Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Seismic-AI: An Ensemble Deep-Learning Platform for Earthquake Risk Classification and Magnitude Prediction

Ziad Tamer Fathi Soltan

ziadtamer756@gmail.com

New Mansoura University

Research Article

Keywords: earthquake risk prediction, ensemble learning, TabNet, SeisBench, STEAD, explainable AI

Posted Date: November 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11816830/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Seismic-AI: An Ensemble Deep-Learning Platform for Earthquake Risk Classification and Magnitude Prediction

Target journal: Earth Science Informatics (Springer Nature) — Research Article.

Status: DRAFT — not submitted.

Author (verified): Ziad Tamer Fathi Soltan, Faculty of Engineering,

New Mansoura University, New Mansoura, Egypt.

Corresponding author: Ziad Tamer Fathi Soltan — ziadtamer756@gmail.com

(confirm before submission; ORCID to be added if available).

Co-authors (on hold): Mahmoud Youssef and Magy El-Benhawy — to be added

only after the author sends their emails, affiliations and roles.

Do not add them to a submission until then.

Code: https://github.com/ziadtamer756-arch/Seismic-AI (branch `main`)

Date of this draft: 19 November 2026

This draft uses ONLY numbers that are actually in the repository's `evaluation/` files. Where those files disagree with each other, the draft says so instead of picking the biggest number. See §5.4 "Results reconciliation — read this before submitting".

Abstract

Earthquake risk assessment from tabular seismic features is a classification and regression problem that can be approached with classical machine learning, deep learning, or an ensemble of both. This paper presents Seismic-AI, an open platform that combines (1) a classical machine-learning comparison, (2) three deep-learning architectures — a deep neural network (DNN), TabNet, and a Wide & Deep network — combined by soft-voting ensemble, (3) SHAP explainability, and (4) a Streamlit research dashboard. The work is evaluated on a prepared earthquake dataset of 20,082

records (`risk_training_data.csv`) with geographic, temporal, depth and historical-seismicity features.

On the held-out test set the deep-learning models reach 90.0–90.4% accuracy, and the soft-voting ensemble reaches 92.33% accuracy (weighted F1 91.58%). A Random Forest baseline reaches 96.94% on the same task in a separate classical comparison, which shows that the deep-learning ensemble does not outperform the strongest classical baseline on this tabular dataset — a result reported honestly here rather than hidden. SeisBench pretrained pickers (PhaseNet, EQTransformer, GPD) on STEAD are used as a baseline for detection and P/S phase picking, a different task from risk classification, so the two are compared qualitatively, not by a single number.

Ablation and leakage checks show that historical-magnitude features do not help: removing them improves accuracy, which points to target-related leakage in the historical features and limits how far the headline figures should be trusted. The platform, data and evaluation files are published openly for reproduction.

Keywords: earthquake risk prediction; ensemble learning; TabNet; SeisBench; STEAD; explainable AI.

1. Introduction

Earthquakes cannot be predicted deterministically, but the risk associated with a seismic event — and its magnitude — can be estimated from its geographic, temporal and historical context. Machine learning has become a standard tool for this estimation, and deep learning has become standard for waveform tasks such as phase picking and event detection.

Most published work treats one of these two worlds: either waveform picking with deep networks, or tabular risk classification with classical models. Seismic-AI brings them together in one

reproducible platform: classical models, deep tabular models, an ensemble, explainability, and an interactive dashboard, with every evaluation result stored as a file in `evaluation/`.

The contributions of this work are:

1. A complete, open pipeline from raw earthquake records to risk classification and magnitude regression.
2. A controlled comparison of classical models (Logistic Regression, Random Forest, XGBoost, MLP) against deep tabular models (DNN, TabNet, Wide & Deep) and their soft-voting ensemble.
3. A SeisBench baseline (PhaseNet, EQTransformer, GPD on STEAD) for the waveform tasks, kept separate from the risk-classification task so the comparison is honest.
4. SHAP-based explanation of which features drive predictions.
5. Ablation and leakage checks, including a negative result: historical-magnitude features do not improve the models.

