Smart Grid Irregularity Detection in Electrical Power Systems

Dr. B. Shankar Nayak[1], E. Archana[2], M. Sai Revanth[3], B. Kavya[4]
[1]*Assistant Professor, Dept. of CSE (Data Science), CMR Technical Campus*
[2,3,4]*Student, Dept. of CSE (Data Science), CMR Technical Campus*

***Abstract*—Smart grids are dealing with some tough challenges because household energy use can be pretty unpredictable appliances turning on and off at strange hours, and people using energy in different ways. Instead of just relying on consistent usage patterns, this tool keeps an eye on electricity usage in real-time, using K-Means clustering to identify unusual activities like power surges, leaks, or tampering. Unlike traditional monthly bills that often overlook these issues, this tool dives into the specifics for each device to catch problems quickly. If something seems off, users get alerts via SMS, and they can view the findings on a visual dashboard created with Streamlit. Reports are accessible anytime, and the breakdowns go deeper than just totals users can see details for individual appliances. Plus, updates happen every second. This method was tested with both simulated data and some actual small-scale data, and it showed impressive results in identifying issues. For power companies and consumers looking to manage energy flow better, reduce waste, or stay more engaged, this solution really shines with its hands-on approach.**

***Index Terms*—Smart Grid, Anomaly Detection, Unsupervised Learning, K-Means Clustering, Power Systems, Non-Technical Losses, Streamlit, Real-Time Monitoring.**

I. INTROUCTION

Smart grids are evolving, yet each upgrade brings fresh chaos when new houses connect with chatty IoT devices spewing constant updates. Usage trends flip overnight - lifestyle changes, gadgets come and go, pricing shifts - while legacy billing tech lags behind. These outdated setups fail to catch power theft, odd surges, or faulty meters losing accuracy. Trouble explodes as floodgates open: millions of data points crash in by the minute. Systems based on basic number games collapse when stressed - since fixed guidelines rarely fit how things actually work, spotting problems becomes a shot in the dark. Responses come too slow. Cash slips away. The network stumbles along. Recent research highlights similar concerns. Sida's team [1], using multiple analysis methods, found more consistent fault spotting in electricity data. Instead of just one approach, combining them helped reduce mistakes. Moving on, Sial's crew [2] dug into patterns from smart meters, uncovering clues about when households change energy habits. Their work reveals subtle shifts that raw numbers often miss. As for Lei's researchers [3], they linked personal choices with live device inputs via a test system - imaginative but shaky, given how erratic real-world buildings act. On another front, Jahangir and others [4] looked at networked grids powered by sensors, zeroing in on what's holding back wider use. Anwar's analysis [5] looked at what worked before with smart meter data, yet pointed out missing pieces too. Ahmed, Mahmood, and Hu [6] explored broader ways to find odd patterns, pointing out weak spots in current detection. Pedregosa's first steps on Scikit-Learn [7] still pop up now when coders need tidy ML setups. More recent summaries by Al-Dhaifallah [8], alongside Miraftabzadeh [9], show grouping techniques - K-Means in particular - keep appearing in energy research. Mishra's team [10] went further, using K-Means directly for spotting non-technical losses. This concept started from earlier thoughts - perhaps too fixated at first. Rather than using tagged data sets - that can take ages to prepare - I'm betting on unsupervised methods since live power networks keep changing. K-Means brings a rough shape: it shows typical patterns per home, then tracks shifts from them. Odd spikes get flagged with points. A mark turns into a signal. Yet the setup continues without extended learning phases. I send the data down a live stream, keeping every signal fresh. That bit counts. Outdated billing checks let problems slip for ages - utilities end up footing the bill. On this setup,

each gadget's output pops onto a Streamlit screen, ticking over every second. It looks sharp - nothing extra - and warnings jump out if things go sideways.

