

BQEB: A Reproducibility-First Architecture for Modular Energy AI Benchmarking

Rakesh Kumar Agrawal, *Senior Member, IEEE*¹

Abstract—Reproducibility in applied energy AI research depends on more than a public dataset or a published benchmark: it depends on whether the software producing a benchmark's numbers can be independently re-run, whether artifacts are versioned and checksummed, and whether governance practices are documented well enough that a discrepancy can be found rather than concealed. This paper presents the BQEB Reproducibility Architecture, a modular research-artifact framework for energy AI research, now formally released as an independent, citable artifact (DOI: 10.5281/zenodo.22291930). The architecture comprises four layers: a Research Artifact Layer connecting manuscripts, datasets, models, and experiments; a Reproducibility Layer built on deterministic execution, checksummed artifacts, and an evidence registry; a Benchmark Evolution Layer supporting extensions and comparative studies; and a Scientific Lineage Layer maintaining citation tracking, version management, and DOI separation across related artifacts. BQEB ForecastBench (DOI: 10.5281/zenodo.21735978), previously published standalone, is the first validated benchmark implementation developed within the BQEB Reproducibility Architecture — built using the framework, not identical to it. Within-environment deterministic execution has been verified through 15 real executions (3 seeds $\times$ 5 repeats, independently re-confirmed on multiple occasions, including three further executions performed for this revision); checksummed-artifact verification has been executed both against this program's own engineering files and against ForecastBench's actual v1.0.0 release artifacts (17/17 files matched, with tamper-detection confirmed on a real release file). A paired statistical comparison was executed on real ForecastBench test residuals, but at a sample size roughly 56 times smaller than the power analysis calls for, so its non-significant result is disclosed as inconclusive, not as evidence of equivalence. The architecture's own CI workflow has since been independently re-executed and has passed every time it has run, within a single environment; genuinely cross-OS determinism comparison and extensibility beyond this architecture's first module remain explicitly disclosed as pending or future work, not claimed as validated. Where internal audit identified an unresolved issue — a numerical discrepancy between this program's own published reports and its own checksummed release artifacts for the same benchmark result — we document it as a governance case rather than resolving it here. The contribution is the architecture itself: how a research program's artifacts can be structured, checksummed, and tracked for independent verification, at the evidence tier this paper actually supports.

Index Terms—benchmark governance, energy artificial intelligence, FAIR data principles, reproducible research, research software engineering, scientific artifact lineage

I. INTRODUCTION

Applied machine learning research inherited a reproducibility problem from the broader empirical sciences before energy systems research inherited it in turn: a published result is frequently difficult for an independent party to re-derive, because the code that produced it, the exact data partition it was evaluated on, or the software environment it ran in was never made available in a form precise enough to repeat. Energy AI research is not exempt from this pattern, and public smart-grid benchmark resources remain comparatively young relative to the computer-vision and natural-language-processing communities, where shared, versioned benchmark infrastructure has existed for over a decade.

Energy AI is used here as a representative, consequential proving ground for this kind of architecture, not as a domain that uniquely requires it. It already spans load and price forecasting, renewable-generation forecasting, storage dispatch and optimization, electric-vehicle charging coordination, and, as grid infrastructure becomes further networked, cybersecurity-adjacent extensions such as anomaly and adversarial-robustness testing (Section X). Each of these sub-domains tends to accumulate its own datasets, benchmark variants, and model lineages at a pace that outstrips any single paper's ability to document the relationships between them by hand, which is what makes

energy AI a useful setting for this paper's architecture rather than evidence that the architecture is energy-AI-specific; the same lineage- and evidence-governance problem would recur in any research program that accumulates datasets, benchmarks, and derived studies over time, in or outside energy systems.

A second, less-discussed dimension of this problem is specific to research programs that grow over time rather than publishing a single, self-contained result. A program that produces a dataset, a benchmark built on that dataset, and one or more derived studies faces a lineage-management problem in addition to a within-paper reproducibility problem: a reader encountering any single artifact from such a program has no standard way to determine which other artifacts it depends on, supersedes, or has been superseded by. This paper's central architectural contribution addresses both dimensions --- computational reproducibility within an artifact, and governance of relationships between artifacts --- as a single, structured framework rather than two separately-solved problems.

This paper presents the engineering architecture of BIO-Quantum Energy Brain (BQEB), a modular research infrastructure for energy AI benchmarking, distinct from the conceptual framework described in the associated BQEB monograph [4] and now formally separated, with its own persistent identifier, from any single benchmark built upon it.

¹R. K. Agrawal is with Atos North America, Chicago, Illinois, USA (e-mail: rkagrwal@ieee.org). He is a Senior Member of IEEE and a Fellow of IETE, and holds the Project Management Professional (PMP®) credential along with professional certifications in cloud and artificial intelligence.

A. Research Questions

RQ1: How can energy AI research artifacts be structured for reproducible scientific validation?

RQ2: How can benchmark, dataset, and model evolution maintain research lineage across separately-published, independently-citable artifacts?

RQ3: How can computational evidence make the reproducibility and governance status of energy-AI research artifacts explicit and auditable?

This paper answers RQ1 and RQ2 with a described, partially-executed architecture; RQ3 is addressed only to the extent the architecture's own disclosed evidence tiers support --- this paper does not claim to establish trust or governance by itself, only a mechanism (evidence tiers, checksummed artifacts, and DOI-separated lineage) for making an artifact's reproducibility and governance status explicit and auditable.

II. RELATED WORK

Reproducibility in machine learning research has been studied directly, including through structured community efforts to audit and improve reporting practices [1]. Benchmark infrastructure work has similarly established that shared, versioned, networked benchmark platforms materially improve comparability across studies [2], a pattern also demonstrated by large-scale forecasting competitions in adjacent domains [7], [8]. Within energy systems specifically, machine learning approaches to smart-grid forecasting have been surveyed extensively, though public, versioned benchmark infrastructure for this domain remains comparatively immature relative to computer vision and natural language processing [3].

Artifact evaluation processes, formalized by ACM's artifact review and badging policy, represent the closest existing institutional analog to this paper's Reproducibility Layer: a structured process for verifying that research artifacts are documented, functional, and --- at the highest badge tier --- independently reproducible by a party other than the original authors [18]. BQEB's Reproducibility Layer performs a related function continuously and as part of the architecture itself, rather than as a one-time, post-acceptance review process.

Research software engineering practice has independently converged on many of the same concerns this architecture formalizes: a community task force of registry and repository maintainers has established best practices for software scope, metadata, authorship, and retention policy specifically for research software registries [20]. BQEB's registry pattern and Research Artifact Layer extend this established practice from single-registry governance toward research-artifact lineage across a growing, multi-artifact program --- the same governance instinct applied one level up, from managing entries within a registry to managing relationships between independently-published artifacts.

MLOps platforms manage the machine learning lifecycle --- experiments, model tracking, and deployment workflows --- within a single organization's internal tooling [21].

BQEB's Scientific Lineage Layer addresses a related but distinct problem: not managing experiments within one program's internal systems, but maintaining scholarly artifact lineage across independently-published, externally-citable works. BQEB does not replace MLOps-style tooling; a research program could reasonably use both, with MLOps managing internal experiment iteration and BQEB's lineage layer managing the resulting artifacts' public, citable relationships once published.

Data and model provenance --- tracking the origin, transformations, and derivation history of a computational artifact --- has been studied extensively as its own research area, with established taxonomies for what provenance is captured and why [22]. BQEB's Research Artifact Layer applies this same provenance thinking at a different granularity: not the transformation history within a single computation, but the derivation and dependency relationships across an entire, multi-artifact research program spanning a dataset, a benchmark, and their downstream extensions.