2. Related work

2.1 Deep-learning pickers and SeisBench

Münchmeyer et al. (2022) benchmarked six deep-learning pickers on eight datasets and three tasks — event detection, phase identification and onset picking — and showed that deep pickers outperform traditional automated picking, while no single picker is best on every dataset (Münchmeyer et al. 2022). Woollam et al. (2022) packaged those models and benchmark datasets into SeisBench, a standardised open-source framework for machine-learning in seismology (Woollam et al. 2022). EQTransformer (Mousavi et al. 2020) performs simultaneous earthquake detection and P/S phase picking with an attentive deep network (Mousavi et al. 2020). The Stanford Earthquake Dataset (STEAD) is the large global benchmark used by these models, with about 1.2 million

waveforms (Mousavi et al. 2019).

These works operate on waveforms. The present work operates on tabular features derived from earthquake catalogues. The two tasks are therefore complementary, not directly comparable, and §4 reports them separately.

2.2 Tabular models for seismic risk

Classical models — Random Forest, XGBoost, Logistic Regression — remain strong on tabular data, often beating deep networks, and TabNet was designed specifically to bring attention-based deep learning to tabular data. This is why the platform evaluates both families on the same split and reports both, including the cases where a classical model wins (§4.1 vs §4.2).

3. Data and methods

3.1 Dataset

The evaluated dataset is `risk_training_data.csv`: **20,082 rows**, 13 columns, stored in the repository. This is the verified size.

An earlier description of the project claimed an "80 GB dataset"; no file in the repository supports that figure, and it must not appear in a submission unless its source is cited. The 20,082-row dataset is the one the models were actually trained and evaluated on.

Columns: `depth`, `latitude`, `longitude`, `hour`, `month`, `day`, `day_of_year`, `season`, `depth_category`, `lat_lon_product`, `events_previous`, `avg_previous_magnitude`, and the classification target `risk_level`, with `magnitude` as the regression target.

The repository also contains `earthquakes.db` (2.1 MB, SQLite) used by the live-prediction modules, and a SeisBench/STEAD baseline setup in `baseline/` for the waveform tasks.

3.2 Preprocessing and features

Preprocessing (`preprocess.py`, `training/preprocess.py`) includes

data cleaning, missing-value handling, feature transformation, scaling (StandardScaler), and preparation of classification and regression targets. Feature engineering adds `day_of_year`, `season`, `depth_category`, `lat_lon_product`, `events_previous` and `avg_previous_magnitude` from the raw fields. The ablation study (§4.4) uses 12 features in the full configuration.

***Reproducibility note (must be fixed before submission):** the root `train_models.py` reads `prepared_earthquake_data.csv`, which is not in the repository; `training/train_models.py` reads `risk_training_data.csv`, which is. One canonical training entry point must be chosen and the other removed or fixed. The preprocessing code also contains a hard-coded Windows path (`F:\Seismic-Agent\earthquakes.db`); it must become a relative path or a value from `config/config.yaml`.*

3.3 Models

Classical models: Logistic Regression, Random Forest, XGBoost, Multi-Layer Perceptron (MLPClassifier for risk, MLPRegressor for magnitude). SMOTE oversampling is used in training (`train_models.py`).

Deep-learning models:

- **DNN** (TensorFlow/Keras): Dense(128) – BatchNormalization – Dropout(0.3) – Dense(64) – Dropout(0.25) – Dense(32) – Softmax; early stopping.
- **TabNet** (PyTorch TabNet): n_d = 32, n_a = 32, n_steps = 5, Adam, learning rate 0.02, sparse feature attention, early stopping.
- **Wide & Deep** (TensorFlow/Keras): wide branch plus deep dense layers, BatchNormalization, Dropout, Softmax output.

Ensemble: soft-voting ensemble of the deep models

("Seismic Ensemble AI"). Note a discrepancy in the repository:

`evaluation/final_results.json` lists the components as DNN +

Wide & Deep + TabNet, while `evaluation/ensemble_results.json`

lists Random Forest + DNN + TabNet + Wide & Deep. The submission

must state one component list, verified by re-running the

ensemble script.

