AI/ML FOR 5G-ENERGY CONSUMPTION MODELLING BY ITU AI/ML IN 5G CHALLENGE

yisak bule

wachemo university

October, 25, 2023

Abstract *5G networks offer significant advantages over previous generations of cellular networks, but they also consume more energy. Accurate modeling of energy consumption is essential for optimizing energy efficiency in 5G networks. This report describes a machine learning-based model for predicting energy consumption of 5G products. The model was developed using a dataset of cell-level, base station and energy consumption data, and it achieved good performance on a holdout dataset. The model can be used to predict energy consumption of different types of 5G products. It can be used by network operators to optimize energy efficiency in their 5G networks, such as by identifying base stations that are consuming more energy than expected or evaluating the impact of different energy-saving measures.*

Keywords – 5G, energy consumption, machine learning, prediction, optimization

1. INTRODUCTION

5G networks are becoming increasingly important for a variety of applications, such as mobile broadband, fixed wireless access, and the Internet of Things. However, 5G networks also consume more energy than previous generations of cellular networks. This is due to a number of factors, including the use of higher frequencies, the deployment of more base stations, and the increased demand for data services [1].

Modeling energy consumption of 5G products is a challenging task. It depends on a variety of factors, such as the type of product, the configuration of the product, and the operating conditions. For example, a base station with more users will consume more energy than a base station with fewer users. Additionally, a base station operating in a dense urban area will consume more energy than a base station operating in a rural area [1].

Accurate modeling of energy consumption is essential for optimizing energy efficiency in 5G networks. By understanding how different factors affect energy consumption, network operators can make informed decisions about how to configure and deploy their networks [1].

This report describes a machine learning-based model for predicting the energy consumption of 5G products. The model was developed using a dataset of cell-level traffic statistics, base station data, and energy consumption data. The model was trained and evaluated on a holdout dataset, and it achieved good performance.

2. PROPOSED SOLUTION

2.1 Data cleaning and preprocessing

.Step 1: Merging the cell_level and base_station DataFrames on the BS and CellName columns allows us to combine the information about each base station with the information about each cell. This will be useful for creating features for predicting energy consumption.

Step 2: Calculating the mean of the missing values in each column of the merged DataFrame gives us an idea of how much data is missing from each column. This information can be used to decide how to handle missing values when training the machine learning model.

Step 3: Pivoting the merged DataFrame on the Time and BS columns, with the CellName column as the columns and the base_cols, esaving_cols, and load columns as the values, allows us to create a DataFrame where each row represents a base station at a specific time, and each column represents a different feature. This format is ideal for training a machine learning model.

Step 4: Resetting the index of the pivoted DataFrame ensures that each row has a unique index. This is necessary for training the machine learning model.

Step 5: Joining the pivoted DataFrame with a DataFrame containing the unique Mode and RUType values for each base station, merged on the BS column, allows us to add these features to the DataFrame. These features may be useful for predicting energy consumption since they capture information about the type of base station and the type of load that is connected to it.

Step 6: Applying the date_features() function to the train, test, and pv DataFrames creates new features that capture the time information in the DataFrame. These features may be useful for predicting energy consumption, since energy consumption often varies over time.

Step 7: Merging the pv DataFrame with the train DataFrame on the Time and BS columns allows us to add the pv features to the train DataFrame. These features may be useful for predicting energy consumption, since they capture information about the amount of solar energy that is being generated.

Step 8: Creating a new column in the merged DataFrame called split, which contains the value test if the Energy column is missing, and the value train otherwise, allows us to split the DataFrame into two DataFrames: one for training the machine learning model and one for evaluating the model.

Step 9: Splitting the merged DataFrame into two DataFrames: train (containing all rows where the split column is equal to train) and test (containing all rows where the split column is equal to test) allows us to train and evaluate the machine learning model without overfitting.