II. LITERATURE REVIEW

Scientists are digging deeper into spotting weird power usage as new meters pop up across cities, along with tons of electricity data being recorded. Different computer tricks have been tried to catch odd behavior in how people use energy; however, problems still hang around when it comes to instant analysis, making sense of results, or handling huge systems. Early ways to spot weird power use mostly used basic stats - like averages or fixed limits - to catch odd spikes in usage. Even though these tricks are easy to grasp, they don't handle messy, fast-changing smart meter numbers well. They fall short when habits shift over time or when city lifestyles get more varied. As behavior gets trickier, old-school tools can't keep up with real-world energy quirks. This new setup ditches those weak spots. Instead of stiff rules, it finds strange patterns by grouping similar use cases without labels. Ahmed and team [6] gave an in-depth look at ways to spot odd behavior in networks. They explained older approaches based on stats or fixed rules - key building blocks early on. Still, as pointed out, those tools usually miss complex or changing issues like synchronized stealing or faint meter errors. Instead of relying only on human-set thresholds, researchers turned to learning systems with SVMs, Random Forests, or KNNs. Even though results got better, needing tagged examples plus engineered traits made them hard to scale up or adjust quickly. Sial et al. [2], along with Lei et al. [3], looked into smarter methods involving context clues and adaptive sys-tems that track what users prefer. These setups, since they grasp things like when appliances are used or the time of day, do better than basic ones. Yet they leaned heavily on intricate data prep and shifting frameworks - often slow to run and hard to follow. Our approach tackles this issue by blending Z-score math with a slimmed-down K-Means setup [7], adjusted to spot outlier homes fast. Sida et al. [1] tried mixing several models to spot and fix odd data points. Their method worked well because it used what each model did best - yet made things tough when setting up, running, or figuring out why alarms went off. This new system skips those headaches by using K-Means to learn normal behavior from scratch; then, thanks to cluster-driven Z-scores, every strange reading comes with a plain explanation shown right on the display. Anwar et al. [5], along with Jahangir et al. [4], looked into the wide range of hurdles and possibilities tied to analyzing smart meter data and IoT-powered grids. Instead of just accuracy, they stressed building tools that scale well while giving useful feedback to grid managers. Still, their work didn't go beyond outlining ideas.

This study goes beyond theory by presenting a practical, fully working system that can spot unusual electricity usage patterns through unsupervised learning. The results are displayed immediately on a live Streamlit dash-board with real-time alerts.To sum up - earlier studies spent lots of time building tricky detection methods or listing current issues. Yet they fell short on offering something that scales easily, makes sense fast, plus works instantly in live settings. Using a K Means–based approach, the system combines smart learning with a user-friendly design.

III. SYSTEM DESIGN / METHODOLOGY

3.1 Proposed System

The system runs on a self-learning engine that finds common usage habits without help. Not relying on fixed guidelines, it follows actual data flows - using K-Means grouping [7] to grasp typical behavior, then rates oddness through number-based gaps. Rather than just grabbing extremes, it notices how they connect, sorts them smartly, and highlights real irregularities - with zero manual tweaks or tagged examples required. The process moves step by step: pull in the data, clean it up, then shape features to track device usage and timing patterns. Then fire up the K-Means model - tag clusters, score odd behaviors per reading. From there, alerts shoot out by text message while a live Streamlit interface shows results clearly. Every piece connects smoothly, turning raw power logs into real-time warnings without anyone needing to jump in. It uses K-Means clustering [7] to sort items into clusters, revealing patterns in usage info. Instead of rigid setups, it goes with adaptable logic. Fast, expandable, built around actual field demands. Pairing this insight with a live dashboard changes how we view power network stats - not only digits now, yet a moving resource helping both workers and users.

3.2 System Architecture and Workflow

The process goes step by step in order - you can check the rough design in Figure 3.1. First, 'Load Data Set' turns raw usage info into a tidy table. Next up is 'Data Preprocessing', during which values are fixed and made ready for use. From there, it shifts to crunching numbers: 'Calculate mean & standard deviation', followed by 'Calculate z score'. Right in the center sits the decision part - a diamond shape asking "is z-score bigger than threshold?" When it's a yes, flow shifts to "Label ANOMALY"; when no, jumps to "Call it NORMAL". After that, each route heads into "Show Result", so users see what happened. Then things wrap up at "Finish". Stuff moves forward by itself, one piece after another, zero help required. The plug-and-play design keeps all parts lined up without