Regression: magnitude prediction with MLPRegressor; reported

in §4.3.

3.4 Evaluation

Accuracy, precision, recall and weighted F1 for classification;

mean absolute error (MAE) and root-mean-square error (RMSE) for

magnitude regression. Cross-validation, hyperparameter tuning,

ablation, leakage testing, confidence intervals and SHAP feature

importance are all stored under `evaluation/`. The dashboard

(`dashboard.py`, Streamlit, English/Arabic toggle) visualises

statistics, predictions, model performance and the global

earthquake map.

4. Results

All numbers below come from the canonical re-run of 19 November 2026 (`evaluation/canonical_results.json`): one protocol for every classical result — stratified 80/20 split, seed 42, StandardScaler, SMOTE on train only. Deep-learning and ensemble figures were not re-run (no TensorFlow/PyTorch on the re-run machine) and are reported as stored in the repository's `evaluation/` files; the source file is named so a reviewer can check it.

4.1 Classical machine learning

From `evaluation/canonical_results.json` (canonical re-run, 19 November 2026 — one protocol: stratified 80/20 split, seed 42, StandardScaler, SMOTE on train only):

Table 1. Classical machine-learning results on the held-out test set.

Model	Accuracy	Precision	Recall	F1
Logistic Regression	0.6702	0.8676	0.6702	0.7278
Random Forest	**0.9694**	0.9704	0.9694	0.9697
XGBoost	0.8728	0.9144	0.8728	0.8862

MLP Neural Network	0.8964	0.9263	0.8964	0.9049

Cross-validation (5-fold, Random Forest, SMOTE inside the pipeline): mean accuracy 0.9689, standard deviation 0.0029 (`evaluation/canonical_results.json`). The old cross-validation figure of 98.15% reported in the repository was not reproduced by the canonical re-run and is superseded; it must not be cited. Cross-validation is reported here as a cross-validation figure, not as test accuracy.

4.2 Deep learning and ensemble

From `evaluation/deep_learning_comparison.json`,

`evaluation/tabnet_results.json` and

`evaluation/wide_deep_results.json`:

Table 2. Deep-learning models and the soft-voting ensemble.

Model	Framework	Accuracy	Weighted F1
DNN	TensorFlow/Keras	0.9004	0.8916
TabNet	PyTorch TabNet	0.9044	0.8936
Wide & Deep	TensorFlow/Keras	0.9042	0.8980
Ensemble (soft voting)	TensorFlow + PyTorch	**0.9233**	**0.9158**

Ensemble figure from `evaluation/final_results.json` and

`evaluation/ensemble_results.json` (identical to 4 decimal places:

accuracy 0.9233258650734378, weighted F1 0.9157817367513021).

The ensemble beats each individual deep model by about 2

percentage points, but it does **not** beat the Random Forest

baseline (96.94%) on this tabular dataset. That is the honest

headline: deep learning adds value over the other deep models,

not over the best classical model.

4.3 Magnitude regression

The magnitude results are not reproducible from the repository as shipped: the file the regression was trained on, `magnitude_training_data.csv`, is not in the repository, and the regression cannot be re-run without it. The two stored magnitude runs in `evaluation/` disagree with each other, and neither can be verified against the data. No magnitude MAE or RMSE figure is cited in this paper. `magnitude_training_data.csv` (or the script that built it) must be restored and the regression re-run under the canonical protocol before any magnitude figure is cited in a submission.

4.4 Ablation study

From `evaluation/canonical_results.json` (Random Forest, canonical protocol; 12 features in the full configuration):

Table 3. Ablation study.

Configuration	Features	Accuracy	F1
Full features	12	0.9694	0.9697

Without historical features	10	**0.9736**	0.9739
Without time features	7	0.9532	0.9540
Geographic features only	3	0.9470	0.9493

Removing historical features *improves* accuracy. Together with §4.5, this is evidence that the historical features carry target-related information rather than generalisable signal. The old ablation file's lower numbers (full features 0.9517) came from a different protocol (class_weight="balanced", no SMOTE), not from the data.

