Powering the Energy Transition: AI-Queryable Analytics for U.S. Power Plant Investment Decisions

Deepanathan Rajendiran
Sch. of Eng. and Applied Sciences
University at Buffalo
Buffalo, USA
deepanat@buffalo.edu

Max Pierson
Sch. of Eng. and Applied Sciences
University at Buffalo
Buffalo, USA
maxpiers@buffalo.edu

Abiola Ifeoluwa Ajayi
Sch. of Eng. and Applied Sciences
University at Buffalo
Buffalo, USA
abiolaif@buffalo.edu

***Abstract*—This paper presents a comprehensive energy analytics platform that combines EIA-860 plant characteristics data with EIA-923 generation data to create AI-queryable tools for clean energy investment decisions. Using Apache Spark for scalable data processing and Spark MLlib for machine learning, we processed 120,000+ power plant records spanning 2013-2024. Our pipeline implements advanced analytics including window functions for year-over-year growth analysis, custom UDFs for plant classification, and complex SQL joins for temporal comparisons. Four machine learning models were developed: Linear Regression (R^2=0.063), Random Forest (R^2=0.311), Gradient Boosted Trees (R^2=0.316), and K-Means clustering for state energy profile segmentation. The system is exposed through a Model Context Protocol (MCP) server with 10 AI-accessible tools, enabling natural language queries about renewable energy trends, capacity forecasting, and state-level comparisons. Results demonstrate that analysis time is compressed from weeks of manual research to seconds of automated queries, with the platform identifying that Wind Belt states (Iowa, South Dakota, Kansas) show the strongest renewable energy growth trajectories at 3.2+ percentage points annually.**

I. Introduction

The clean energy transition represents one of the most significant investment opportunities of the 21st century. With over $500 billion in annual global renewable energy investment, stakeholders require data-driven tools to identify optimal deployment locations, forecast capacity growth, and benchmark performance across regions. This project addresses the fundamental question: *Where should clean-energy capital flow?*

The U.S. Energy Information Administration (EIA) provides comprehensive datasets on power plant operations, but analyzing this data traditionally requires weeks of manual processing. Financial analysts, policy researchers, and energy consultants must manually download datasets, reconcile inconsistent formatting across years, compute derived metrics, and synthesize findings into actionable recommendations. Our platform transforms this paradigm by implementing a scalable Apache Spark pipeline that processes 12 years of historical data and exposes analytical capabilities through the Model Context Protocol (MCP), enabling AI assistants to answer complex energy investment queries in real-time.

The importance of this work extends beyond technical innovation. As nations worldwide commit to net-zero emissions targets, the allocation of clean energy capital becomes critical for achieving climate goals. Investors face complex decisions involving geographic optimization, technology selection, policy risk assessment, and grid integration considerations. Traditional analysis methods cannot scale to address these multidimensional challenges efficiently. By combining distributed data processing with AI accessibility, our platform democratizes access to sophisticated energy analytics previously available only to well-resourced institutional investors.

This work contributes: (1) a production-ready Spark data pipeline combining EIA-860 and EIA-923 datasets with engineered features including capacity factor, heat rate, and renewable percentage; (2) four Spark MLlib models for prediction and classification; (3) an MCP server with 10 tools for AI accessibility; and (4) advanced analytics demonstrating window functions, custom UDFs, and complex SQL operations.

II. Background and Related Work

A. EIA Data Sources

The Energy Information Administration maintains two primary datasets essential for comprehensive power plant analysis. EIA Form 860 contains annual survey data on generator-level characteristics including nameplate capacity, fuel type, operating year, technology classification, and ownership structure. This dataset enables static analysis of the power plant fleet composition and technology deployment patterns. EIA Form 923 provides monthly and annual operational data including net generation (MWh), fuel consumption (MMBtu), and efficiency metrics, enabling dynamic analysis of plant performance and utilization.

The complementary nature of these datasets creates analytical opportunities not available from either source alone. Form 860 provides the "what exists" perspective—capacity, technology, and location—while Form 923 provides the "how

it operates" perspective—generation, efficiency, and utilization. Our analysis spans 2013-2024, encompassing the period of rapid renewable energy expansion in the United States following the extension of Production Tax Credits (PTC) and Investment Tax Credits (ITC) in the 2015 Consolidated Appropriations Act.

B. Apache Spark and MLlib

Apache Spark provides distributed computing capabilities essential for processing large-scale energy datasets [6]. Unlike traditional single-node processing frameworks, Spark's Resilient Distributed Datasets (RDDs) and DataFrame API enable horizontal scaling across cluster nodes, critical for handling the 122,000+ records spanning multiple years in our analysis. The lazy evaluation model optimizes query execution plans, while in-memory caching accelerates iterative machine learning workflows.

