AIR FORCE OPERATIONAL ENERGY

SWETHA RAJKUMAR
JENNIFER LIANG
ANNIE PANG

Berkeley

ABSTRACT

We want to develop an improved energy logistics planning tool by training a machine learning model to optimize risk.

The tool's purpose is to "operate" a logistics network, taking as input data the structure and properties of a supply network and outputting a plan that supplies the demand of locations in the network ('nodes'), subject to constraints (e.g., the capacity of pipelines). These nodes may represent bases, ports, oil fields, etc., while the connections between them ('edges') may represent pipelines, railroads, shipping lanes, or other means of transporting fuel.

This tool is meant to improve a manual process that has historically been driven by the time/monetary cost of distributing fuel, rather than the perceived risk of executing certain plans (in terms of lost assets). As the Air Force looks to transition to a risk-based framework, this tool has the long-term goal of providing both a simulation and planning capability, for both strategic planning and real-life execution.

We advanced the tool's efficiency and modularity for inclusion in larger systems with the integration of AI/ML techniques. There are still key capability gaps that must be overcome before it can see regular use. In particular, failure states are likely in the current architecture; this is an artifact of both our current understanding of how to model the problem, as well as the particulars of the software framework we utilize.

BACKGROUND

The data generation process involved using a Monte Carlo approach to generate candidate plans and then scoring those plans with a simulation function.

- Data generation steps:
  - Supply data (fuel energy data) from an excel spreadsheet which contains node/edge relationships and corresponding risks
  - Topological sorting of nodes in the Directed Acyclic Graph (DAG) to ensure that we are traversing the graph from top to bottom
    - Necessary to meet fuel conservation constraints
  - Propagating random flow amounts down the graph to satisfy demand. Each generated flow object tracks the amount of fuel that was propagated down each edge
  - Applying the simulation function to score each flow plan based on one random draw of possible risks (the score represents the total amount of demand that was satisfied by the plan for that given draw)
- Our final dataset contained roughly 72,000 flow objects with corresponding simulation scores

MODEL

We used a multiple regression model to train our data. For every flow object, the flow along each edge is an input variable and the score for the entire flow plan is the output variable. We chose to use a regression model because it works for best with the data points we have (various numerical features that correspond to a numerical output).

We discovered from exploratory data analysis that the input data for our problem has a very high dimensionality (as seen in Figure 6, our supply network contains 49 nodes and 119 edges, and more extensive networks would have even bigger dimensionalities). Figure 1 also demonstrates that the flow along edges have very unclear relationships—both with other edge flows and with the outputted score. Therefore, this type of data is an excellent candidate for deep learning. For our tool, we used Google's TensorFlow package to build a multiple regression model. The formal architecture of our neural network can be seen in Figure 2.

Our model enables rapid assignment of scores to candidate flow objects, and its assessment of the score for a flow object would ideally include the knowledge of all possible draws (risk impacts) that could be simulated, as seen in the pipeline illustrated in Figure 7. In operation, the tool would iteratively generate candidate flow plans until a sufficient score is reached. The main steps are outlined below, and can also be visualized in the histogram in Figure 8:

- Generate (random) candidate flow plan
- Use TensorFlow model to check its predicted score
- Repeat until predicted score exceeds some threshold

CONCLUSION

This work improves the generation of candidate fuel logistics plans in wargaming/modeling/simulation, which improves fuel planning techniques both in strategic plans and in actual operations.

By applying machine learning techniques to a fuel logistics network, we are able to optimize against risk without complex techniques poorly suited for the range of outcomes and decision criteria in the military. This technique incorporates uncertainty and can be used in everything from table-top exercises to large-scale simulations.

We are using deep learning techniques as a shortcut in assessing the effectiveness of a plan, but future work could train models to generate plans directly given the configuration of a network and its risks.

Future work could also automate the process of assessing adding edges, mitigating risks, and defending an area more strongly against attacks with the goal to improve our network.