4.5 Leakage test

From `evaluation/canonical_results.json` (Random Forest, canonical protocol; the leakage figures exactly reproduce the old `evaluation/leakage_results.json`):

Table 4. Leakage test.

Configuration	Accuracy	F1
Full features	0.9694	0.9697
Without historical magnitude	**0.9744**	0.9746
Only geographic + time	0.9721	0.9723

Again, removing the historical-magnitude feature improves the

result. A model that gets better when a feature is removed is a

model whose headline figure is inflated by that feature. The

paper therefore recommends the "without historical magnitude" configuration as the conservative reporting baseline. The temporal split in §4.8 now provides the forward-time check this section called for, and it confirms the concern (§5.2).

4.6 Explainability (SHAP)

From `evaluation/shap_importance_report.json`, top features by

SHAP importance:

Table 5. SHAP feature importance (top 7 of 12 features).

Rank	Feature	Importance
1	events_previous	0.2038
2	longitude	0.1575
3	lat_lon_product	0.1252
4	latitude	0.1183
5	depth	0.1169
6	avg_previous_magnitude	0.0751
7	depth_category	0.0627

The model leans most on recent seismic history

(`events_previous`) and on location, which is consistent with

the leakage finding in §4.5: the most important feature is also

the most suspect one.

4.7 SeisBench baseline (waveform task)

From `evaluation/seisbench_baseline_real_results.json` and `evaluation/final_experiment_summary.md`: pretrained PhaseNet, EQTransformer and GPD from SeisBench on STEAD, for earthquake detection and P/S phase picking. The stored file reports Matthews correlation coefficient (MCC) of 1.00 for PhaseNet, 1.00 for EQTransformer and 0.99 for GPD.

Do not present these as measured results of this work without verification. MCC values of exactly 1.00 are implausible as a measured experiment and match published benchmark values; the file itself cites Münchmeyer et al. (2022) as its source. In a submission, either reproduce the baseline run and report the measured residuals (P-arrival and S-arrival errors), or cite the published benchmark as a literature baseline and label it as such.

The baseline and the proposed model solve different tasks (detection/picking vs risk classification), so no single accuracy comparison between them is valid; Table 1 in the repository's `evaluation/results_table.md` already makes this distinction, and this draft keeps it.

4.8 Temporal split

A temporal split tests forward-time performance: train on the earlier 80% of `day_of_year`, test on the later 20% (`evaluation/canonical_results.json`). Random Forest reaches accuracy 0.8564 and weighted F1 0.8577 (precision 0.8650) — about 11 percentage points below the random-split headline of 96.94%. The headline figures therefore overstate how the model would perform on genuinely future earthquakes. The no-historical-magnitude configuration remains the conservative primary result (§4.5, §5.2).

5. Discussion

5.1 What the results actually show

The ensemble works as designed: soft voting over three different deep architectures lifts accuracy from ~90.4% (best single deep model, TabNet) to 92.33%. The gain is real but modest, and it

does not close the gap to Random Forest (96.94%) on this dataset.

That is a normal outcome for tabular data, where tree ensembles

often beat deep networks, and reporting it is a strength of the

paper, not a weakness: it tells a reviewer the authors compared

against a strong baseline instead of a weak one.

5.2 The historical-features problem

The two most important findings are negative ones. First, the

model's most important feature (`events_previous`, SHAP 0.2038)

is a historical feature. Second, removing historical features

improves accuracy in both the ablation study (96.94% → 97.36%)

and the leakage test (96.94% → 97.44%). If the historical

features contained only generalisable signal, removing them

would hurt. That they help when removed means the headline

figures are partly driven by information that would not be

available — or not reliable — for a genuinely future earthquake.

The paper's recommendation is therefore to report the no-historical-magnitude configuration as the primary result. The temporal split in §4.8 confirms the concern: on later events, Random Forest drops to 85.64% accuracy, about 11 points below the random-split headline.