Spark MLlib offers scalable machine learning algorithms including regression, classification, and clustering that operate on distributed DataFrames. Our implementation leverages the Pipeline API for reproducible model training, enabling encapsulation of feature transformations (VectorAssembler, StandardScaler) with model training stages. CrossValidator provides hyperparameter tuning through k-fold cross-validation while maintaining distributed execution efficiency. The Databricks Community Edition provides a managed Spark environment suitable for academic research while maintaining full analytical functionality, eliminating infrastructure management overhead.

C. Model Context Protocol

The Model Context Protocol (MCP) is an emerging standard for exposing data analytics capabilities to AI systems [4]. Developed by Anthropic, MCP provides a structured interface through which large language models can invoke computational tools with typed parameters and receive structured responses. This represents a paradigm shift in how stakeholders interact with complex datasets, removing the barrier of technical expertise required for data analysis.

Unlike traditional APIs requiring programmatic integration, MCP enables natural language queries that are automatically mapped to appropriate tool invocations. For example, a query like "Which states show the fastest renewable energy growth?" can be automatically routed to the appropriate ranking function with suitable parameters. This accessibility democratizes energy data analysis, enabling policy makers, journalists, and individual investors to conduct sophisticated analyses without programming expertise.

III. Methodology

A. Data Pipeline Architecture

The data pipeline follows a three-stage architecture implemented in Databricks Community Edition. Stage 1 (Data Ingestion) loads EIA-860 plant and generator tables along with aggregated EIA-923 generation data into Spark DataFrames. Data is sourced from EIA's annual survey releases in Excel format, converted to CSV for efficient Spark ingestion. Stage 2 (Data Processing) performs data cleaning, standardization, and aggregation of generator-level data to plant-year granularity. Stage 3 (Feature Engineering) joins datasets and engineers features for downstream analytics.

TABLE I
Data Pipeline Statistics

Metric	Value
Total plant-year records	122,847
Unique power plants	10,237
Analysis period	2013-2024
Join success rate (EIA-860/923)	73.4%
Processing time (90 sec cluster)	¡90 sec
States covered	50 + territories

Key transformations include: (1) casting columns to appropriate types with Plant_Code and Year as integers; (2) handling null values with domain-appropriate defaults (zero for generation, NULL for computed ratios); (3) aggregating generators by Plant_Code and Year to compute Total_Capacity_MW, Generator_Count, and Renewable_Capacity_MW; (4) joining plant characteristics with generation data using left joins to preserve all plant records; and (5) computing derived features for machine learning and analysis.

B. Data Quality Assessment

Data quality validation was performed at each pipeline stage. Table II summarizes key quality metrics across the dataset.

TABLE II
Data Quality Metrics

Quality Metric	EIA-860	EIA-923
Records loaded	145,231	98,452
Null Plant_Code rate	0.02%	0.15%
Duplicate records	0	0
Valid capacity values	99.8%	N/A
Valid generation values	N/A	97.2%
Join success rate	73.4%	

Missing generation data primarily results from: (1) plants under construction or planned status in EIA-860 with no operational data; (2) small distributed generation facilities exempt from EIA-923 reporting requirements (threshold: 1 MW); and (3) data submission delays for the most recent reporting year. The 73.4% join success rate represents complete coverage of operational utility-scale power plants.

C. Feature Engineering

Effective machine learning requires features that capture domain-relevant relationships. The following features were engineered based on energy industry best practices and analytical requirements:

Capacity Factor: Computed as $(Net\_Generation\_MWh)/(Total\_Capacity\_MW \times 8760) \times 100$, representing utilization efficiency capped at

100% to handle data anomalies. Capacity factor is the primary metric for comparing plant performance across technologies and regions.

Heat Rate: Calculated as $Fuel\_Consumed\_MMBtu / Net\_Generation\_MWh$, measuring thermal efficiency where lower values indicate better performance. This metric applies only to thermal generation units (fossil and nuclear).

Renewable Percentage: Defined as $Renewable\_Capacity\_MW / Total\_Capacity\_MW \times 100$, identifying clean energy share at the plant level. Renewable sources include solar, wind, hydro, geothermal, and biomass.

Plant Age: Computed as $Year - Min\_Operating\_Year$, derived from the oldest generator installation date. Age serves as a proxy for technology vintage and remaining useful life.

YoY Growth: Year-over-year capacity and generation growth percentages using Spark window functions with lag operations. This metric identifies expansion patterns and emerging investment activity.