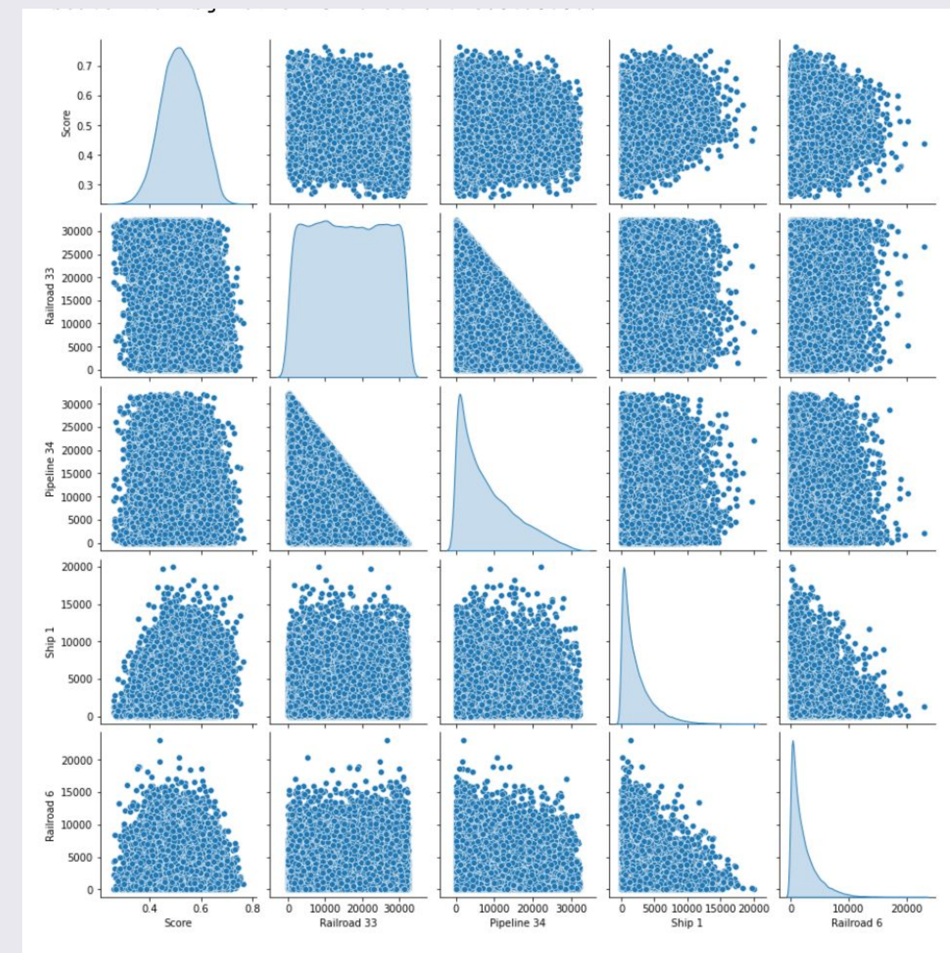

Figure 1: Seaborn graph that shows relationships between selected edge flows and between those edge flows and the plan's score

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 normalization (Normalizatio  (None, 119)              239
 n)