5.3 The dashboard as a research tool

The Streamlit dashboard is not only a demo: it is the

reproducibility surface of the project (live prediction,

statistics, model performance, global map, English/Arabic).

A reviewer can run it against the published code and data and

check the headline numbers. The repository currently has three

dashboard variants (`dashboard.py`, `app/dashboard.py`,

`dashboard_before_deep_learning.py`); one canonical dashboard

must be declared before submission.

5.4 Results reconciliation — read this before submitting

A clean canonical re-run was done on 19 November 2026 (`training/canonical_rerun.py`, `evaluation/canonical_results.json` and `evaluation/canonical_results.md`): one protocol for every classical

result — stratified 80/20 split, seed 42, StandardScaler, SMOTE on train only. It resolves most of the conflicts the repository's old evaluation files contained.

Resolved

1. Classification accuracy: canonical Random Forest 0.9694 / F1 0.9697; XGBoost 0.8728 / 0.8862; MLP 0.8964 / 0.9049; Logistic Regression 0.6702 / 0.7278 (exact match with the old file). The classification figures in `model_metrics.json` (accuracy 95.24%) are superseded by the canonical re-run and must not be cited.

2. Ablation: canonical full features 0.9694 / 0.9697; without historical features 0.9736 / 0.9739; without time features 0.9532 / 0.9540; geographic only 0.9470 / 0.9493. The old ablation file's lower numbers came from a different protocol (class_weight="balanced", no SMOTE), not from the data.

3. Leakage: the old leakage figures are exactly reproduced — full features 0.9694 / 0.9697; without `avg_previous_magnitude` 0.9744 / 0.9746; geographic + time 0.9721 / 0.9723.

4. Cross-validation: 5-fold Random Forest, SMOTE inside the pipeline: 0.9689 ± 0.0029. The old cross-validation figure of 98.15% reported in the repository was not reproduced by the canonical re-run and is superseded.

5. Temporal split (new, §4.8): train on the earlier 80% of `day_of_year`, test on the later 20% — Random Forest accuracy 0.8564, F1 0.8577, about 11 points below the random-split headline. The headline figures overstate forward-time performance.

6. Canonical confusion matrix: `evaluation/confusion_matrix_canonical.png` — Random Forest [[604, 29, 0], [72, 3231, 11], [0, 11, 59]].

Still open

7. Magnitude regression (§4.3): not reproducible — `magnitude_training_data.csv` is missing from the repository. The stored MAE/RMSE values disagree with each other and neither can be verified; no magnitude MAE or RMSE figure may be cited until the file is restored.

8. Confidence interval file (`evaluation/confidence_interval.json`) claims accuracy 0.9960 with a 95% CI of [0.9940, 0.9978]. No test result in the repository reaches 99.6%. Do not cite this number until its computation is reproduced and explained.

9. Deep learning and ensemble: not re-run (no TensorFlow/PyTorch on the re-run machine). DNN 0.9004, TabNet 0.9044, Wide & Deep 0.9042 and ensemble 0.9233 are reported as stored in the repository's evaluation files.

10. Ensemble components differ between `final_results.json` and `ensemble_results.json` (§3.3).

11. SeisBench MCC of 1.00 (§4.7): verify or re-label as a literature baseline.

12. "80 GB dataset" claim: unverified; the evaluated dataset is 20,082 rows, 1.5 MB. Remove the claim or cite its source.

6. Limitations

- The dataset is tabular catalogue features, not waveforms; the

work does not pick phases or detect events itself — that is

the SeisBench baseline's job.

- The historical features show signs of target-related leakage (§5.2); a temporal split has now been run (§4.8) and confirms the concern: Random Forest drops to 85.64% accuracy on later events.
- Model weights (`models/*.pkl`, Git LFS, 65–230 MB each) are

large; a submission must decide whether to host them via LFS

or as a release asset, and a reviewer must be able to run the

code without them (e.g. by retraining from the published data).