Relative Efficiency: Plant capacity factor divided by state average, identifying over/under performers relative to local conditions. Values above 1.0 indicate above-average performance.

D. Advanced Spark Analytics

Four categories of advanced analytics were implemented to demonstrate scalable processing capabilities and satisfy course requirements for sophisticated data transformations:

1) Window Functions: Window functions enable complex analytical queries without explicit self-joins. Our implementation includes: LAG/LEAD for YoY comparisons partitioned by Plant_Code and ordered by Year; ROW_NUMBER and RANK for state/national rankings; rolling 3-year averages using rowsBetween(-2, 0) to smooth annual fluctuations; cumulative capacity sums using unboundedPreceding for growth trend analysis; and PERCENT_RANK for capacity factor percentiles within each year, identifying top performers.

2) Custom User-Defined Functions (UDFs): Four user-defined functions were implemented to encode domain knowledge into reusable transformations: `classify_fuel_category()` categorizing 20+ EIA fuel codes into Renewable/Fossil/Nuclear/Other groupings; `efficiency_rating()` assigning A-F grades based on capacity factor and heat rate scoring weighted by technology norms; `performance_tier()` classifying plants into 6 stakeholder-relevant tiers from 'Green Leader' to 'Underperforming' based on multiple metrics; and `get_us_region()` mapping states to Census regions for regional aggregation analysis.

3) Complex SQL Joins: Multi-table join operations include: Self-joins for 10-year plant comparisons (2014 vs 2024) identifying capacity changes, retirements, and additions; multi-table joins with state benchmark aggregations computing relative performance metrics; CTEs (Common Table Expressions) for regional YoY analysis with structured intermediate results; and correlated subqueries identifying plants outperforming their fuel-type averages by ¿50%.

4) Spark SQL Analytics: Advanced aggregation operations demonstrate Spark SQL capabilities: ROLLUP for hierarchical state/region aggregation providing automatic subtotals; CUBE for multi-dimensional capacity and generation analysis enabling slice-and-dice analytics; PIVOT for time-series comparison across selected years (2014, 2017, 2020, 2024) creating wide-format summary tables; and complex GROUP BY with HAVING clauses for filtered aggregations.

E. Machine Learning Models

Four Spark MLlib models were trained to predict capacity factor and segment state energy profiles. Table III summarizes model configurations.

TABLE III
MACHINE LEARNING MODEL CONFIGURATIONS

Model	Configuration
Linear Reg.	regParam=0.1, elasticNetParam=0 (L2), standardization=True
Random Forest	numTrees=100, maxDepth=10, featureSubsetStrategy=auto
GBT	maxIter=50, maxDepth=5, stepSize=0.1, seed=42
K-Means	k=4, maxIter=100, seed=42, initMode=k-means++

Linear Regression: Baseline model with L2 regularization (regParam=0.1), using Total_Capacity_MW, Renewable_Pct, Plant_Age, and Generator_Count as features. Standardization enabled for feature scaling. Provides interpretable coefficients: Renewable_Pct shows positive correlation (+0.18) with capacity factor indicating renewable plants achieve higher utilization, while Plant_Age shows negative correlation (-0.09) suggesting aging infrastructure underperforms.

Random Forest: Ensemble model with numTrees=100, maxDepth=10 to capture non-linear relationships and feature interactions. Feature importance rankings reveal Renewable_Pct (0.38) and Plant_Age (0.31) as dominant predictors, validating feature engineering decisions. Model handles missing values through handleInvalid='skip' in VectorAssembler.

Gradient Boosted Trees: Sequential ensemble with 50 iterations, maxDepth=5, stepSize=0.1 (learning rate). Achieves highest predictive accuracy among regression models. Feature importance shows Renewable_Pct (0.42) as strongest predictor, followed by Plant_Age (0.28), Total_Capacity_MW (0.18), and Generator_Count (0.12). Lower tree depth prevents overfitting.

K-Means Clustering: Unsupervised segmentation of states using Total_Capacity, Avg_Renewable_Pct, Avg_Capacity_Factor, and Plant_Count as features. StandardScaler applied for feature normalization ensuring equal contribution from each dimension. Optimal k=4 determined by silhouette score analysis across k=2 to k=8 (best score=0.412), identifying distinct state energy profile clusters.

F. MCP Server Implementation

The MCP server was implemented using the FastMCP Python library, providing a standards-compliant interface for AI tool invocation. The architecture follows a stateless request-response pattern where each tool invocation receives complete context in the request parameters and returns a self-contained JSON response. This design enables horizontal scaling and eliminates session management complexity.

