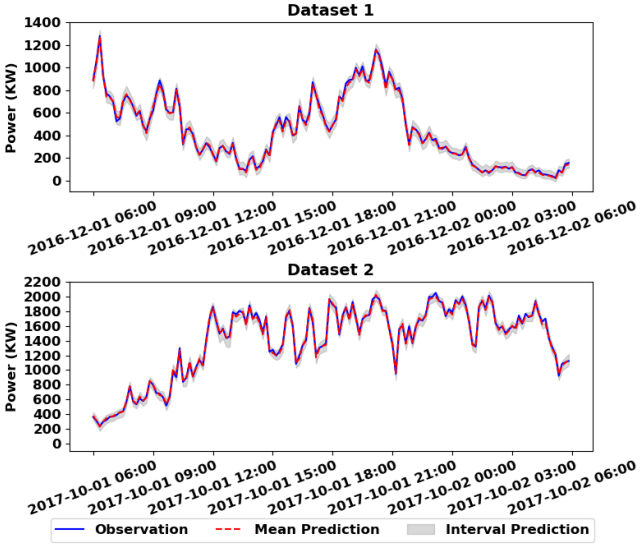

Figure 8. Observed and predicted wind power for Dataset 1 and Dataset 2, with a 10 minute forecasting horizon, using the VMD-FFNN model.

By making the full workflow, data processing steps, and results openly accessible, this case study enables reviewers and other researchers to independently verify the findings, assess methodological consistency, and apply the same reproducible pipeline to alternative wind energy datasets.

5 Conclusions

Reproducible research (RR) in wind energy, particularly in data-driven areas such as wind power forecasting, remains less formally established than in neighbouring computational fields, despite the growing scale, complexity, and societal importance of wind energy studies. This paper addresses this gap by proposing a structured, sector-specific framework for reproducibility and replicability that is tailored to the practical realities of wind energy research.

This paper makes three primary contributions. First, it proposes the first wind energy specific reproducible research workflow, explicitly designed to address challenges unique to the field, including large heterogeneous datasets, complex preprocessing pipelines, and sensitivity to computational environments. Second, it provides a fully documented, end-to-end demonstration using an open-access wind power dataset and a publicly available GitHub repository, enabling direct replication and extension of the presented results. Third, it translates foundational reproducible research principles—such as those articulated by Peng, the National Academies, and Knuth's concept of literate programming—into actionable, domain-specific guidance that can be readily adopted by wind energy researchers.

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

The case study presented in this paper offers several important practical takeaways. It demonstrates that reproducibility is achievable even for complex machine-learning-based wind power forecasting pipelines when data, code, environments, and documentation are systematically organised. The study highlights the importance of preserving intermediate datasets, explicitly documenting preprocessing decisions, recording model hyperparameters, and capturing the computational environment to ensure consistent results across systems. Crucially, the ability to reproduce comparable forecasting performance on an alternative open dataset underscores the role of reproducible workflows in validating model robustness beyond a single proprietary data source.

More broadly, the case study illustrates that reproducibility should not be treated as a final dissemination task, but rather as a continuous process embedded throughout conceptualisation, implementation, and dissemination. Practices such as modular code design, version control from project inception, literate programming, and clear repository structure were shown to significantly lower the barrier for independent verification and reuse.

By adopting the framework and tools outlined in this paper, the wind energy research community can take concrete steps toward strengthening the credibility, transparency, and long-term value of scientific outputs. Improved reproducibility supports more reliable benchmarking of forecasting methods, facilitates collaboration between researchers and practitioners, and enhances confidence in results that increasingly inform operational and policy decisions. As wind energy research continues to expand in scale and impact, embedding reproducible research practices as a standard component of methodological rigor will be essential for ensuring robust, trustworthy, and sustainable scientific progress.

Code and data availability. The code and documentation for the case study are publicly available at the RR4WindEnergy GitHub repository (https://github.com/oconnd29/RR4WindEnergy). The wind farm dataset used in the case study is openly available on Zenodo Plumley (2022). To support computational reproducibility, the software environment is specified in `requirements.txt` within the repository. For long-term archiving, the code should be referenced using a tagged release and (where available) an associated DOI.