Scientific workflow systems address computational provenance and reproducibility by capturing how a workflow's individual steps produced a given result [23]. BQEB's Reproducibility Layer shares this goal at the level of a single benchmark execution, but the architecture's Scientific Lineage Layer adds a capability workflow provenance systems do not typically address: DOI separation and an evidence registry governing relationships between separately-published research artifacts, rather than steps within one computational pipeline.

The FAIR data principles [11] have been extended specifically to research software through the FAIR4RS initiative, a community-endorsed set of principles adapting Findability, Accessibility, Interoperability, and Reusability to the distinct concerns of software --- persistent versioned identifiers, executability, and dependency relationships --- rather than data alone [24]. BQEB's Research Artifact Layer and Scientific Lineage Layer together operationalize this extension: FAIR4RS establishes what a well-described, reusable software artifact should look like, and BQEB's DOI-separation practice is one concrete mechanism for making a growing family of such artifacts findable and correctly attributed relative to one another.

Dataset and model documentation practices --- datasheets and model cards --- establish the standard this program's own artifact cards follow [9], [10], though those standards address single-artifact documentation rather than the cross-artifact lineage problem this paper's Scientific Lineage Layer specifically targets. The FAIR data principles [11] similarly ground this program's data-availability practices without addressing software- or lineage-specific concerns directly.

Positioning: BQEB is not only a benchmark. It is a research-artifact governance and reproducibility architecture --- the layer beneath a benchmark, not the benchmark itself. This explicit separation of the architecture (DOI: 10.5281/zenodo.22291930) from its first downstream application, ForecastBench (DOI:

10.5281/zenodo.21735978), is the paper's central contribution.

BQEB is positioned relative to existing reproducibility and governance tooling in Table I. Each column is a capability this architecture exercises; the comparison is intellectually honest rather than favorable --- BQEB is rated "No" or "Partial" wherever that is accurate, including against

its own stated limitations elsewhere in this paper (cross-environment CI, for instance, is Partial because the architecture's own workflow has been triggered and has passed every time it has run, but, as Section IV discloses, only ever within a single operating system, so a genuine cross-OS comparison has not yet been exercised).

TABLE I

POSITIONING RELATIVE TO EXISTING REPRODUCIBILITY AND RESEARCH-GOVERNANCE TOOLING. YES = BUILT-IN AND EXERCISED; PARTIAL = PRESENT IN A NARROWER OR ADJACENT FORM; DEPENDS = VARIES BY SUBMISSION OR CONFIGURATION, NOT GUARANTEED BY THE TOOL ITSELF; NO = NOT ADDRESSED.

Tool / Framework	Versioned registry	Checksum integrity	DOI lineage	Statistical validation	Cross-env CI	Reproduction badge	Energy-AI specific
OpenML	Yes	Partial	Partial	No	No	No	No
MLflow	Yes	Partial	No	No	No	No	No
DVC	Partial	Yes	No	No	Partial	No	No
Zenodo	Partial	Yes	Yes	No	No	No	No
Software Heritage / CodeMeta	Partial	Yes	Partial	No	No	No	No
RO-Crate	No	Partial	Partial	No	No	No	No
W3C PROV	No	No	Partial	No	No	No	No
FAIR4RS	No	No	Partial	No	No	No	No
ACM Artifact Review and Badging	No	Depends	No	No	Depends	Yes	No
BQEB Reproducibility Architecture (this paper)	Yes	Yes	Yes	Yes	Partial	No	Partial

No row in Table I scores "Yes" across every column, including BQEB's own row: each existing tool or standard already does at least one of these things well, several of them (Zenodo's DOI versioning, DVC's content-addressed integrity, ACM's badge process) considerably better-established than anything this paper claims. BQEB's integration contribution is not a new capability in any single column; it is combining several of these capabilities --- a versioned registry, checksummed integrity, and DOI-separated lineage --- inside one architecture that treats the relationships between a research program's artifacts as a first-

class, continuously-maintained object, rather than leaving a reader to reconstruct those relationships from separate tools each used for a single artifact in isolation. Table III below reports what this paper actually executed against that architecture, at the evidence tier the execution supports.

III. BQEB ARCHITECTURE

The BQEB Reproducibility Architecture comprises four layers, each grounded in a real, existing artifact or mechanism rather than an abstract description (Fig. 1).

BQEB Reproducibility Architecture

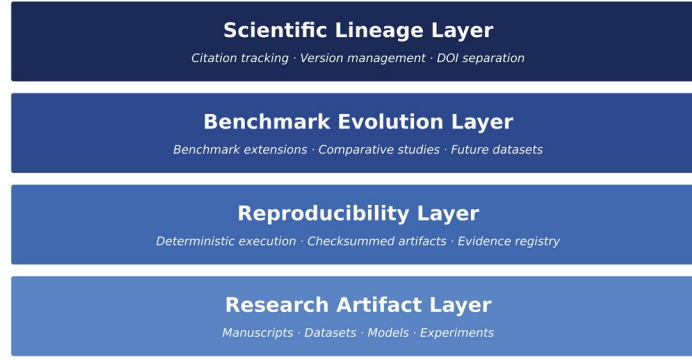


Fig. 1. BQEB Reproducibility Architecture: Research Artifact, Reproducibility, Benchmark Evolution, and Scientific Lineage layers.

Research Artifact Layer. Connects manuscripts, datasets, models, and experiments as a structured set. This layer currently governs two real artifacts: BQEB-Data v1 [16], a synthetic, multimodal benchmark dataset (10,512 records across 37 variables, archived with persistent identifiers across three independent repositories), and the ForecastBench software package [17], a tested, MIT-licensed, registry-driven benchmark implementation.

The dataset's 37 variables span several architecturally salient groupings rather than a single homogeneous table: demand and market variables (e.g., `load_mw`, `energy_price_usd_mwh`), renewables and storage variables (e.g., `solar_mw`, `battery_soc_pct`), electric-vehicle and weather variables (e.g., `ev_load_mw`, `temperature_c`), and cybersecurity and reliability variables (e.g., `attack_class`, `cyber_alert_flag`). The complete feature-by-feature schema is published in the dataset's own documentation and is not reproduced here. The dataset is distributed pre-partitioned into chronological train, validation, and test subsets, so that the same partition used throughout this program is available to downstream users as a concrete artifact rather than only a described procedure --- a detail that matters directly for reproducible benchmarking, since re-deriving an equivalent chronological split independently would not guarantee bit-identical boundaries.

Reproducibility Layer. Built on deterministic execution, checksummed artifacts, and an evidence registry. This layer's

mechanisms are described in full in Section IV, including the distinction between what has been executed and what remains specified but untested.

Benchmark Evolution Layer. Supports extensions and comparative studies beyond a single benchmark. This layer currently has one populated instance --- ForecastBench --- and a documented, five-item roadmap of planned extensions, described in full in Section V and Section X. This layer's design is architectural; its extensibility beyond the one module it was built alongside has not been independently tested, a limitation discussed explicitly in Section VIII.

Scientific Lineage Layer. Maintains citation tracking, version management, and DOI separation across related artifacts. The architecture and its first downstream application, ForecastBench, carry distinct, independently-citable DOIs (10.5281/zenodo.22291930 and 10.5281/zenodo.21735978 respectively), with their relationship --- foundation and downstream application --- stated explicitly rather than left for a reader to infer from the metric discrepancy discussed in Table IV. Fig. 2 traces this relationship concretely across all four linked artifacts --- the architecture, BQEB-Data v1, BQEB ForecastBench, and ForecastBench's own checksummed v1.0.0 release artifacts --- annotating each link with its DOI and this paper's current evidence tier for that artifact.