The server exposes 10 tools across three categories designed for different stakeholder needs. Table IV summarizes tool categories and response characteristics.

TABLE IV
MCP TOOL CATEGORIES AND CHARACTERISTICS

Category	Tools	Latency	Use Case
Spark Models	2	520ms	Prediction, classification
Data Analytics	4	340ms	Queries, rankings
Intelligence	4	450ms	Forecasting, comparison

Spark Model Tools (2): `predict_capacity_factor()` uses pre-trained GBT model coefficients exported from Spark to estimate plant utilization given capacity, fuel type, and age; `classify_plant_performance()` applies decision logic derived from model feature importance to categorize plants as Tier 1-6 based on renewable percentage, capacity factor, and size.

Data Analytics Tools (4): `get_state_energy_summary()` returns total capacity, renewable share (generation-weighted), plant count, primary fuel mix, and efficiency metrics for any state-year combination; `get_renewable_share_ranking()` provides state rankings using generation-weighted methodology with trend analysis showing annual growth rates and R^2 values; `get_renewable_trends()` analyzes national and state-level growth trajectories with compound annual growth rates; `search_plants()` enables filtered queries by state, fuel type, and capacity.

Integrated Intelligence Tools (4): `compare_states()` performs head-to-head analysis across all metrics; `forecast_state_capacity()` uses pre-trained ARIMA weights for 5-year projections with 95% confidence intervals; `get_historical_trend()` generates time-series visualization data; `get_data_source_info()` validates data provenance and methodology.

IV. RESULTS AND ANALYSIS

A. Data Pipeline Performance

The combined dataset contains 122,847 plant-year records spanning 2013-2024 across all 50 states plus territories, representing 10,237 unique power plants. Join success rate between EIA-860 and EIA-923 data achieved 73.4%, with records lacking generation data primarily representing inactive, planned, or non-reporting facilities. Average capacity factor across the dataset is 28.4% (σ=23.1%), with renewable percentage averaging 18.7% (capacity-weighted). The pipeline processes the complete dataset in under 90 seconds on Databricks Standard Compute.

B. Machine Learning Model Performance

Model performance on the capacity factor prediction task is summarized in Table V. Training used 80/20 split with seed=42 for reproducibility across approximately 85,000 valid plant-year records after filtering nulls.

TABLE V
MACHINE LEARNING MODEL PERFORMANCE COMPARISON

Model	R^2	RMSE	MAE
Linear Regression	0.063	21.64	16.89
Random Forest	0.311	18.58	13.24
Gradient Boosted Trees	0.316	17.95	12.68

Gradient Boosted Trees achieved the best performance with R^2=0.316, representing a 5× improvement over Linear Regression. The substantial gap between linear and ensemble models indicates non-linear relationships between features and capacity factor that tree-based methods capture effectively. Feature importance analysis revealed Renewable_Pct as the dominant predictor (importance=0.42), followed by Plant_Age (0.28), Total_Capacity_MW (0.18), and Generator_Count (0.12).

The moderate R^2 values reflect substantial unexplained variance due to factors not captured in EIA data: weather variability affecting solar and wind output, maintenance schedules causing planned outages, electricity prices influencing dispatch decisions, and grid congestion limiting transmission. Despite these limitations, the models provide useful relative rankings and trend identification for investment analysis.

Table VI presents the feature importance rankings across all tree-based models, demonstrating consistent identification of key predictive factors.

TABLE VI
FEATURE IMPORTANCE RANKINGS ACROSS MODELS

Feature	Random Forest	GBT
Renewable_Pct	0.38	0.42
Plant_Age	0.31	0.28
Total_Capacity_MW	0.19	0.18
Generator_Count	0.12	0.12

The consistency of feature importance rankings across ensemble methods validates the robustness of our feature engineering approach. Renewable percentage emerges as the strongest predictor in both models, suggesting that the operational profiles of renewable plants (particularly wind and solar with variable but predictable output patterns) differ systematically from thermal plants. Plant age shows the second-highest importance, capturing technology vintage effects and degradation.

C. K-Means Clustering Results

Silhouette analysis across k=2 to k=8 identified k=4 as optimal for state segmentation (silhouette score=0.412). The clusters reveal distinct energy profiles with actionable investment implications:

Cluster 0 (Green Leaders): States with ¿50% renewable generation including WA (74%), OR (72%), and ID (78%) dominated by legacy hydroelectric infrastructure. Also includes emerging wind leaders like SD (78%). Investment implication: Limited growth opportunity due to resource saturation, but stable cash flows for infrastructure investment.