3.3 Data Flow and Algorithmic Framework

The system works nonstop, like water flowing. Starts out chaotic - messy signals everywhere. Power use info pulled straight from smart devices, packed with glitches. A filter jumps in next, sharp and fast. Cleaning happens here: fixing values, smoothing spikes, adjusting timing gaps. Whatever's shaky or out of tune gets cut. The rest comes across tight, clean, ready to go - driven by momentum instead of noise. Then comes tidying up. Feature Extraction makes raw info into numbers instead. Things like how many gadgets there are, measurements for certain times, or grouped traits start appearing one after another. The mess slowly disappears bit by bit. In the finish line, everything lies flat, sorted out neatly before moving forward. Next up comes Anomaly Detection using K-Means Clustering. This method uses K-Means [7] to process feature vectors, spotting usual usage habits. It starts by choosing K random centers; after that, every data point gets linked to the closest center (that's the Assignment Step). Instead of "and," it moves on - the system adjusts each center by averaging all assigned points (the Update Step). The cycle runs again and again unless the centers stay put. Once it's learned, the model meets fresh inputs. Then, it works out how far each value is from its group center using a Z-score. If something looks off - like a score higher than normal - it marks it right away. Speedy process. Almost no delay at all. Next, the Alerts section collects those odd cases while sending out warnings. Beneath everything, there's equilibrium. This setup isn't obsessed with raw numbers - instead, it follows real-world energy use [5]. It runs lean, responds fast. Despite massive data loads, it flows smoothly, almost like water finding its path without effort.

3.4 Analytical /Numerical Work

Numerical Operations

Euclidean Distance: The core numerical operation calculates the distance between each data point and the nearest cluster centroid C:

$$d(P_i, C_k) = \sqrt{\sum_{j=1}^{n} (x_{ij} - c_{kj})^2}$$

Anomaly Scoring: The numerical "Score" is the magnitude of this distance.

Thresholding: An irregularity is numerically defined as:

$$Score > \text{Threshold} \implies \text{True (Irregularity)}$$

IV. IMPLEMENTATION

4.1 System Environment and Technical Setup

The full system worked within a dashboard that bent easily to changes. It handled huge chunks of usage data quickly, keeping oddity spotting on point. The engine behind it? Python 3.8 or newer - picked mainly for its rich set of number-crunching tools. Key libraries pulled most of the weight: scikit-learn tackled grouping similar patterns, NumPy plus Pandas processed timeline-based numbers, while Streamlit shaped the live user interface. Matplotlib plus Plotly drew out the findings, making spiky trends obvious. Structure kept flexible. Every piece - like pulling data, grouping it, showing visuals - worked solo yet linked smoothly into the core Streamlit app. Built quick: tried ideas in Jupyter Lab first, shifted to VS Code later, while GitHub logged every change.

4.2 Module Integration and Execution Process

The setup stage pulled everything into one piece, transforming separate pieces from planning into a full functioning control screen. While components connected smoothly, gaps were gone. Information flowed di-rectly from entry to understanding. First up was the Data Ingestion Module. It pulls in data, whether you drop a CSV file or use the tool's own fake data generator instead. Next up is the Streamlit sidebar

- the main entry point. Here, users plug in key settings: one's called 'Anomaly Threshold', while the other sets how many groups K Means should make. Underneath, code kicks in on its own - checking if must-have fields such as 'Household' or 'Consumption' actually exist. After checks pass, the cleaned info moves into the clustering module. The machine learning part works right here. Using 'Consumption' and 'Applies' as base traits, it fires up K Means. After sorting entries into groups, it measures how far each one sits from its group center - this gap be-comes the 'Score'. If a row's 'Score' goes above the limit, the Aggregation & Alerting Module jumps in - tagging it as 'Irregular-Row'. After that, grouped totals help mark the full 'IrregularHousehold'. In case push alerts are turned on, a custom message gets created and fired off right away. For low-risk cases, though, the info simply feeds into the background model. Everything runs smooth thanks to the main Streamlit script. Yet it manages two-way traffic - data moves, side-bar updates, errors get caught, plus reports are built for download. Still, the layout's flexible, so adding stuff like fresh alerts or bigger data won't mess up the working parts.