Author contributions. D. O'Connor was the primary author, developed the repository structure, and led manuscript preparation. H. Nguyen contributed to workflow design and manuscript preparation. J. M. González Sopeña developed the codebase used in the case study. V. Pakrashi and B. Ghosh supervised the research and contributed to methodology development and manuscript revision. All authors reviewed and approved the final manuscript.

Competing interests. The authors declare that they have no competing interests.

Acknowledgements. The authors acknowledge the use of artificial intelligence tools for limited assistance in correcting grammar and refining phrasing in portions of this work. All substantive content, analysis, and conclusions remain the sole responsibility of the author.

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

References

Allaire, J., Teague, C., Scheidegger, C., Xie, Y., and Dervieux, C.: Quarto, https://doi.org/10.5281/zenodo.5960048, 2024.

Alston, J. M. and Rick, J. A.: A beginner's guide to conducting reproducible research, Bulletin of the Ecological Society of America, 102, 1–14, 2021.

Anaconda Software Distribution: Computer software. Vers. 2-2.4.0. Anaconda, https://anaconda.com, 2016.

Analytics, C.: Web of Science Core Collection, https://www.webofscience.com, accessed: Dec. 09, 2025, 2026.

Boettiger, C.: An introduction to Docker for reproducible research, ACM SIGOPS Operating Systems Review, 49, 71–79, 2015.

Brandl, G.: Sphinx documentation, URL http://sphinx-doc. org/sphinx. pdf, 2021.

Chacon, S. and Straub, B.: Pro git, Springer Nature, 2014.

Chen, X., Dallmeier-Tiessen, S., Dasler, R., Feger, S., Fokianos, P., Gonzalez, J. B., Hirvonsalo, H., Kousidis, D., Lavasa, A., Mele, S., et al.: Open is not enough, Nature Physics, 15, 113–119, 2019.

Colom, M., Kerautret, B., and Krähenbühl, A.: An overview of platforms for reproducible research and augmented publications, in: Reproducible Research in Pattern Recognition: Second International Workshop, RRPR 2018, Beijing, China, August 20, 2018, Revised Selected Papers 2, pp. 25–39, Springer, 2019.

Conrad, T. O., Ferrer, E., Mietchen, D., Pusch, L., Stegmüller, J., and Schubotz, M.: Making Mathematical Research Data FAIR: Pathways to Improved Data Sharing, Scientific Data, 11, 676, 2024.

Crome, J. A.: Build a Better README, in: The Perl and Raku Conference, Las Vegas, Nevada, USA, 2024.

Dropbox, I.: Dropbox, https://www.dropbox.com, cloud storage and file sharing service, 2024.

European Commission: Open Science, https://research-and-innovation.ec.europa.eu/strategy/strategy-2020-2024/our-digital-future/open-science_en, 2024.

European Commission: Open Science Policy, https://research-and-innovation.ec.europa.eu/knowledge-publications-tools-and-data/open-science_en, accessed: Jan. 15, 2026, 2025.

European Organization For Nuclear Research and OpenAIRE: Zenodo, https://doi.org/10.25495/7GXK-RD71, 2013.

Frederik, J., Kröger, L., Gülker, G., and van Wingerden, J.-W.: Data-driven repetitive control: Wind tunnel experiments under turbulent conditions, Control Engineering Practice, 80, 105–115, 2018.

Github: GitHub, https://github.com/, 2020.

GitHub: Git Large File Storage, https://git-lfs.com/, 2024.

Google: Google Drive, https://www.google.com/drive/, cloud storage and file sharing service, 2024.

Hocquet, A. and Wieber, F.: Epistemic issues in computational reproducibility: software as the elephant in the room, European Journal for Philosophy of Science, 11, https://doi.org/10.1007/s13194-021-00362-9, 2021.

Ioannidis, J. P., Allison, D. B., Ball, C. A., Coulibaly, I., Cui, X., Culhane, A. C., Falchi, M., Furlanello, C., Game, L., Jurman, G., et al.: Repeatability of published microarray gene expression analyses, Nature genetics, 41, 149–155, 2009.

Ioannidis, J. P. A.: Why Most Published Research Findings Are False, PLOS Medicine, 2, null, https://doi.org/10.1371/journal.pmed.0020124, 2005.