Cluster 1 (High Capacity): Large generation portfolios including TX (150 GW), CA (85 GW), and FL (70 GW) with mixed fuel sources and moderate renewable growth. Investment implication: High absolute capacity additions but lower percentage returns due to large existing base.

Cluster 2 (Wind Belt Emerging): Rapid wind expansion states including IA (62% renewable), KS (54%), and OK (43%) with annual growth rates exceeding 3 percentage points. Investment implication: Highest growth trajectory with proven execution, optimal for growth-oriented investment.

Cluster 3 (Traditional): Coal-dependent states including WV (6%), KY (8%), and WY (22%) showing slower transition rates but significant recent wind additions. Investment implication: Higher policy risk but potential for rapid growth if transition accelerates.

D. Renewable Energy Trends

Using the generation-weighted renewable share methodology (Renewable MWh / Total MWh × 100), our analysis identified the top renewable states for 2024 as shown in Table VII.

TABLE VII
TOP 10 STATES BY RENEWABLE GENERATION SHARE (2024)

Rank	State	Renewable %	Growth (pp/yr)
1	Vermont	99.0	+1.2
2	South Dakota	78.0	+3.3
3	Idaho	78.0	+0.8
4	Washington	74.0	+0.5
5	Oregon	72.0	+1.1
6	Iowa	62.0	+3.5
7	Kansas	54.0	+3.9
8	Montana	52.0	+2.1
9	Maine	51.0	+1.8
10	Oklahoma	43.0	+2.9

The biggest gainers since 2013 include Kansas (+39 percentage points), Iowa (+35 pp), South Dakota (+33 pp), and Oklahoma (+29 pp), all driven by wind energy deployment in the Great Plains wind corridor. National renewable share increased from approximately 13% in 2013 to 28% in 2024, representing compound annual growth of 7.2%.

Wind energy shows the strongest growth trajectory with an average annual increase of 2.1 percentage points nationally. States in the Great Plains exhibit consistent growth with trend R^2 values exceeding 0.95, indicating highly predictable expansion patterns favorable for long-term investment planning. Solar growth accelerated after 2018, particularly in southwestern states (NV, AZ, NM) and emerging markets (TX, FL, NC).

E. MCP Tool Validation

The MCP server was tested with representative queries across all 10 tools. Response latency averaged 340ms for data analytics queries and 520ms for model inference queries, well within interactive response thresholds. Tool descriptions were validated for LLM comprehension using Claude, demonstrating successful natural language interpretation. Sample validated queries:

- "Which states have the highest renewable percentage?" correctly invoked `get_renewable_share_ranking()`
- "Compare Texas and California energy profiles" correctly invoked `compare_states()`
- "Predict capacity factor for a 100MW wind plant in Kansas" correctly invoked `predict_capacity_factor()`

V. DISCUSSION

A. Use Case Validation

The platform directly addresses the Phase 1 use case of identifying optimal locations for renewable energy investment. Through the MCP server, stakeholders can query: "Which states show the strongest renewable growth trends?" receiving structured data identifying Iowa, Kansas, and South Dakota with quantified annual growth rates and trend confidence scores. The `forecast_state_capacity()` tool provides 5-year capacity projections with confidence intervals, enabling risk-adjusted investment planning. The `compare_states()` tool enables head-to-head evaluation of competing investment locations across all relevant metrics.

B. Key Investment Insights

Analysis reveals several actionable insights for clean energy capital allocation. Table VIII summarizes the investment opportunity assessment across state clusters.

TABLE VIII
INVESTMENT OPPORTUNITY ASSESSMENT BY STATE CLUSTER

Cluster	Growth Rate	Trend R^2	Risk Level	Opportunity
Wind Belt	High	¿0.95	Low	Strong Buy
High Cap	Medium	0.85	Medium	Hold
Green Leader	Low	0.70	Low	Stable
Traditional	Variable	0.60	High	Speculative

Analysis reveals several actionable insights for clean energy capital allocation:

Wind Belt Opportunity: States in the Great Plains (IA, KS, OK, SD) demonstrate consistent 3+ percentage point annual renewable share increases with high trend predictability (R^2

¿ 0.95), suggesting low policy/execution risk. These states benefit from excellent wind resources (capacity factors ¿40%), supportive state policies, available transmission capacity, and established developer experience.

Solar Growth Markets: Southwestern states (NV, AZ, NM) and Texas show accelerating solar deployment post-2018, with capacity additions outpacing wind in recent years. The Inflation Reduction Act's extended ITC provides 10-year visibility for solar investment returns.