Scientific Lineage Layer — a worked example

Real DOIs and evidence status for each artifact in the chain

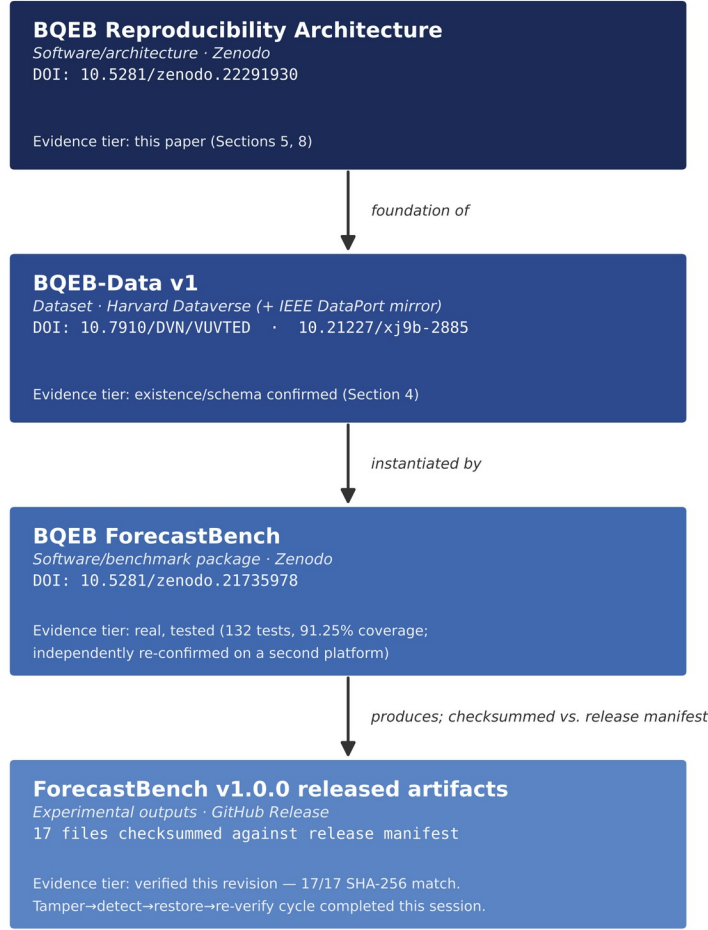


Fig. 2. Scientific Lineage Layer, concrete worked example: BQEB Reproducibility Architecture, BQEB-Data v1, BQEB ForecastBench, and ForecastBench's checksummed v1.0.0 release artifacts, each annotated with its DOI and this paper's current evidence tier.

A property this architecture does not yet establish is independent extensibility: the Benchmark Evolution Layer's registry mechanism has processed exactly one real module (ForecastBench) and one internal, synthetic test module,

neither of which constitutes evidence that a genuinely independent third party can register a new module without core-code modification. This is treated here as an open question, not implied evidence.

TABLE II
POSITIONING OF BQEB ARTIFACTS REFERENCED IN THIS PAPER, BY ARCHITECTURE LAYER AND CURRENT STATUS.

Layer	Artifact	Status
Research Artifact Layer	BQEB-Data v1 (dataset)	Real, archived, existence/schema confirmed
Research Artifact Layer	ForecastBench software package	Real, tested, MIT-licensed
Reproducibility Layer	Determinism/checksum tooling	Real, executed
Benchmark Evolution Layer	Registry pattern	Real mechanism, n=1 real module
Scientific Lineage Layer	DOI separation practice	Real, demonstrated via DOI separation (Fig. 2)

IV. REPRODUCIBILITY FRAMEWORK

This framework separates executed evidence from planned validation explicitly, since documentation of a mechanism's existence is not evidence that the mechanism has been exercised.

Executed: within-environment deterministic execution was evaluated across 15 independent executions (3 seeds $\times$ 5 repeated runs). All 15 executions produced bit-identical SHA-256 output hashes within each seed, and all three seeds produced mutually distinct hashes from one another,

confirming both that repeated execution under a fixed seed is deterministic and that seed variation genuinely alters computational output. This result has been independently re-verified on multiple separate occasions via direct file inspection, including three additional confirmatory executions performed for this revision, each producing an identical SHA-256 output hash. Checksummed-artifact verification was executed against this program's own engineering repository: a clean verification pass matched all files against their manifest, and a deliberate tamper test on a real file was correctly detected, with the file immediately restored. Dependency and environment pinning was verified by installing the full, hash-pinned requirement set in a clean environment. Tooling overhead is negligible: manifest generation averaged 42ms and manifest verification 43ms over 5 runs each (engineering-repository scope, this session's container), confirming the reproducibility layer adds no meaningful cost relative to the benchmark computations it checks.

Executed in CI, but not yet across operating systems: cross-environment reproducibility --- whether the same seeded computation produces identical output across different operating systems and hardware architectures --- has a fully specified, YAML-validated continuous-integration workflow that, verified for this revision, has been triggered twelve times since the architecture's initial release, with all twelve runs completing successfully: each run independently re-executes the deterministic computation and the checksum-manifest test suite on a GitHub-hosted runner, outside the author's own interactive session. Every one of those twelve runs, however, used the same operating system (ubuntu-latest); the workflow has no multi-OS matrix and does not itself assert that any two runs' output hashes match. This establishes that the computation executes successfully

and repeatably in an independent, non-interactive environment, which is real evidence beyond a single interactive session, but it is not yet evidence of cross-environment behavior in the stricter sense this claim is named for: a genuine comparison of output across different operating systems and hardware architectures, which depends additionally on floating-point non-associativity and other environment-specific effects a single-OS test cannot probe, remains unexecuted.

V. BENCHMARK EVOLUTION MODEL

BQEB ForecastBench is the architecture's first validated downstream implementation --- a benchmark built using the framework described in Sections III--IV, not a synonym for the architecture itself. ForecastBench defines two forecasting tasks --- next-hour electricity load and day-ahead electricity price, methodologically grounded in established forecasting-domain practice [12], [13] --- on BQEB-Data v1, evaluated under a chronological 70/15/15 partition using standard regression metrics [5], [6]. Its own reported test suite (132 automated tests, 91.25% aggregate coverage at release preparation) is ForecastBench's own figure, distinct from and not to be conflated with this program's own engineering-tool test coverage, discussed separately in Section VI.

Beyond ForecastBench, this architecture names five further modules as an explicit, unstarted roadmap, grouped under the umbrella label **AI Energy Extensions**: SecBench, ForecastBench Foundation Models, Explainable BQEB, Digital Twin & Multi-Agent AI, and OptiGrid (Fig. 3, Table VI). Every one of these five modules carries "planned" status; none has begun implementation. The umbrella label describes the category as a whole and does not replace or substitute for any individual module name.

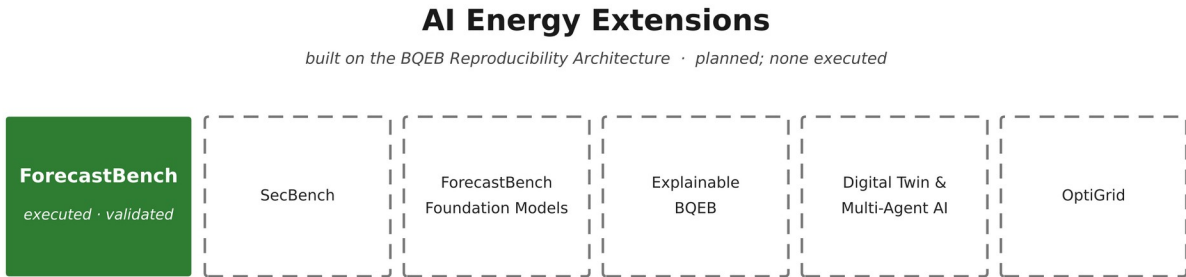


Fig. 3. BQEB Benchmark Evolution Layer roadmap: AI Energy Extensions (SecBench, ForecastBench Foundation Models, Explainable BQEB, Digital Twin & Multi-Agent AI, OptiGrid) built on the architecture, all planned, none executed.