2.2 Feature engineering

2.2.1 Adding and counting features

The Python code you provided defines a function called count_feat(), which takes a Pandas DataFrame and a training DataFrame as input and returns a DataFrame with new features extracted from the two DataFrames. The new features are based on the counts of different variables in the training DataFrame, grouped by the BS (base station) variable.

The fe_en() function is a comprehensive feature engineering function for energy consumption prediction of 5G base stations. It uses a variety of techniques to extract features from the data, including: summing features, counting features, counting unique values, normalizing features by energy saving, clustering features, and performing principal component analysis (PCA). The function also groups the data by base station and computes the maximum value for each feature. This is useful for identifying base stations with unusual patterns. The function returns two DataFrames: one for the training data and one for the test data. This allows you to train your machine learning model on the training data and evaluate its performance on the test data.

In this step, I was computed the sum of base station features in row wise and count the unique value and how many value are appeard (Nona Na value). And I also clustered them and used PCA. Another one techniques was grouping by BS and computing the max value then merge them on BS, this technique was computed for count columns and sum columns.

2.2.2 Moving statics and lag features

The lags_features() function is a comprehensive feature engineering function for energy consumption prediction of 5G base stations using lagged features. It uses a variety of techniques to extract features from the data, including computing differences between consecutive values, computing ratios between consecutive values, computing rolling mean, standard deviation, maximum, minimum, quantiles, and forecast values of features, and shifting features forward and backward to create lagged features.

The function takes two DataFrames as input: one for the training data and one for the test data. It returns two DataFrames as output: one with the training data features and one with the test data features. The function also returns a list of the lag values used to create the lagged features.

The technique of taking lag features is a powerful way to extract information from data for time series prediction tasks. In the context of energy consumption prediction of 5G base stations, lag features can be used to capture the temporal patterns of energy consumption and load.

The lag features are well-chosen and likely to be useful for energy consumption prediction. By taking lag values of the load, energy, and load and energy differences, you are capturing the short-term and long-term trends in these variables. The moving average, max, min, std, and quantile features are also useful for capturing the overall characteristics of the data and any outliers.

The use of a statistical model to predict load values in the next hour is another good idea. This feature can be used to capture the impact of future load on energy consumption. Overall, this feature engineering techniques are likely to be effective for energy consumption prediction of 5G base stations.

2.2.3 Feature engineering in BS inside

The bs_corelation() function is a comprehensive feature engineering function for calculating base station-specific features that can be used to predict energy consumption. It takes four DataFrames as input: the main DataFrame containing all of the data, the training DataFrame, the validation DataFrame, and the test DataFrame.

The function first groups the data by base station and then calculates a variety of features for each base station, including:

- ❖ Cosine similarity between the load time series and the energy time series
- ❖ Coefficients of the load, hour, and count features in a linear regression model predicting energy consumption
- ❖ Mean absolute error of the linear regression model predicting energy consumption
- ❖ Correlation coefficient between the first-order differences of the load and energy time series

The function then merges these features back into the main DataFrame. Finally, the bs_corelation() function trains a linear regression model and a random forest regressor model to predict energy consumption for each base station. It also calculates the mean absolute error of each model on the cross-validation set.

2.2.4 Target and mean encoding

In this funcation, we use of target and mean encoding for energy consumption prediction. We use a variety of techniques to encode categorical and temporal features, including:

- ❖ Encoding the BS and BS_hour features using the mean of the Energy target variable (target encoding)
- ❖ Encoding the load feature using the mean of the entire dataset (mean encoding)
- ❖ Encoding the diff1_eng and diff1_load_t+1 features (first-order differences of the Energy and load features) using target encoding
- ❖ Encoding the label_eng and label_load features (binary variables indicating whether the energy consumption or load increased) using target encoding

It was showed that using these encoding techniques can significantly improved the performance of machine learning models for energy consumption prediction.