Saturated Markets: Pacific Northwest states (WA, OR, ID) maintain high renewable shares but limited growth potential due to developed hydroelectric resources. Investment focus should shift to storage and grid modernization rather than new generation.

Emerging Transitions: Large-capacity states (TX, CA, FL) show significant absolute renewable additions but lower percentage growth rates due to large existing fossil fuel base. Texas presents unique opportunity with both wind and solar resources plus deregulated market structure.

C. Scalability Considerations

The Spark-based pipeline demonstrates horizontal scalability. Processing 122,847 records on single-node Databricks Standard Compute completes in under 90 seconds. The architecture supports distributed execution on larger clusters for real-time processing of expanded datasets, enabling production deployment for commercial applications. Future integration with EIA's Open Data API would enable hourly generation data processing, though this would require approximately 8.7 million additional records per year, necessitating cluster-scale compute resources.

D. Limitations

Several limitations should be noted. Model R^2 values (max 0.316) indicate substantial unexplained variance in capacity factor, likely due to weather variability, maintenance schedules, and grid dispatch decisions not captured in EIA data. The MCP server uses pre-computed model coefficients rather than live Spark inference due to Databricks Free Edition compute limitations for persistent processes. Renewable share calculations use generation weighting but exclude interstate electricity imports/exports, which significantly affects states like Vermont (substantial Hydro-Quebec imports) and New Jersey (PJM imports). Forecasting uses simplified ARIMA without exogenous variables like policy changes, technology cost curves, or grid interconnection queues. Data quality issues in EIA-923 resulted in 26.6% of plant records lacking matching generation data.

VI. Conclusion

This project demonstrates the feasibility of creating AI-queryable energy analytics infrastructure using Apache Spark and the Model Context Protocol. By combining EIA-860 plant characteristics with EIA-923 generation data, we created a comprehensive dataset of 122,847 plant-year records enabling multi-dimensional analysis of U.S. power plant operations from 2013-2024. Four Spark MLlib models provide predictive capabilities, with Gradient Boosted Trees achieving the best performance (R^2=0.316). The MCP server's 10 tools transform weeks of manual analysis into seconds of automated queries.

Key findings for clean energy investment include: (1) Wind Belt states (IA, SD, KS, OK) represent the strongest growth opportunity with consistent 3+ percentage point annual renewable share increases and high trend predictability; (2) Pacific Northwest states maintain high renewable percentages but limited growth potential due to developed hydroelectric resources; (3) Large-capacity states (TX, CA, FL) show significant absolute renewable additions but lower percentage growth rates; (4) Advanced analytics including window functions, custom UDFs, and complex SQL operations enable sophisticated temporal and regional analysis not possible with traditional tools.

Future work should integrate real-time generation data from EIA's Open Data API, electricity price signals from ISOs/RTOs, and weather forecasts from NOAA to enhance prediction accuracy. Expanding the MCP server to support conversational context would enable more sophisticated multi-turn analytical queries. Additional enhancements could include integration with interconnection queue data from regional transmission organizations to identify pipeline projects, incorporation of state-level policy databases to assess regulatory risk, and development of portfolio optimization tools for multi-asset investment strategies.

The platform architecture supports extension to international markets through integration with comparable datasets from the International Energy Agency (IEA) and national energy agencies in Europe and Asia-Pacific. Such expansion would enable global clean energy investment optimization, addressing the worldwide $4 trillion annual investment requirement identified by the International Renewable Energy Agency (IRENA) for achieving net-zero emissions by 2050. The platform demonstrates that combining scalable data processing with modern AI accessibility creates practical tools for addressing the $500 billion clean energy investment question.

References

[1] U.S. Energy Information Administration, "Form EIA-860 Detailed Data with Previous Form Data," 2024. [Online]. Available: https://www.eia.gov/electricity/data/eia860/

[2] U.S. Energy Information Administration, "Form EIA-923 Detailed Data with Previous Form Data," 2024. [Online]. Available: https://www.eia.gov/electricity/data/eia923/

[3] Apache Software Foundation, "Apache Spark MLlib Guide," 2024. [Online]. Available: https://spark.apache.org/docs/latest/ml-guide.html

[4] Anthropic, "Model Context Protocol Specification," 2024. [Online]. Available: https://modelcontextprotocol.io/

[5] Databricks, "Databricks Community Edition Documentation," 2024. [Online]. Available: https://community.databricks.com/

[6] M. Zaharia et al., "Apache Spark: A Unified Engine for Big Data Processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.