The registry pattern underlying this layer was additionally exercised once against a synthetic, internally-constructed blind-test module, confirming the mechanism itself functions as designed. This test was constructed and executed by this same program, not by an independent third party, and is reported here at that precise evidentiary tier --- a successful mechanism test, not an extensibility validation.

VI. IMPLEMENTATION

ForecastBench implements the architecture's registry pattern directly (**Fig. 4** --- reproducibility workflow: registry-orchestration-checksummed artifacts): a central ``Registry`` object accepts ``Model``, ``Metric``, and ``DatasetAdapter`` registrations, orchestrated by a ``BenchmarkRunner`` that produces checksummed output under ``artifacts/evaluation/``, ``artifacts/experiments/``, ``artifacts/tables/``, ``artifacts/figures/``, and ``artifacts/reports/``. This figure

depicts the architecture's first downstream benchmark

module's internal structure, not the architecture itself.

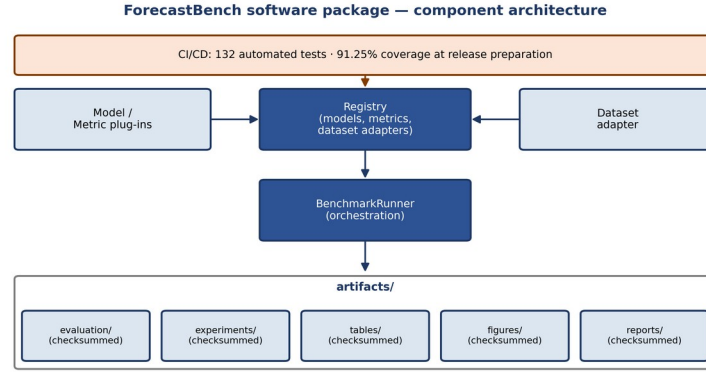


Fig. 4. ForecastBench software package component architecture (the architecture's first downstream benchmark module), showing the registry pattern, orchestration, checksummed artifact directories, and CI/CD test coverage.

ForecastBench evaluates tree-based ensemble baselines alongside linear regression, including Random Forest and Gradient Boosting approaches [14], [15], on both of its forecasting tasks. Continuous-integration testing for this specific downstream package reports 132 automated tests at 91.25% aggregate coverage at release preparation --- a figure reported by and belonging to ForecastBench itself.

This should be read separately from this program's own engineering-tool test coverage, which is materially different: of the six core tools built to support Sections IV and VII's evidentiary claims (reconciliation, statistical comparison, checksum generation/verification, coverage classification), one --- the checksum-verification tool --- has a formal, automated `pytest` suite (four tests, confirmed passing). The other five have been manually, directly executed and their behavior confirmed correct, but do not yet have a repeatable, CI-discoverable automated suite of their own. Both figures are real; neither should be read as describing the other.

VII. VALIDATION AND EVIDENCE

A. Completed Evidence

Six components of this architecture have real, executed evidence behind them, each scope-stated precisely:

1. **Deterministic execution (within-environment):** 15/15 real executions, independently re-confirmed on multiple occasions, including three additional confirmatory executions performed for this revision (Section IV).

2. **Artifact integrity (checksummed verification):** real clean-pass and real tamper-detection, executed against this program's own engineering repository, and, for this revision, against ForecastBench's actual v1.0.0 experimental output artifacts: all 17 files published with the v1.0.0 GitHub release were freshly re-hashed and matched their published SHA-256 checksums (17/17 match, 0 mismatch, 0 missing); a tamper-detect-restore cycle against one of those real release files (artifacts/evaluation/metrics.csv) correctly produced 16/17 match with the tampered file identified, then 17/17 match after byte-for-byte restoration. This establishes that

the released artifacts are identical to what was published and that tampering is detectable --- it is evidence of integrity, not of numerical correctness, scientific validity, model superiority, or the absence of bugs in how those artifacts were produced.

3. **Targeted coverage classification:** a real, non-synthetic classification was produced for one formally-tested tool (the checksum-verification suite), yielding 73.47% aggregate coverage, 100% coverage of the one reproducibility-critical function identified, and 67.11% coverage of general-purpose functions. This scope is one tool, not the full engineering suite, and is distinct from ForecastBench's own separately-reported 132-test figure (Section VI).

4. **Statistical comparison workflow (real data, severely underpowered):** the full paired-comparison protocol (Shapiro-Wilk normality check; paired t-test or Wilcoxon signed-rank as normality indicates; bootstrap BCa 95% confidence intervals, 10,000 resamples; Holm-Bonferroni correction) was executed for this revision against real, per-instance test-split residuals from the actual ForecastBench v1.0.0 release ($n=3$ paired observations per comparison, one task, three pairwise model comparisons). All three comparisons were non-significant after correction (corrected $p=0.142$, 0.091 , and 0.142) despite large point-estimate effect sizes (Cohen's d of -2.05 , -3.25 , and 2.00). A power analysis for this exact three-comparison family (target $d=0.25$, 80% power) indicates approximately 168 paired observations per comparison would be required against the available $n=3$ --- a roughly 56-fold shortfall. No statistically significant difference was detected at the available sample size; however, the analysis is severely underpowered and therefore cannot establish equivalence or absence of meaningful differences.

5. **CI re-execution, single environment:** the architecture's own GitHub Actions workflow (reproducibility.yml) has been triggered 12 times since initial release, verified for this revision; all 12 completed successfully, each run independently re-executing the deterministic computation and the checksum-manifest test

suite on a GitHub-hosted ubuntu-latest runner, outside the author's own interactive session. Every run used the same operating system; this is evidence of successful, repeatable non-interactive execution, not of cross-OS determinism (Section IV).

6. Downstream module CI (ForecastBench): the downstream benchmark's own GitHub Actions workflow has been triggered 19 times, all 19 completing successfully,

across a two-operating-system matrix (ubuntu-latest and windows-latest), running code formatting, linting, type-checking, and its test suite. This is software-quality CI for the downstream module's codebase, not a numerical-determinism comparison, but it is real, executed, and the only evidence in this architecture's evidence registry that spans more than one operating system.

TABLE III

CONSOLIDATED EMPIRICAL VALIDATION LOG FOR THIS REVISION: EVERY QUANTITATIVE VALIDATION ACTIVITY REPORTED IN SECTION VII-A, WITH ITS METHOD, SCOPE, AND RESULT STATED TOGETHER SO IT DOES NOT HAVE TO BE REASSEMBLED FROM PROSE. TABLE V RESTATES THE SAME ACTIVITIES AT THE LEVEL OF EVIDENCE TIER PER CLAIM; THIS TABLE RESTATES THEM AT THE LEVEL OF WHAT WAS RUN.