2.2.5 Probability of increasing with load and clustering

The get_prob_static() function takes two DataFrames as input: the main DataFrame containing all of the data and the training DataFrame. It returns a single DataFrame containing the following features:

- ❖ imporve_probability: The probability of energy consumption increasing in the next hour, based on the base station.
- ❖ std_diff: The standard deviation of the difference in energy consumption between the current hour and the previous hour, based on the base station.
- ❖ mean_diff: The mean of the difference in energy consumption between the current hour and the previous hour, based on the base station.
- ❖ diff1_hour_mean: The mean of the difference in energy consumption between the current hour and the previous hour, based on the base station and the hour class.
- ❖ diff1_hour_std: The standard deviation of the difference in energy consumption between the current hour and the previous hour, based on the base station and the hour class.
- ❖ imporve_probability_hour: The probability of energy consumption increasing in the next hour, based on the base station and the hour class.
- ❖ imporve_laod_eng_probability: The probability of energy consumption and load increasing in the next hour, based on the base station.
- ❖ imporve_load_eng_probability_hour: The probability of energy consumption and load increasing in the next hour, based on the base station and the hour class.

The function first calculates the following features for each base station in the training DataFrame:

- label_diff_eng: A binary variable indicating whether the energy consumption increased in the next hour.
- label_diff_load: A binary variable indicating whether the load increased in the next hour.
- diff_eng_load: A binary variable indicating whether both the energy consumption and the load increased in the next hour.

The function then merges these features with the main DataFrame.

The get_cluster() function takes two DataFrames as input: the main DataFrame containing all of the data and the training DataFrame. It returns a single DataFrame containing the following features:

- cluster: The cluster number assigned to the base station.
- cluster_without_ru: The cluster number assigned to the base station, without considering the RUType feature.

The function first trains a KMeans clustering model on the base station features in the training DataFrame. The model is trained with 10 clusters. The function then assigns each base station in the training DataFrame and the main DataFrame to a cluster. Finally, the function returns a DataFrame containing the cluster numbers for each base station.

The get_prob_static() and get_cluster() functions can be used to create features for predicting energy consumption. The get_prob_static() function creates features that capture the historical energy consumption patterns of each base station. The get_cluster() function creates features that capture the similarities between base stations.

2.2.6 One hot encoding the quantile value

The get_quantile() function takes a DataFrame, a column name, a suffix, and an optional training DataFrame as input. It calculates the 5th, 15th, 25th, 35th, 45th, 60th, 75th, 85th, and 95th percentiles of the column in the training DataFrame (or the input DataFrame if no training DataFrame is provided). It then creates a new column in the input DataFrame with the suffix _quantile and assigns each row to a quantile bin based on the value of the column in the original column.

The one_hot_encode() function takes a DataFrame, a list of column names, and a threshold as input. It creates new columns in the DataFrame for each unique value in each of the column names, and then assigns each row to 1 for the column corresponding to the value of the original column in the input DataFrame. If the value of the original column is not present in the unique values for the column name, then the value of the new column is set to 0.

The code provided then uses these two functions to create new features in the df DataFrame. First, it uses the get_quantile() function to create a new column called load_quantile that contains the quantile bin for the sum_load column. Then, it uses the one_hot_encode() function to create new columns for each unique value in the load_quantile column.

Next, it uses the get_quantile() function to create a new column called energy_bsc_quantile that contains the quantile bin for the Energy column. Finally, it uses the one_hot_encode() function to create new columns for each unique value in the energy_bsc_quantile column.

3. Results

3.1 model selection

The LightGBM (LGBM) model is a good choice for predicting energy consumption because it is a fast and efficient model that can handle large and complex datasets. LGBM is also a gradient boosting model, which means that it can learn complex relationships between the features and the target variable.