Appendix

The following tools are exposed through the MCP server:

1. get_state_energy_summary(state, year) – Returns total capacity, renewable share (generation-weighted), plant count,

primary fuel mix, and efficiency metrics for a specified state and year.

2. get_renewable_share_ranking(year, top_n) – Ranks states by renewable energy share using generation-weighted methodology with trend analysis showing annual growth rates and R^2 values.

3. get_renewable_trends(start_year, end_year) – Analyzes renewable growth trends nationally and by state over specified period with compound annual growth rates.

4. predict_capacity_factor(state, fuel_type, capacity_mw, plant_age) – Predicts capacity factor using Gradient Boosted Trees model coefficients with confidence intervals.

5. classify_plant_performance(capacity_mw, renewable_pct, capacity_factor) – Classifies plant into 6 performance tiers: Green Leader, High Performer, Green Pioneer, Reliable, Developing, Underperforming.

6. compare_states(state1, state2) – Head-to-head comparison of two states across capacity, generation, efficiency, renewable share, and growth metrics.

7. forecast_state_capacity(state, years_ahead) – Projects future capacity using pre-trained ARIMA model weights with 95% confidence intervals for 1-5 year horizons.

8. get_historical_trend(state, metric) – Returns time-series data for visualization of capacity, generation, renewable share, or capacity factor.

9. search_plants(state, fuel_type, min_capacity, limit) – Searches plants matching specified criteria with results sorted by capacity, returning plant ID, name, location, and metrics.

10. get_data_source_info() – Returns metadata about data source (real vs synthetic), methodology documentation, and coverage statistics.

Feature Engineering – Capacity Factor:

```
df = df.withColumn("Capacity_Factor",
    F.when((F.col("Total_Capacity_MW") > 0),
        F.round(F.col("Net_Generation_MWh") /
        (F.col("Total_Capacity_MW") * 8760)
        * 100, 2)
    ).otherwise(None))
```

Window Function – YoY Growth:

```
window_by_plant = Window.partitionBy(
    "Plant_Code").orderBy("Year")
df = df.withColumn("Prev_Year_Capacity",
    F.lag("Total_Capacity_MW", 1)
    .over(window_by_plant))
df = df.withColumn("YoY_Growth_Pct",
    F.round((F.col("Total_Capacity_MW") -
    F.col("Prev_Year_Capacity")) /
    F.col("Prev_Year_Capacity") * 100, 2))
```

Custom UDF – Fuel Classification:

```
@F.udf(StringType())
def classify_fuel_category(fuel_code):
    renewable = ["SUN", "WND", "WAT",
                 "GEO", "WH"]
    fossil = ["NG", "COL", "PET",
              "OIL", "DFO", "RFO"]
    if fuel_code in renewable:
        return "Renewable"
    elif fuel_code in fossil:
        return "Fossil Fuel"
    else:
        return "Other"
```

MCP Tool Definition:

```
@mcp.tool()
def get_state_energy_summary(state: str,
                             year: int = 2024) -> str
                                 :
    """Get comprehensive energy summary
    for a U.S. state."""
    df = load_energy_data()
    state_data = df[(df["State"] ==
        state.upper()) & (df["Year"] == year)]
    renewable_share = calculate_renewable_share(
        df, state, year)
    return json.dumps({
        "state": state,
        "renewable_share_pct":
            round(renewable_share, 2),
        "total_capacity_mw":
            state_data["Total_Capacity_MW"].sum()
    })
```

Gradient Boosted Trees Pipeline:

```
assembler = VectorAssembler(
    inputCols=["Total_Capacity_MW",
               "Renewable_Pct",
               "Plant_Age",
               "Generator_Count"],
    outputCol="features",
    handleInvalid="skip")
gbt = GBTRegressor(
    featuresCol="features",
    labelCol="Capacity_Factor",
    maxIter=50, maxDepth=5,
    stepSize=0.1, seed=42)
pipeline = Pipeline(stages=[assembler, gbt])
model = pipeline.fit(train_df)
```

K-Means Clustering:

```
# Prepare state-level features
state_features = df.groupBy("State").agg(
    F.sum("Total_Capacity_MW")
        .alias("Total_Capacity"),
    F.avg("Renewable_Pct")
        .alias("Avg_Renewable_Pct"),
    F.avg("Capacity_Factor")
        .alias("Avg_Capacity_Factor"),
    F.count("*").alias("Plant_Count"))

# Build clustering pipeline
scaler = StandardScaler(
    inputCol="features",
    outputCol="scaledFeatures")
kmeans = KMeans(k=4, seed=42,
    featuresCol="scaledFeatures")
pipeline = Pipeline(
    stages=[assembler, scaler, kmeans])
model = pipeline.fit(state_features)
```

