**🎤 Slide 1: Introduction & Objective**

👤 *Speaker 1*

"Good [morning/afternoon] everyone. We’re a team of five working on optimizing load prediction for smart grids using historical and environmental data. Our **main objective** is to build a model that minimizes prediction error of electricity load in real-time using autoregressive and exogenous inputs.

Smart grids are a critical part of energy infrastructure. Predicting load accurately helps in optimizing power generation, reducing waste, balancing supply-demand, and even cutting operational costs. That’s why this optimization problem is not only mathematical, but environmentally and economically relevant."

**🎤 Slide 2: Motivation & Relevance**

👤 *Speaker 2*

"The motivation behind this project lies in the rising global energy demands and the need for smarter, data-driven systems to manage power flow. Poor load forecasting can lead to under-utilized energy or blackouts. In contrast, **fine-tuned load prediction** allows energy providers to prepare the grid for upcoming demand more accurately.

Our work focuses on minimizing the one-step-ahead prediction error – a practical goal that simulates real-time forecasting, which is what actual smart grids require."

**🎤 Slide 3: Objective Function & Variables**

👤 *Speaker 3*

"Mathematically, our objective is to minimize the following function:

J(φ,η)=∑t=1T(Xt−X^t(φ,η))2J(\varphi, \eta) = \sum\_{t=1}^T \left(X\_t - \hat{X}\_t(\varphi, \eta)\right)^2J(φ,η)=t=1∑T​(Xt​−X^t​(φ,η))2

Here:

* XtX\_tXt​ is the actual load at time *t*,
* X^t\hat{X}\_tX^t​ is the predicted load,
* φ\varphiφ are autoregressive coefficients (using past load),
* η\etaη are weights assigned to past environmental and operational features.

The predicted load is computed as:

X^t=∑i=1pφiXt−i+∑j=1b∑m=1Mηm,jum,t−j\hat{X}\_t = \sum\_{i=1}^p \varphi\_i X\_{t-i} + \sum\_{j=1}^b \sum\_{m=1}^M \eta\_{m,j} u\_{m,t-j}X^t​=i=1∑p​φi​Xt−i​+j=1∑b​m=1∑M​ηm,j​um,t−j​

So essentially, we use *p* past load values and *b* lags of *M* exogenous variables like voltage, current, solar/wind input, and electricity price."

**🎤 Slide 4: Methodology & Tools Used**

👤 *Speaker 4*

"To solve this, we approached it as a nested optimization problem. Here's a breakdown:

1. **Data Preprocessing**: We extracted lagged features from the dataset for both autoregressive and exogenous inputs.
2. **Model Construction**: For each combination of ppp and bbb, we built a linear model of the form Aθ=yA\theta = yAθ=y, where θ=[φ;η]\theta = [\varphi; \eta]θ=[φ;η].
3. **Solver**: We used a **closed-form least squares solution** and also implemented a **GPU-accelerated version** using CuPy for efficiency.
4. **Grid Search**: We ran an outer loop to search for the best ppp and bbb between 50 and 300 in steps of 50.
5. **Evaluation**: We split the data 80/20 and used test set SSE as our evaluation metric.

**Tools used**: Python, NumPy, CuPy for GPU optimization, Matplotlib for visualization, and Statsmodels for baseline comparison."

**🎤 Slide 5: Data & Results**

👤 *Speaker 5*

"We used the **Smart Grid Real-Time Load Monitoring Dataset** from Kaggle. It includes variables like power, temperature, humidity, and solar/wind data collected in 15-minute intervals.

After GPU-accelerated grid search, our best model achieved the lowest test SSE at:

* p=3p = 3p=3, b=1b = 1b=1 (with b=2 very close),
* The lowest test SSE was under **89,500**,
* The heatmap visualization clearly shows the performance trends across different lag combinations.

This result validates that **short-term past values** of both load and environmental conditions are most useful for real-time prediction."

**🎤 Closing**

👥 *All Together*

"Thank you for your time. We’re happy to take questions or discuss how this approach can be extended to non-linear models or real-world deployments."