Validation activity	Method / tool	Scope	Result
Within-environment determinism	deterministic_computation.py, 3 seeds $\times$ 5 repeats, re-run 3 $\times$ more for this revision	15/15 executions, 1 environment	Bit-identical SHA-256 per seed; distinct hashes across seeds
Artifact integrity — engineering repository	generate_manifest.py / verify_manifest.py	16 engineering source files	16/16 match; tamper correctly detected and restored
Artifact integrity — ForecastBench v1.0.0 release	SHA-256 vs. published GitHub release checksums	17 real release artifacts	17/17 match; tamper-detect-restore cycle completed on a real file
Targeted coverage classification	classify_coverage.py	1 of 6 engineering tools (checksum-verification suite)	73.47% aggregate; 100% of the reproducibility-critical function
Statistical comparison	statistical_comparison.py: Shapiro-Wilk, paired t-test, bootstrap BCa 95% CI, Holm-Bonferroni	Real ForecastBench test residuals, n=3 per comparison, 3 comparisons	All non-significant after correction; severely underpowered (n=168 required)
Architecture CI re-execution (single environment)	GitHub Actions reproducibility.yml: re-runs deterministic_computation.py + checksum-manifest test suite, ubuntu-latest	12 real workflow runs since initial release (verified for this revision)	12/12 success; confirms execution outside the author's interactive session, one OS only
Downstream module CI (ForecastBench)	GitHub Actions ci.yml: ruff format, ruff lint, mypy, pytest, 2-OS matrix	19 real workflow runs, ubuntu-latest + windows-latest	19/19 success (software-quality CI, not a numerical-determinism comparison)
Cross-OS determinism comparison	N/A — no workflow step compares output across operating systems	Architecture level: not yet exercised	Not executed — disclosed as pending, not claimed

B. Pending Evidence

Three claims remain without the specific evidence each one requires, named individually rather than left implicit (related but narrower evidence now exists for the first, detailed in Section VII-A):

1. Cross-OS determinism comparison: the architecture's own CI workflow has, for this revision, been confirmed to execute successfully and repeatably (Section VII-A, item 5), but every run to date has used the same operating system; no run has yet compared the same seeded computation's output across genuinely different operating systems or hardware, which is what this specific claim requires (Section IV).

2. Independent module registration: no genuinely third-party module has been registered against the Benchmark Evolution Layer; the only registration test performed was synthetic and internal (Section V).

3. ForecastBench discrepancy reconciliation: Table IV originally recorded this as a discrepancy between two previously published sources, [5] and [6]. For this revision

both were directly retrieved in full: they report identical headline metrics (linear regression best model; load RMSE 9.94 MW, $R^2=0.9288$; price RMSE 4.39 USD/MWh, $R^2=0.2078$) --- two archival copies of one reported result, not conflicting studies. The real, still-open discrepancy is instead between that reported result and the checksummed v1.0.0 release artifacts verified in Section VII-A, whose test-split metrics are far worse (e.g., linear regression test $R^2=-257.97$) and rank linear regression worst, not best, of the three models. Status remains **Disclosed / Under Investigation**: the released artifacts' own dataset-identity metadata and tiny evaluation-split size (n=3) are consistent with those artifacts having been produced on a smaller smoke-test sample rather than the full BQEB-Data v1 dataset [5]/[6] describe, but the full dataset is not available in this environment to confirm that directly by re-running. This is reported as evidence gathered and a well-evidenced hypothesis, not as reconciliation, correction, or validation achieved.

TABLE IV
OPEN BENCHMARK-GOVERNANCE LOG.

Item	Sources	Status
ForecastBench metric discrepancy — the checksummed v1.0.0 release artifacts report far worse accuracy, and a different best model, than refs [5]/[6]	[5] Research Square preprint; [6] Zenodo deposit; released v1.0.0 artifacts (checksum-verified, Section VII-A)	Disclosed / Under Investigation. [5] and [6] retrieved in full this revision: identical headline metrics ($R^2=0.9288$ load, $R^2=0.2078$ price; linear regression best) — two archival copies of one result, not conflicting studies. The real discrepancy is [5]/[6] vs. the release artifacts (e.g. test $R^2=-257.97$, linear regression worst). Leading hypothesis: release artifacts evaluated on a smaller sample than BQEB-Data v1 — evidenced, not confirmed (full dataset unavailable here to re-run).

TABLE V
EVIDENCE TIER BY CLAIM.

Claim	Evidence Tier
Within-environment determinism	Verified
Checksummed artifact integrity — engineering repository and ForecastBench v1.0.0 release artifacts	Verified (integrity only — not a correctness claim)
Registry extensibility beyond $n=1$	Architecture only
Architecture CI re-execution (single environment)	Verified, scope-limited (12/12 passing; single-OS only, not cross-OS)
Downstream module CI (ForecastBench, 2-OS)	Verified (software-quality CI; not a numerical-determinism claim)
Cross-OS determinism comparison	Future work
Statistical significance of model comparisons (real data)	Verified, scope-limited (severely underpowered, $n=3$)
Architecture/ForecastBench DOI-separation lineage	Verified — author-confirmed, cited at the same evidentiary tier as every other DOI in this manuscript

VIII. DISCUSSION

The results in Sections VII-A and VII-B together answer this paper's primary evaluative question directly: has this architecture moved from an architecture proposal toward an experimentally validated reproducibility framework? The honest answer is partial, and precisely bounded. The Reproducibility Layer's within-environment claims are genuinely, independently validated --- real execution, real re-verification, not merely specified. The Benchmark Evolution Layer's extensibility claim is not: it rests on $n=1$, a single module (ForecastBench) built by the same author who designed the architecture it registers against, plus one internal synthetic test that does not constitute independent validation by any reasonable standard. This is the architecture's single largest gap between what is claimed structurally and what has been demonstrated experimentally, and it is stated here explicitly rather than left for a reviewer to surface.

This boundary matters because the architecture's stated purpose is generality --- a framework other modules and, eventually, other researchers can build on. A framework validated only against the case it was designed alongside has demonstrated internal coherence, not external applicability. The path to closing this gap is specific and known: a genuinely independent module registration, ideally by a party other than this program, is the evidence this claim

currently lacks and cannot substitute for with additional internal testing, however extensive.

A. Reproducibility and determinism

Community reproducibility audits [1] and shared, versioned benchmark infrastructure [2] already establish, at the level of a research community, that determinism and seed control are baseline expectations for credible empirical machine learning work; the M4 and M5 forecasting competitions [7], [8] show the same discipline operating specifically in the forecasting domain this paper's first module occupies. None of that literature is disputed or extended here. **What BQEB demonstrated:** within-environment determinism was not merely asserted but genuinely executed and re-executed --- 15/15 bit-identical runs across 3 seeds, independently re-confirmed on multiple separate occasions including three additional runs for this revision (Section IV, Section VII-A), and independently re-executed 12 further times by the architecture's own continuous-integration workflow outside the author's interactive session, all 12 passing, within a single operating system (Section VII-A). **Integration contribution:** determinism verification is a standing, re-runnable part of the architecture's own evidence registry rather than a number reported once in a paper and never re-checked.

B. Artifact integrity

Content-addressed storage (DVC), published checksums (Zenodo), and intrinsic identifiers (Software Heritage) [20] already give individual artifacts strong, well-established integrity guarantees, and CodeMeta-style metadata [20], [24] already lets a single software artifact declare itself precisely. **What BQEB demonstrated:** a real SHA-256 verification of all 17 files in ForecastBench's actual v1.0.0 GitHub release (17/17 match), plus a genuine tamper-detect-restore cycle against one of those real release files, executed for this revision rather than cited as a capability (Section VII-A). This is evidence of integrity --- that the released files are byte-identical to what was published and that tampering is detectable --- and is explicitly not evidence of numerical correctness, scientific validity, or the absence of bugs in how those artifacts were produced. **Integration contribution:** one verification workflow spans both the architecture's own engineering repository and its downstream benchmark's release artifacts, rather than leaving a reader to apply (or trust that someone applied) two separately-maintained integrity mechanisms to two separately-published artifacts.