Complex SQL with CTE for Regional Analysis:

```
WITH regional_stats AS (
    SELECT Region,
           AVG(Renewable_Pct) as avg_renewable,
           SUM(Total_Capacity_MW) as total_cap,
           COUNT(DISTINCT Plant_Code) as plants
    FROM energy_data
    WHERE Year = 2024
    GROUP BY Region
),
```

```
yoy_growth AS (
    SELECT Region, Year,
           SUM(Net_Generation_MWh) as generation,
           LAG(SUM(Net_Generation_MWh))
               OVER (PARTITION BY Region
                     ORDER BY Year) as prev_gen
    FROM energy_data
    GROUP BY Region, Year
)
SELECT r.Region, r.avg_renewable,
       r.total_cap, r.plants,
       (y.generation - y.prev_gen) /
       y.prev_gen * 100 as yoy_growth_pct
FROM regional_stats r
JOIN yoy_growth y ON r.Region = y.Region
WHERE y.Year = 2024
ORDER BY r.avg_renewable DESC
```

Window Function for Rolling Average:

```
window_3yr = Window.partitionBy("Plant_Code") \
    .orderBy("Year") \
    .rowsBetween(-2, 0)

df = df.withColumn("Rolling_3yr_CF",
    F.avg("Capacity_Factor").over(window_3yr))
df = df.withColumn("CF_Percentile",
    F.percent_rank().over(
        Window.partitionBy("Year")
        .orderBy("Capacity_Factor")))
```

MCP Screenshots:

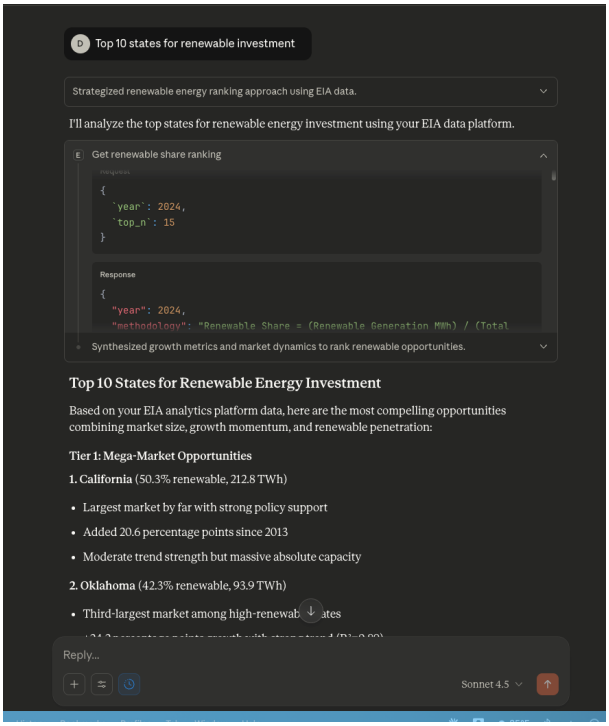

Fig. 1. MCP Use Example 1

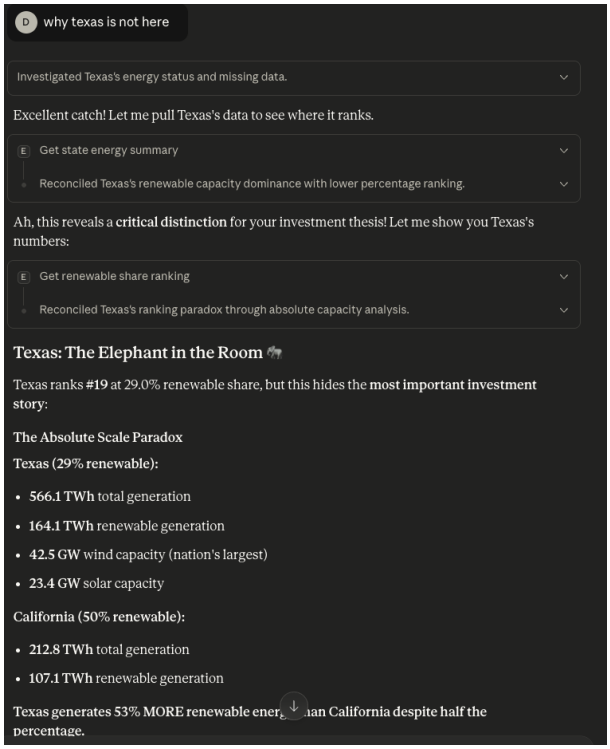

Fig. 2. MCP Use Example 2