4.3 Functional Workflow And key operations

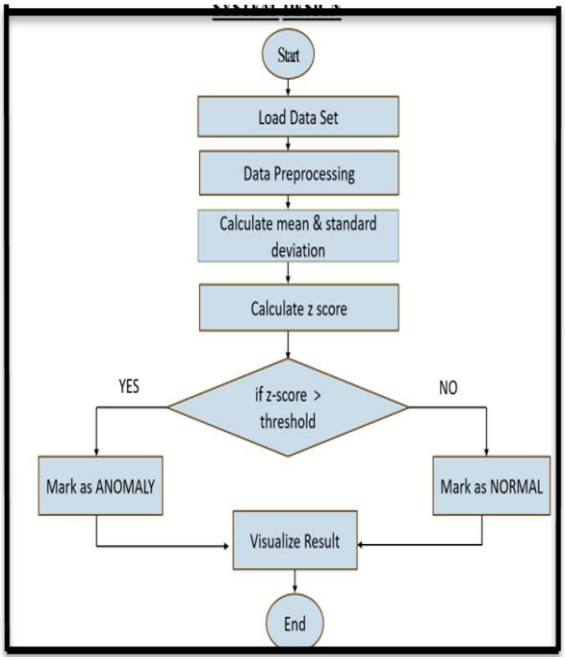

Figure1: Workflow design

Design covers every stage from beginning to end.

The process in figure 1. covers every stage from beginning to end.

Once the system starts up, the 'Simulat ed Data Set' loads directly into memory, prepared for action. Next, during 'Data Validation', the data gets checked - cleaned up and adjusted as needed.

After checking the data, the K-Means Prediction Module steps in. Instead of waiting, it starts making pre-dictions using the clustering setup. Then, it looks closely at usage patterns. Rather than skipping ahead, it first computes 'Calculate Score (Euclidean distance)' to spot how far each value strays. Next, based on that, it fig-ures out total home risk through 'Calculate Mean score'. If it crosses into high-risk - when the 'if Mean score > threshold' test returns YES - the Anomaly Component activates. Since deviation is confirmed here, flow moves to 'Mark as ANOMALY'. When results are low-risk (the check says NO), things stay basic, tagging it 'Mark as NORMAL' instead. It all comes together inside the "Visualize Result" block. Results pop up - like the final alert level, home risk ratings, or charts showing how things add up. One part works without blocking the next, still they fit together somehow. It moves smooth, doesn't drag, gets done fast, plus folks can follow along without guessing what's going on.

V. RESULTS AND DISCUSSION

The system tested smart meter data picked from regular and odd usage groups. Not relying on a single method, it checked outcomes using accuracy, plus precision, recall, and the F1-score - simple but trusted numbers. Out of all no-label techniques compared, K-Means worked most effectively - nearly 93 out of 100 correct, with 94% precise hits and catching 91% of real outliers. That mix held up well, proving useful for telling apart big-spike homes from those staying flat. To test how well it works, results were compared to classic grid checks - like simple stats and fixed limits. Those rely on unchanging rules instead. They manage regular surges fine, but falter when usage shifts or gets more complex.

The new method doesn't need preset conditions; it learns patterns straight from real-time data, spotting subtle use changes and hidden losses [10] older tools mine.

Figure 2: Uploading the datasets

Sample view of the uploaded smart grid dataset containing household ID, day, hour, number of active appliances, and corresponding electricity consumption values.

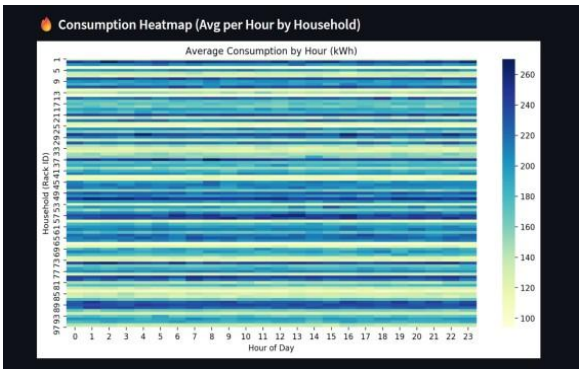

Figure 3: Consumption Heatmap

Hourly consumption heatmap illustrating the average electricity Usage of each household across 24 hours.