- The dashboard is triplicated; the training entry point is

duplicated; a hard-coded Windows path exists (§3.2).

- The clean re-run of 19 November 2026 resolved the classical-model conflicts (§5.4); the numbers in §4 are now from one canonical protocol, except the deep-learning and ensemble figures (not re-run) and the magnitude regression (not reproducible — §4.3).

7. Conclusion

Seismic-AI is a complete, open platform for earthquake risk

classification and magnitude prediction: classical and deep

tabular models, a soft-voting ensemble at 92.33% accuracy

(weighted F1 91.58%) on 20,082 records, SHAP explainability, a

SeisBench waveform baseline, and an interactive dashboard.

Its most useful result is not the highest number but the honest comparison: the ensemble improves on every deep model and still loses to Random Forest, and the historical features that drive the headline figures do not survive a leakage check. A temporal split (§4.8) drops Random Forest to 85.64% accuracy on later events, so the headline figures overstate forward-time performance. With the clean re-run and the temporal split done (19 November 2026) and a declared canonical pipeline, the work is submittable as a reproducible applied-ML paper — once the magnitude data are restored, the deep models re-run, and the ensemble components resolved (§5.4, Appendix B).

Data and code availability

- Code: https://github.com/ziadtamer756-arch/Seismic-AI

(branch `main`, MIT licence).

- Data: `risk_training_data.csv` (20,082 rows) in the repository;

`earthquakes.db` (SQLite, 2.1 MB) in the repository.

- Evaluation files: `evaluation/` in the repository

(results tables, model comparisons, ablation, leakage,

SHAP, confidence intervals, figures).

- Model weights: `models/*.pkl` via Git LFS (98 MB, 230 MB, 65 MB); retraining from the published data is the recommended reproduction route.
- No new data were collected for this paper; the STEAD baseline uses the published STEAD benchmark through SeisBench.

List of abbreviations

AI artificial intelligence; DNN deep neural network; F1 F1-score; LFS Large File Storage; MAE mean absolute error; MCC Matthews correlation coefficient; MLP multi-layer perceptron; RMSE root-mean-square error; SHAP SHapley Additive exPlanations; STEAD Stanford Earthquake Dataset.

Statements and Declarations

Funding: The authors declare that no funds, grants, or other support were received during the preparation of this manuscript. (Update if funding applies.)

Competing Interests: The authors have no relevant financial or non-financial interests to disclose.

Author Contributions: Ziad Tamer Fathi Soltan: conceptualization, software, data curation, formal analysis, writing — original draft. (Co-author contributions to be added with Mahmoud Youssef and Magy El-Benhawy, once confirmed.)

Data Availability: The datasets and evaluation files analysed during the current study are available in the Seismic-AI repository, https://github.com/ziadtamer756-arch/Seismic-AI (`risk_training_data.csv`, `earthquakes.db`, `evaluation/`). The STEAD benchmark is available through SeisBench.

Ethics approval: Not applicable. This study uses published

seismic catalogue data and involves no human participants, animals, or personal data.

Use of AI tools: AI-assisted tools were used for drafting and copy-editing support; the author takes full responsibility for the content and verified all reported numbers against the repository's evaluation files.

Acknowledgements

To be completed: supervisor and any funding acknowledgement.

Do not submit with this section empty if funding applies.

References

Mousavi SM, Ellsworth WL, Zhu W, Chuang LY, Beroza GC (2020) Earthquake transformer — an attentive deep-learning model for simultaneous earthquake detection and phase picking. Nat Commun 11:3952. https://doi.org/10.1038/s41467-020-17591-w

Mousavi SM, Sheng Y, Zhu W, Beroza GC (2019) STanford EArthquake Dataset (STEAD): a global data set of seismic signals for AI. IEEE Access 7:179464–179476. https://doi.org/10.1109/ACCESS.2019.2947848