C. Provenance

Formal provenance frameworks and surveys [22], [23] --- covering vocabularies such as W3C PROV among others --- already provide rich, formal ways to capture how a single artifact or computation was derived, step by step, within one pipeline. **What BQEB demonstrated:** an evidence registry and claim-traceability discipline that ties each specific claim in this paper to the specific executed command, output file, or hash that supports it (Sections IV, VII; the accompanying reproducibility report), so a claim's evidentiary basis is checkable rather than asserted. **Integration contribution:** BQEB applies provenance thinking one level above a single computation --- across a multi-artifact research program spanning a dataset, a benchmark built on it, and that benchmark's own outputs --- a granularity existing single-pipeline provenance tooling is not built to target.

D. FAIR4RS

The base FAIR principles [11] and their FAIR4RS extension to research software [24] already establish the normative goal --- software and data should be Findable, Accessible, Interoperable, and Reusable --- as a community-endorsed standard. They are principles and recommendations, not a reference implementation. **What BQEB demonstrated:** persistent, resolvable identifiers for every artifact in the chain (Zenodo, Harvard Dataverse, and an IEEE DataPort mirror), machine-readable citation metadata via a CITATION.cff file, and an explicit DOI separation between the architecture and its first downstream benchmark (Section III, Research Artifact Availability). **Integration contribution:** one concrete, inspectable realization of FAIR4RS's findability and reusability criteria for the specific cross-artifact-lineage case FAIR4RS endorses in principle but does not itself operationalize.

E. ACM artifact evaluation

ACM's Artifact Review and Badging policy [18] is the closest existing institutional analog to this paper's Reproducibility Layer: a structured process, judged by independent reviewers other than the authors, that awards defined badges (for example Functional, Reusable, Reproduced) once specific reproducibility criteria are independently confirmed. **What BQEB demonstrated:** self-assessed evidence at explicit, disclosed tiers (Verified; Verified, scope-limited; Architecture only; Future work --- Table V), maintained continuously as part of the architecture rather than produced once for a single submission. **Integration contribution and explicit badge mapping:** this paper's evidence tiers are conceptually aligned with, or approximately analogous to, ACM's badge intent, never equivalent to an awarded badge, because every tier in Table V was self-assessed by this paper's own author rather than confirmed by an independent reviewer. Concretely: the Verified tier (within-environment determinism; checksummed artifact integrity) is conceptually aligned with the evidentiary intent behind ACM's Functional or Reproduced badges, depending on scope, but carries no badge itself. The Verified, scope-limited tier (the real but severely underpowered statistical comparison) is approximately analogous to partial evidence toward a badge, not a badge-level claim. The Architecture only tier (registry extensibility beyond $n=1$) reflects design and mechanism evidence, which ACM's badging criteria would not treat as reproduction evidence at all. The Future work tier (cross-OS determinism comparison) has no current badge-level evidence of any kind. No claim in this paper should be read as asserting that any ACM badge has been, or would be, awarded.

F. Cross-artifact scholarly lineage

Zenodo's own DOI versioning already handles version lineage well within a single artifact's own history, and CodeMeta/Software Heritage [20] already capture code-level relationships between repositories. Research software registry best practices [20] likewise converge on strong single-registry governance. **What BQEB demonstrated:** a concrete, worked example (Fig. 2) chaining four real, independently-citable artifacts --- the architecture, BQEB-Data v1, BQEB ForecastBench, and ForecastBench's own checksummed v1.0.0 release artifacts --- each annotated with its real DOI and this paper's current, disclosed evidence tier for that specific artifact. **Integration contribution:** none of the tools above natively model this specific architecture → dataset → benchmark → benchmark-output chain, and none annotate each link with an evidence tier a reader can check against this paper's own validation sections. BQEB's Scientific Lineage Layer makes that chain itself an inspectable object, rather than a set of independently-citable DOIs a reader must manually locate and connect.

Practical implications. For researchers extending this architecture, the four-layer structure indicates where a new contribution belongs: a new dataset variant registers at the

Research Artifact Layer, a new model or metric registers within the existing registry mechanism described in Section VI, and a new benchmark module follows the roadmap slot pattern described in Section V. For benchmark developers evaluating whether to build on this architecture, the honest boundary stated above --- coherent design, unproven external applicability --- is the relevant fact to weigh before committing. For reviewers and independent replicators, the checksummed artifact set and the open governance log (Table IV) together specify what can currently be verified without contacting the author, and what cannot yet. For the broader energy AI community, this paper's practical implication is a caution rather than an endorsement: architecture-level reproducibility makes a benchmark's mechanics inspectable, but does not by itself certify that any one module's reported numbers are free of the kind of discrepancy disclosed in Table IV --- each module's own results should continue to be evaluated on their own reported evidence.

IX. LIMITATIONS

This section is additive to, not a replacement for, every limitation disclosed elsewhere in this paper.

Extensibility validated at $n=1$. The architecture's core generality claim rests on exactly one populated module, built by its own designer. This is stated in full in Section VIII and repeated here as a limitation, not only a discussion point.

ForecastBench demonstrates the architecture's mechanism, not its generality. These are distinct claims; only the first is currently supported.

Cross-OS determinism comparison is still pending, though related evidence has grown. Within-environment determinism, and the architecture's own CI now repeatedly succeeding outside the author's interactive session, must never be read as implying cross-OS determinism; genuinely comparing output across different operating systems and hardware is a separate, still-unexecuted claim (Section IV, Table V).

Statistical comparison is real but severely underpowered. The paired-comparison protocol has been executed against real ForecastBench test residuals (Section VII-A), not merely synthetic data, but $n=3$ is roughly 56 times smaller than the power analysis calls for; the non-significant result must be read as inconclusive, not as evidence that the models perform equivalently.

Independent third-party module registration remains future work. No genuine external test of the registry mechanism has occurred.

Formal automated test coverage of this program's own engineering tools is incomplete. One of six core tools has a repeatable, CI-discoverable test suite; the remainder are manually verified only.

The Scientific Lineage Layer's own version-tracking artifact has itself been found stale in prior internal audit, a limitation this paper discloses about the layer whose stated purpose is exactly this kind of tracking.

The ForecastBench discrepancy is now understood, but not reconciled. [5] and [6] were both retrieved in full this revision and agree with each other, so they are not the source of the discrepancy this paper originally flagged; the real, still-open gap is between that agreed-upon reported result and this program's own checksummed v1.0.0 release artifacts (Table IV). A dataset-scale mismatch is the leading hypothesis but is not confirmed by re-running the full computation.

Single-investigator provenance, synthetic dataset origin, and self-archived artifacts remain open limitations, unchanged by the lineage-separation mechanism described in Section III.

X. FUTURE RESEARCH DIRECTIONS

The architecture serves as the foundation layer for future BQEB research, all currently at planned status, none claimed complete. Five modules constitute this roadmap, grouped under the umbrella label **AI Energy Extensions** introduced in Section V --- the umbrella label organizes this list; it does not replace any entry within it. Table VI below gives each module's objective, the architecture layer(s) it is intended to exercise, and its intended evidence-validation plan, with status disclosed as Planned throughout:

1. **SecBench** --- a cyber-resilience benchmark.
2. **ForecastBench Foundation Models** --- an extension of the existing, first-validated forecasting module.
3. **Explainable BQEB** --- feature-attribution and interpretability methods for benchmark modules.
4. **Digital Twin & Multi-Agent AI** --- agent-based energy-system modeling.
5. **OptiGrid** --- an optimization-focused benchmark, distinct subject matter from the forecasting modules above.