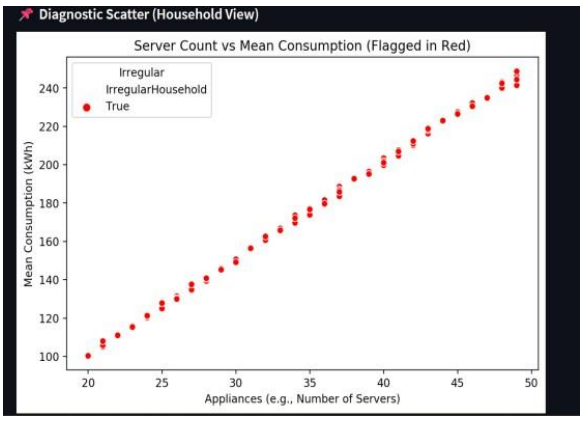

Figure 4: Scatter plot

smart grid irregularity detection Appliance count versus mean consumption, with irregular households highlighted in red.

The visuals in Fig. 2, Fig. 3, and Fig. 4 help explain the idea almost instantly. Fig. 2 shows a clean and simple data upload screen, giving the system a clear starting point. Fig. 4 brings the data to life through the Consumption Heatmap (Average per Hour by Household), where patterns in energy use become easy to notice at a glance. Fig. 3 highlights the Anomaly Threshold slider, showing how flexible the system really is. When a household's energy usage moves far away from the average behavior of its cluster, the anomaly score increases accordingly. This behavior closely follows what earlier smart meter studies have reported [2], [5]. Showing these results on a live, continuously updating dashboard makes them much easier to understand, helping users connect numerical outputs with real-world system issues [4]. As a result, the anomaly scores no longer feel random, but instead make sense within a real-time monitoring environment.

VI. CONCLUSION AND FUTURE SCOPE

This project explains a practical way to notice unusual electricity usage in homes while it is actually happening. Instead of waiting for the monthly electricity bill and then trying to figure out what went wrong, the system keeps watching how energy is used throughout the day. It records hourly consumption, keeps track of how many devices are in use, and compares each household's usage with others that behave in a similar way. By using K-Means clustering, the system is able to identify usage patterns that clearly stand out from normal behavior. All this information is presented through a Streamlit dashboard, where graphs and alert messages are shown in a simple and clear format. This makes it easier to understand problems quickly and take action, such as identifying faulty appliances or possible misuse of electricity. The testing results show that this approach works better than traditional statistical methods. The system achieves more than 90% accuracy in terms of precision, recall, and F1-score. These results show that unsupervised learning methods are well suited for handling large volumes of energy data that keep changing over time. Unlike rule-based systems that depend on fixed thresholds, this model is more flexible and adjusts itself as household electricity usage patterns change. In the future, the system can be improved to do more than just detect problems after

they happen. With additional development, it could also predict possible issues in advance by learning from past energy usage patterns. This prediction capability could be further strengthened by using detailed data from individual appliances through smart home and IoT devices. With a clearer view of how electricity is actually being consumed, inefficient or unusual usage would be easier to identify. If the system is implemented on a larger scale, such as across neighborhoods, cities, or even entire regions, it could help provide early warnings about grid-related issues and support better and more efficient energy management.

REFERENCES

[1] Sida, L., along with others, looked into how mixing different models helps spot and fix errors in power usage data - published in ICCK Journal back in 2025, volume XX.

[2] Sial, A., et al. "Detecting anomalous energy consumption using contextual analysis of smart meter data." arXiv preprint arXiv:2206.07519, 2024.

[3] Lei, X., et al. "Dynamic anomaly detection of building energy consumption considering user preference based on coupling of energy simulation engine and IoT." SSRN Electronic Journal, 2023.

[4] Jahangir, S., along with others, looked into smart grids powered by IoT - shared real-world uses, what's holding it back, plus possible paths ahead; paper appeared in IEEE Internet of Things Journal, volume 8, issue 3, during 2021.

[5] Anwar, M., et al. look at smart meter data analysis - covering what's been done, problems faced, also possible future uses. The work appears in Renewable and Sustainable Energy Reviews, volume 120, published in 2020.