3.2 Model score

models	mae	mape	info
model_1	1.185	0.042	if BS present in train data
model_2	2.57	0.092	if BS not present in train data
model_3	5.35	0.24	if BS and RUType not present in train data
model_4	5.7	0.21	if BS and RUType not present in train data amd antennas more than 32
overall model		0.074	all data

3.3 Feature importance

model 1

diff1_eng_BS_hour_enc_	1329
pred_rf_bsc	1315
Energy_BS_hour_enc_	1314
diff1_load_t+1_BS_hour_enc_	1109
trend_forcast	897
dayofweekhour	861
corr_bsc	852
load_t+1_BS_hour_enc_	851
daydayofweekhour	840
diff1_hour_std	824
load_quantile_6.0	801
coef_load_bsc	792
label_eng_BS_hour_enc_	771
mae_cv_rf_bsc	768
coef_hour_bsc	755
trend_lower_forcast	748
all_sum_cols	737
label_load_BS_hour_enc_	715
imporve_probability	714
load_quantile_7.0	709

model 2

load_Cell0_q2_BS_all_rows	487
BS_all_sum_cols_bsc	483
load_Cell0_min_BS_all_rows	450
load_Cell0_std_BS_all_rows	401
load_Cell0_skew_BS_all_rows	336
load_Cell0_q3_BS_all_rows	323
hour	311
all_sum_cols	306
load_Cell0_median_BS_all_rows	301
load_Cell0_q1_BS_all_rows	297
load_Cell0_range_BS_all_rows	287
BS_sum_load_bsc	284
load_Cell0_max_BS_all_rows	250
load_Cell0_size_BS_all_rows	242
load_Cell0_q8_BS_all_rows	232
load_Cell0_mean_BS_all_rows	229
BS_sum_Esaving_bsc	215
load_Cell0_q6_BS_all_rows	206
sum_load	203
load_Cell0_q7_BS_all_rows	188

model 3

BS_sum_Esaving_bsc	3216
BS_sum_load_bsc	3166
BS_all_sum_cols_bsc	2534
BS_n_load	2475
hour	1544
BS_sum_TXpower_bsc	573
hour//2	508
TXpower_Cell0	413
daydayofweekhour	332
load_Cell0_max_RUType_all_rows	328
diff2_load	302
hour_class	293
hour//6	275
load_Cell0_mean_RUType_all_rows	274
sum_load+sum_Antennas+count_load_aut	269
diff1_load	264
sum_load+sum_TXpower+count_load_aut	244
sum_load*sum_Antennas*sum_Frequency_aut	235
sum_load-sum_Antennas-sum_Frequency_aut	226

sum_load+sum_Antennas_aut	220

model 4

load_Cell0_q2_BS_all_rows	1339
load_Cell0_size_BS_all_rows	1203
BS_all_sum_cols_bsc	1157
load_Cell0_min_BS_all_rows	1066
load_Cell0_skew_BS_all_rows	1057
load_Cell0_std_BS_all_rows	1028
BS_sum_load_bsc	907
BS_sum_Esaving_bsc	880
hour	849
load_Cell0_q1_BS_all_rows	800
load_Cell0_max_BS_all_rows	757
load_Cell0_q8_BS_all_rows	739
load_Cell0_q7_BS_all_rows	675
load_Cell0_median_BS_all_rows	662
load_Cell0_mean_BS_all_rows	630
BS_n_load	577
BS_sum_TXpower_bsc	407
yhat_upper_forcast	390
additive_terms_forcast	362
hour//2	339

CONCLUSION

The model was developed using a dataset of cell-level traffic statistics, base station and energy consumption data. The model was trained and evaluated on a holdout dataset, and it achieved good performance. The model can be used to predict energy consumption of different types of 5G products, such as base stations, small cells, and user equipment. The model can be used by network operators to optimize energy efficiency in their 5G networks. For example, the model can be used to identify base stations that are consuming more energy than expected. The model can also be used to evaluate the impact of different energy-saving measures, such as turning off base stations during periods of low traffic.

ACKNOWLEDGEMENT

Thank you, ITU team for organizing the competition and zindi team for hosting this great competition .

REFERENCES

[1]https://challenge.aiforgood.itu.int/match/matchitem/83

https://zindi.africa/competitions/aiml-for-5g-energy-consumption-modelling