Münchmeyer J, Woollam J, Rietbrock A, Tilmann F, Lange D, Bornstein T, Diehl T, Giunchi C, Haslinger F, Jozinović D, Michelini A, Saul J, Soto H (2022) Which picker fits my data? A quantitative evaluation of deep learning based seismic pickers. J Geophys Res Solid Earth 127(1):e2021JB023499. https://doi.org/10.1029/2021JB023499

Woollam J, Münchmeyer J, Tilmann F, Rietbrock A, Lange D, Bornstein T, Diehl T, Giunchi C, Haslinger F, Jozinović D, Michelini A, Saul J, Soto H (2022) SeisBench — a toolbox for machine learning in seismology. Seismol Res Lett 93(3):1695–1709. https://doi.org/10.1785/0220210324

Zhu W, Beroza GC (2019) PhaseNet: a deep-neural-network-based seismic arrival-time picking method. Geophys J Int 216(1):261–273. https://doi.org/10.1093/gji/ggy423

References are in Springer (author–year) style, alphabetised

by first-author surname, as required by Earth Science

Informatics. All five entries are verified against the publisher records (19 Nov 2026). No other references are cited in this draft — do not add a reference you have not read.

Appendix A — Earth Science Informatics compliance checklist

- [x] Journal: Earth Science Informatics (Springer Nature);

  article type: Research Article.

- [x] Abstract 150–250 words (this draft: ~240 words, no

  undefined abbreviations, no references).

- [x] Keywords: 6 (journal requires 4–6).
- [x] Title page: author name, affiliation, corresponding

  author and email. ORCID: to be added if available.

- [x] Structure: Title/Abstract, Introduction, Methods,

  Results, Discussion, Conclusions, List of abbreviations,

  Acknowledgements, References, Tables (the journal allows

  alternative structure; "Related work" is kept as §2 and

  "Limitations" as §6).

- [x] Headings: no more than three levels.
- [x] References: author–year in text, list alphabetised by

  first-author surname, DOIs as full links (Springer style).

- [x] Statements and Declarations: Funding, Competing

  Interests, Author Contributions, Data Availability,

  Ethics, AI-tools use — all drafted (submissions without

  them are returned as incomplete).

- [x] Tables numbered with captions (Tables 1–5).
- [ ] Figures: confusion matrix (`evaluation/confusion_matrix_canonical.png`, Random Forest, canonical re-run) and SHAP plot from `evaluation/` — prepare figure files per Springer's artwork guidelines.
- [ ] Manuscript file: convert to .docx, 10-point Times

  Roman, automatic page numbering (journal requires Word).

- [ ] Supplementary material: evaluation JSON files and the

dashboard link.

Appendix B — Open items before submission (in order)

1. **Journal chosen: Earth Science Informatics.** Formatting

adapted to its author guide (this draft). Remaining:

convert to .docx and prepare figure files.

2. **Clean re-run** (§5.4): DONE 19 November 2026 — one split (stratified 80/20, seed 42), one canonical script (`training/canonical_rerun.py`), one results file (`evaluation/canonical_results.json`); classical conflicts resolved (§5.4). Canonical confusion matrix: `evaluation/confusion_matrix_canonical.png` (Random Forest [[604, 29, 0], [72, 3231, 11], [0, 11, 59]]).

training script, one results file; resolve the seven conflicts.

3. **Temporal split** (§5.2, §4.8): DONE 19 November 2026 — train on the earlier 80% of `day_of_year`, test on the later 20%: Random Forest accuracy 0.8564, F1 0.8577. Report the no-historical-magnitude configuration as primary.

later events; report the no-historical-magnitude

configuration as primary.

4. **Verify or re-label the SeisBench MCC values** (§4.7).

5. **Verify the Mousavi et al. (2019) and Zhu and Beroza

(2019) references** and add any further references

actually read.

6. **Add co-authors** (Mahmoud Youssef, Magy El-Benhawy) only

with their emails, affiliations, roles and consent.

7. **Declare the canonical dashboard** and fix the Windows

hard-coded path and the missing

`prepared_earthquake_data.csv` entry point.

8. **Decide LFS vs release asset** for the model weights.