Jones, S. and Grootveld, M.: How FAIR are your data, URL: https://zenodo. org/records/1065991, 10, 2017.

Kaggle: Kaggle, https://www.kaggle.com/, 2024.

https://doi.org/10.5194/wes-2026-52
Preprint. Discussion started: 31 March 2026

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., and Willing, C.: Jupyter Notebooks – a publishing format for reproducible computational workflows, in: Positioning and Power in Academic Publishing: Players, Agents and Agendas, edited by Loizides, F. and Schmidt, B., pp. 87 – 90, IOS Press, 2016.

Knuth, D. E.: Literate programming, The computer journal, 27, 97–111, 1984.

Lenhard, J. and Küster, U.: Reproducibility and the concept of numerical solution, Minds and Machines, 29, 19–36, 2019.

Meid, A. D.: Teaching reproducible research for medical students and postgraduate pharmaceutical scientists, BMC Research Notes, 14, 445, 2021.

National Academies of Sciences, Policy, Affairs, G., on Research Data, B., Information, on Engineering, D., Sciences, P., on Applied, C., Statistics, T., on Mathematical Sciences, B., et al.: Reproducibility and replicability in science, National Academies Press, 2019.

O'Connor, D.: RR4WindEnergy, https://github.com/oconnd29/RR4WindEnergy, gitHub repository, accessed: 2026-03-25, 2023.

Pelser, T., Weinand, J. M., Kuckertz, P., McKenna, R., Linssen, J., and Stolten, D.: Reviewing accuracy & reproducibility of large-scale wind resource assessments, Advances in Applied Energy, p. 100158, 2023.

Peng, R. D.: Reproducible research in computational science, Science, 334, 1226–1227, 2011.

Perez-Becker, S., Marten, D., Nayeri, C. N., and Paschereit, C. O.: Implementation and Validation of an Advanced Wind Energy Controller in Aero-Servo-Elastic Simulations Using the Lifting Line Free Vortex Wake Model, Energies, 14, 783, 2021.

Plumley, C.: Kelmarsh wind farm data, https://doi.org/10.5281/zenodo.8252025, accessed 1 August 2025, 2022.

Schmidt, B., Chiarelli, A., Loffreda, L., and Sondervan, J.: Emerging roles and responsibilities of libraries in support of reproducible research, 2024.

Semmelrock, H., Ross-Hellauer, T., Kopeinik, S., Theiler, D., Haberl, A., Thalmann, S., and Kowald, D.: Reproducibility in machine-learning-based research: Overview, barriers, and drivers, AI Magazine, 46, e70 002, https://doi.org/https://doi.org/10.1002/aaai.70002, 2025.

Singers, M.: Awesome README, https://github.com/matiassingers/awesome-readme, 2024.

Singh, J.: FigShare, Journal of Pharmacology and Pharmacotherapeutics, 2, 138–138, 2011.

Sopeña, J. M. G., Pakrashi, V., and Ghosh, B.: A benchmarking framework for performance evaluation of statistical wind power forecasting models, Sustainable Energy Technologies and Assessments, 57, 103 246, 2023.

van Rossum, G., Warsaw, B., and Coghlan, A.: PEP 8 – Style Guide for Python Code, https://peps.python.org/pep-0008/, python Enhancement Proposal 8. Status: Active. Created: 05-Jul-2001., 2001.

White, H., Carrier, S., Thompson, A., Greenberg, J., and Scherle, R.: The Dryad Data Repository: A Singapore Framework Metadata Architecture in a DSpace Environment., in: Dublin core conference, pp. 157–162, 2008.

Wilson, G. et al.: Good enough practices in scientific computing, PLOS Computational Biology, https://doi.org/10.1371/journal.pcbi.1005510, 2017.

Xie, Y., Allaire, J., and Grolemund, G.: R Markdown: The Definitive Guide, Chapman and Hall/CRC, Boca Raton, Florida, ISBN 9781138359338, https://bookdown.org/yihui/rmarkdown, 2018.

Zheng, Z.: Reasons, challenges, and some tools for doing reproducible transportation research, Communications in Transportation Research, 1, 100 004, 2021.