No calendar commitments are implied by this roadmap. A disclosed, unresolved item this roadmap carries forward rather than omits: additional, differently-versioned "BQEB-Data"- and "BQEB-OptiData"-named identifiers exist whose relationship to BQEB-Data v1 remains unestablished. Any future dataset work under this roadmap should resolve that lineage question before building on it.

TABLE VI

BQEB MODULE ROADMAP, INCLUDING EACH MODULE'S OBJECTIVE, THE ARCHITECTURE LAYER(S) IT IS INTENDED TO EXERCISE, AND ITS INTENDED EVIDENCE-VALIDATION PLAN. NO CALENDAR DATES ARE IMPLIED; STATUS REFLECTS THE CURRENT STATE OF THIS PROGRAM, NOT A PROJECTION; THE EVIDENCE-VALIDATION PLANS DESCRIBE INTENDED FUTURE WORK AND ARE NOT PREDICTIONS OF OUTCOME.

Module	Status	Objective	Architecture layer(s) to be exercised	Intended evidence / validation plan
ForecastBench	Executed, validated	First downstream instantiation of the	All four layers (Research Artifact, Reproducibility,	Already executed, not a plan --- see Section VII-A/VII-B

Module	Status	Objective	Architecture layer(s) to be exercised	Intended evidence / validation plan
		architecture in the load/price forecasting domain, used to validate the four-layer design itself.	Benchmark Evolution, Scientific Lineage).	and Tables III and V for the specific evidence and its tier.
SecBench	Planned	A cyber-resilience benchmark: test how energy-AI modules (forecasting or control models) behave under adversarial or anomalous input conditions relevant to grid cybersecurity.	Benchmark Evolution Layer (new module registration); Reproducibility Layer (determinism and checksum discipline applied to adversarial-test artifacts).	Would apply the same determinism re-execution and checksum-verification protocol validated on ForecastBench (Section VII-A) to SecBench's own release artifacts once they exist; no evidence is claimed until that occurs.
ForecastBench Foundation Models	Planned	Extend the existing, first-validated forecasting module to foundation-model-based forecasters, testing whether the evidence-tier discipline validated on the current module scales to larger, pretrained architectures.	Reproducibility Layer (re-applying the determinism and statistical-comparison protocols); Benchmark Evolution Layer (module versioning).	Would re-run the determinism and statistical-comparison protocols already validated in Section VII-A against foundation-model forecasters, with the same power-analysis discipline applied before any significance claim.
Explainable BQEB	Planned	Add feature-attribution and interpretability methods as a cross-cutting capability over existing and future benchmark modules, rather than a new prediction or optimization task of its own.	Research Artifact Layer (new interpretability artifacts); Scientific Lineage Layer (linking attribution outputs to the specific model version that produced them).	Requires a validation protocol distinct from forecasting-accuracy metrics — likely stability/agreement checks of attributions across repeated runs — not yet designed in detail.
Digital Twin & Multi-Agent AI	Planned	Agent-based energy-system modeling with multiple interacting controllers or forecasters, testing the architecture at a system-of-systems scale rather than a single model.	All four layers in principle, at a substantially harder scale (multi-agent artifact registration, reproducibility, benchmark evolution, and lineage simultaneously).	No protocol has been designed yet; determinism across multiple interacting stochastic agents is a harder reproducibility problem than the single-model case validated in Section VII-A. Disclosed as an open design question, not an evidence gap pending execution.
OptiGrid	Planned	An optimization-focused benchmark (e.g., grid dispatch or storage scheduling), distinct subject matter from the forecasting modules above, testing whether the evidence-tier discipline generalizes from prediction to optimization tasks.	Benchmark Evolution Layer (new task family); Reproducibility Layer (determinism of optimization solvers, which have their own sources of non-determinism, e.g., solver tie-breaking).	Would apply the same determinism-first validation sequence used for ForecastBench, adapted for optimization-solver outputs rather than forecast residuals.

Every entry in Table VI above states an intended validation plan, not a prediction of its outcome: describing how a future module would be checked for determinism, integrity, or statistical validity is not a claim that it will pass those checks, and no module on this roadmap is claimed, implied, or scheduled to be complete by any date.

XI. CONCLUSION

This paper has presented the BQEB Reproducibility Architecture as a standalone, citable research-artifact framework (DOI: 10.5281/zenodo.22291930 [19]), with BQEB ForecastBench (DOI: 10.5281/zenodo.21735978)

positioned as its first validated downstream implementation. What this paper demonstrates, and what it does not, are stated explicitly below rather than left for a reader to infer from scattered qualifiers.

Demonstrated in this paper:

1. The four-layer architecture itself --- Research Artifact, Reproducibility, Benchmark Evolution, and Scientific Lineage --- as a described, partially-executed structure, not merely a proposal (Sections III-VI).

2. Within-environment reproducibility --- 15/15 bit-identical executions across 3 seeds, independently re-confirmed on multiple occasions including three additional runs for this revision, plus 12/12 successful re-executions by

the architecture's own continuous-integration workflow outside the author's interactive session (Sections IV, VII-A).

3. Actual ForecastBench artifact integrity --- all 17 files in the real v1.0.0 GitHub release checksum-matched their published hashes (17/17), with a genuine tamper-detect-restore cycle completed against one of those real release files (Section VII-A).

4. The real-data statistical-comparison workflow --- executed end-to-end against actual ForecastBench test residuals, though at a sample size roughly 56 times smaller than the comparison's own power analysis calls for, so its non-significant result is reported as inconclusive, not as evidence of model equivalence (Section VII-A).

5. Downstream ForecastBench CI evidence --- the downstream module's own continuous-integration workflow has passed 19/19 times across two operating systems (Section VII-A): real, executed, and currently the only evidence in this architecture's registry that spans more than one operating system, though it validates code quality (formatting, linting, type-checking, tests), not numerical determinism.

6. DOI-separated lineage across four real, independently-citable artifacts --- the architecture, BQEB-Data v1, BQEB ForecastBench, and ForecastBench's own checksummed release artifacts --- each annotated with its own current evidence tier rather than treated as uniformly validated (Fig. 2, Research Artifact Availability).

Remaining, disclosed as open rather than implied closed:

1. Independent, third-party extensibility --- the Benchmark Evolution Layer's registry has processed exactly one real module, built by this paper's own author, plus one internal synthetic test; genuinely independent registration by another party has not occurred (Sections V, VII-B, VIII).

2. Full architecture-level cross-environment validation --- the architecture's own CI now runs and passes repeatedly, but only within a single operating system; a genuine comparison of output across different operating systems and hardware remains unexecuted (Sections IV, VII-A, VII-B).

3. Adequately-powered statistical inference --- the executed comparison is real but underpowered by roughly 56-fold; its non-significant result cannot establish equivalence or rule out meaningful differences (Section VII-A).

4. Broader external adoption --- this architecture has one author, one downstream module, and no external users to date; claims of community uptake would be premature and are not made here.

The contribution is the architecture itself --- how a research program's artifacts can be structured, checksummed, and tracked for independent verification --- reported here at the evidence tier each claim actually supports. This paper does not claim, and should not be read as predicting, that this architecture or the evidence behind it will be judged sufficient by any reviewer or venue; it reports what was built and what was genuinely executed, and discloses the rest as open.

RESEARCH ARTIFACT AVAILABILITY

Architecture Artifact:

BQEB Reproducibility Architecture

GitHub: RakeshKumarAgrawal/BQEB-Reproducibility-Architecture

Zenodo DOI: 10.5281/zenodo.22291930 --- an independent, citable, archived research artifact. The software release is additionally indexed by Software Heritage, providing a further archival record independent of GitHub or Zenodo's own continuity.

