	Data generation	Model (finetuned on PF)	Post-processing
Function	generate data with random n-1 and n-2 contingencies A	solve power flow	* compute branch current * check for branch current violations (overloading * check for bus voltage violations
Inputs	* aggregated load profiles * config file * case file	* PV at PV nodes * PQ at PQ nodes * Vθ at slack * scenario index	* branch params 1 * bus params 2 * PF solutions 3 * scenario index 4
Outputs	*for each scenario: - node features PQVθ - adjacency list - index of lines/transformers that are dropped B * branch parameters (admittance and capacity) 1 C * bus parameters (voltage bounds and base voltage) 2	3 * PF solutions 4 * scenario index (needed to map scenarios to table of dropped branches) B	* bus voltage violations * branch current violations

Remarks

Ⓐ we don't generate all n-1 (or n-2) contingencies since that would be too large (ieee300 has ~400 branches, so we would have to generate 400 contingencies per load scenario = 400 x 10000
= $4 \cdot 10^6$ scenarios.

Instead, for each load scenario, we generate 20 contingency scenario. For each contingency scenario, we select one ($p=0.5$) or two branches at random and drop them

Ⓑ we store the idx of the branches that we drop in each scenario to know which are the branches for which we dont need to compute the current during post processing.

we store the scenario idx with each scenario so we can map the scenario to the right idx of dropped branches

Ⓒ we need some additional branch and bus parameters to compute the branch currents and check for violations hence why we had to add these outputs to the data gen pipeline