Replication Instructions

Environment Setup

1. **Install required software**:
   - Python 3.8 or higher
   - All dependencies listed in `requirements.pdf`
2. **Clone the repository or unzip the provided code package**:

Dataset Preparation

1. **Download the datasets**
2. **Place datasets in the correct structure**:

```
datasets/
├── batlik/
│   ├── dataset1.csv
│   └── dataset2.csv
├── dconvert/
│   ├── dataset1.csv
...
```

3. **Verify dataset format**:
   - Each CSV file should contain configuration options as columns
   - The last column must be the performance metric

Running the Experiment

1. **Execute the main script**:

```
python deep.py
```

   Alternatively, run all cells in the `deep.ipynb` jupyter notebook

2. **For full experiment replication**:
   - Ensure all nine systems are included in the `systems` list
   - Set `num_repeats = 10` for statistical significance
   - Use `train_frac = 0.7` for the training/testing split
   - Maintain the default neural network parameters

Statistical Analysis

To replicate the statistical significance analysis:

1. **Examine the output files**:
   - `results/data/all_results.csv`: Contains average metrics
   - `results/data/all_results_detailed.csv`: Contains per-repeat metrics
2. **Verify Wilcoxon signed-rank test results**:
   - The p-values are reported in the `*_p_value` columns in the results CSV
   - $p < 0.05$ indicates a statistically significant difference

Expected Results

The deep learning approach should show:

1. **Performance improvements**:
   - Average MAPE improvement: ~48% over linear regression
   - Average MAE improvement: ~39% over linear regression
   - Average RMSE improvement: ~30% over linear regression
   - Average R2 improvement: ~13% over linear regression
2. **System-specific variations**:
   - Better improvements on systems with more complex configurations
   - Some systems may show minimal improvements
3. **Statistical significance**:

- Significant improvements ($p < 0.05$) in ~80% of the datasets

Visualizing Results

To create visualization of the results, run the `visualize.ipynb` jupyter notebook