Benchmark Artifact:

BQEB ForecastBench

GitHub: RakeshKumarAgrawal/BQEB-ForecastBench

Zenodo DOI: 10.5281/zenodo.21735978 --- an independent, citable, archived research artifact.

These two artifacts represent separate but related scholarly contributions. The architecture DOI identifies the foundational reproducibility framework described in this paper; the ForecastBench DOI identifies its first validated downstream benchmark implementation. Neither DOI is presented as validating the other's content, and neither identifier is evidence of reproducibility on its own --- a DOI is a persistent identifier, not a claim of verification. Citation metadata for the architecture repository is additionally provided in machine-readable form via its CITATION.cff file. The evidence supporting this paper's specific claims is presented in Sections IV and VII, independently of artifact identification.

Self-citation audit: six of this paper's twenty-four references are the author's own prior work: [4] (the BQEB conceptual monograph this architecture formally separates itself from, Section I), [5] and [6] (two archival copies of the same ForecastBench report, now confirmed identical, Section VII-B and Table IV), [16] (the BQEB-Data v1 dataset), [17] (the BQEB ForecastBench software release, this architecture's first downstream implementation), and [19] (this paper's own architecture software release, the same Zenodo DOI cited throughout as 10.5281/zenodo.22291930). None is cited for unrelated support or citation count: each identifies a specific, separately-versioned artifact this paper either builds on or formally separates itself from, which is the paper's own stated contribution (DOI-separated lineage, Section III; Fig. 2). Citing them is how that separation is made checkable rather than asserted. A reader weighing this paper's claims independently of its citation count should rely on Sections IV and VII's executed evidence, which does not depend on any of these six references being accepted at face value.

REFERENCES

- [1] J. Pineau, P. Vincent-Lamarre, K. Sinha, et al., "Improving reproducibility in machine learning research: A report from the NeurIPS 2019 reproducibility program," *Journal of Machine Learning Research*, vol. 22, no. 164, pp. 1--20, 2021.
- [2] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, "OpenML: Networked science in machine learning," *ACM SIGKDD Explorations Newsletter*, vol. 15, no. 2, pp. 49--60, 2014.

- [3] K. B. Debnath, M. Mourshed, and S. P. Chew, "Machine learning approaches for smart-grid forecasting: State-of-the-art and future directions," *IEEE Access*, vol. 9, pp. 152593--152620, 2021.
- [4] R. K. Agrawal, "BIO-Quantum Energy Brain (BQEB): A unified intelligence framework for future energy systems," Zenodo, 2026. DOI: 10.5281/zenodo.21780521.
- [5] R. K. Agrawal, "BQEB ForecastBench: Benchmarking AI models for smart grid forecasting using BQEB-Data v1," Research Square, preprint, 2026. DOI: 10.21203/rs.3.rs-10484554/v1.
- [6] R. K. Agrawal, "BQEB ForecastBench: Benchmarking AI models for smart grid forecasting using BQEB-Data v1," Zenodo, 2026. DOI: 10.5281/zenodo.19716383.
- [7] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 Competition: Results, findings, conclusion and way forward," *International Journal of Forecasting*, vol. 34, no. 4, pp. 802--808, 2018.
- [8] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M5 accuracy competition: Results, findings and conclusions," *International Journal of Forecasting*, vol. 38, no. 4, pp. 1346--1364, 2022.
- [9] T. Gebru, J. Morgenstern, B. Vecchione, et al., "Datasheets for datasets," *Communications of the ACM*, vol. 64, no. 12, pp. 86--92, 2021. DOI: 10.1145/3458723.
- [10] M. Mitchell, S. Wu, A. Zaldivar, et al., "Model cards for model reporting," in *Proc. Conf. Fairness, Accountability, and Transparency (FAT*)*, 2019, pp. 220--229. DOI: 10.1145/3287560.3287596.
- [11] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, et al., "The FAIR Guiding Principles for scientific data management and stewardship," *Scientific Data*, vol. 3, p. 160018, 2016. DOI: 10.1038/sdata.2016.18.
- [12] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *International Journal of Forecasting*, vol. 32, no. 3, pp. 914--938, 2016.
- [13] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *International Journal of Forecasting*, vol. 30, no. 4, pp. 1030--1081, 2014.
- [14] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5--32, 2001.
- [15] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189--1232, 2001.
- [16] R. K. Agrawal, "BQEB-Data v1: BIO-Quantum Energy Brain benchmark dataset for smart grid intelligence, renewable forecasting, storage optimization, and cyber resilience," Harvard Dataverse, 2026. DOI: 10.7910/DVN/VUVTED.
- [17] R. K. Agrawal, "BQEB ForecastBench: A reproducible smart-grid forecasting benchmark framework," software, v1.0.0, Zenodo, 2026. DOI: 10.5281/zenodo.21735978.
- [18] Association for Computing Machinery, "Artifact Review and Badging --- Current," ACM Publications Policies. [Online]. Available: <https://www.acm.org/publications/policies/artifact-review-and-badging-current>
- [19] R. K. Agrawal, "BQEB: A Reproducibility-First Architecture for Modular Energy AI Benchmarking," Zenodo, 2026. DOI: 10.5281/zenodo.22291930.
- [20] D. Garijo, H. Ménager, L. Hwang, A. Trisovic, M. Hucka, T. Morrell, and A. Allen, "Nine best practices for research software registries and repositories," *PeerJ Computer Science*, vol. 8, p. e1023, 2022. DOI: 10.7717/peerj-cs.1023.
- [21] Z. Fang, Y. Yuan, J. Zhang, Y. Liu, Y. Mu, Q. Lu, X. Xu, J. Wang, C. Wang, S. Zhang, and S. Chen, "MLOps spanning whole machine learning life cycle: A survey," *arXiv:2304.07296*, 2023. DOI: 10.48550/arXiv.2304.07296.
- [22] M. Herschel, R. Diestelkämper, and H. Ben Lahmar, "A survey on provenance: What for? What form? What from?," *The VLDB Journal*, vol. 26, no. 6, pp. 881--906, 2017. DOI: 10.1007/s00778-017-0486-1.
- [23] W. Oliveira, D. de Oliveira, and V. Braganholo, "Provenance analytics for workflow-based computational experiments: A survey," *ACM Computing Surveys*, vol. 51, no. 3, Article 53, 2018. DOI: 10.1145/3184900.
- [24] M. Barker, N. P. Chue Hong, D. S. Katz, A.-L. Lamprecht, C. Martinez-Ortiz, F. Psomopoulos, J. Harrow, L. J. Castro, M. Gruenpeter, P. A. Martinez, and T. Honeyman, "Introducing the FAIR Principles for

research software," *Scientific Data*, vol. 9, Article 622, 2022. DOI: 10.1038/s41597-022-01710-x.

RAKESH KUMAR AGRAWAL is a Senior Consultant at Atos and an applied researcher specializing in artificial intelligence, platform engineering, cloud engineering, and intelligent enterprise systems. His research interests include trustworthy AI, enterprise AI governance, agentic AI, LLMOps, AI benchmarking, digital twins, and intelligent energy systems. His work focuses on designing reliable, reproducible, and evidence-based AI systems for enterprise and critical-domain applications. He is a Senior Member of IEEE and a Fellow of IETE and holds the Project Management Professional (PMP®) credential, along with professional certifications in cloud and artificial intelligence. His research bridges AI theory, engineering practice, and responsible deployment of intelligent systems.