 dense (Dense)               (None, 128)               15360

 dense_1 (Dense)             (None, 128)               16512

 dense_2 (Dense)             (None, 64)                8256

 dense_3 (Dense)             (None, 1)                 65

=================================================================
Total params: 40,432
Trainable params: 40,193
Non-trainable params: 239
_________________________________________________________________
```

Figure 2: TensorFlow Model Summary

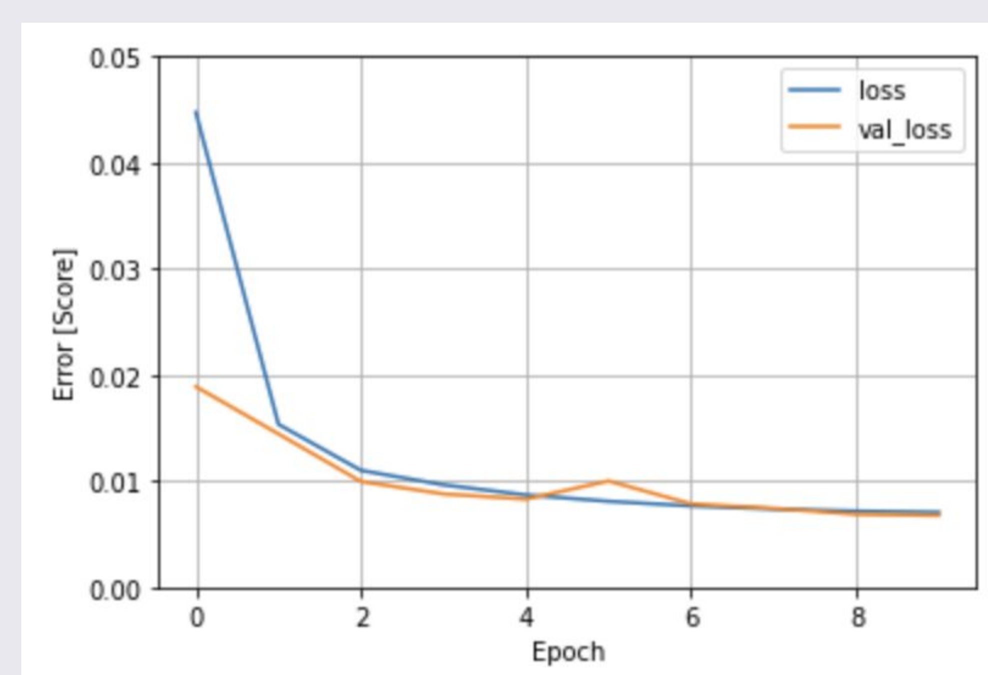

Figure 3: Training and validation losses measured with the Mean Absolute Error (MAE) metric

RESULTS

Figures 3, 4, and 5 depict visualizations from our training and testing process. From the results, we can see that our model had no overfitting or underfitting, meaning that multiple regression was a simple, yet effective solution to our problem.

VISUALIZATION

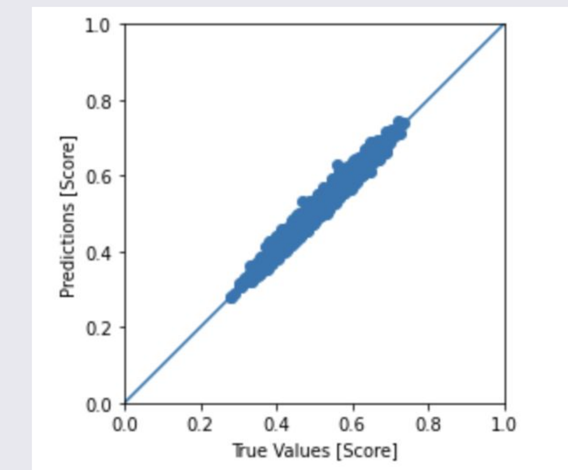

Figure 4: A plot that shows how closely the predictions on the testing set match up with the true scores

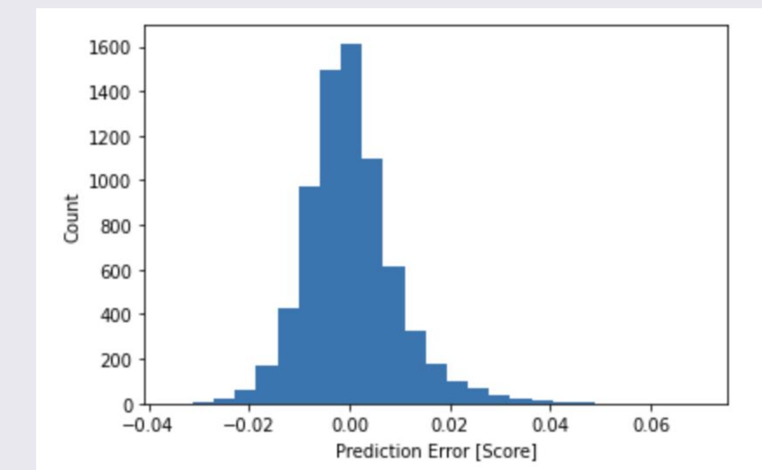

Figure 5: Error distribution for the predictions on the testing set

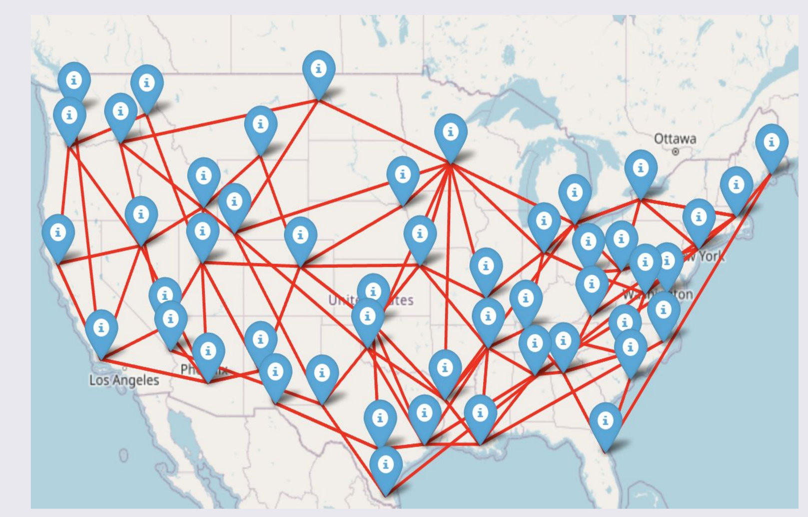

Figure 6: Visual of supply network (49 nodes and 119 edges)

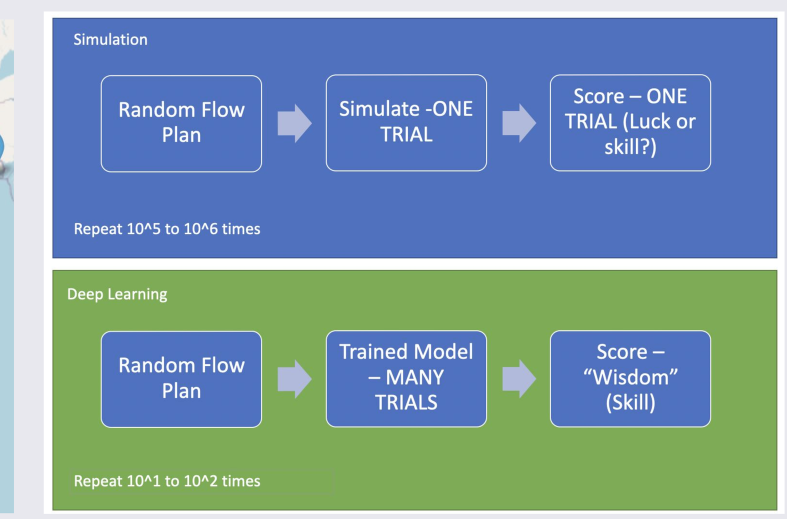

Figure 7: Comparison of simple simulation vs. deep learning

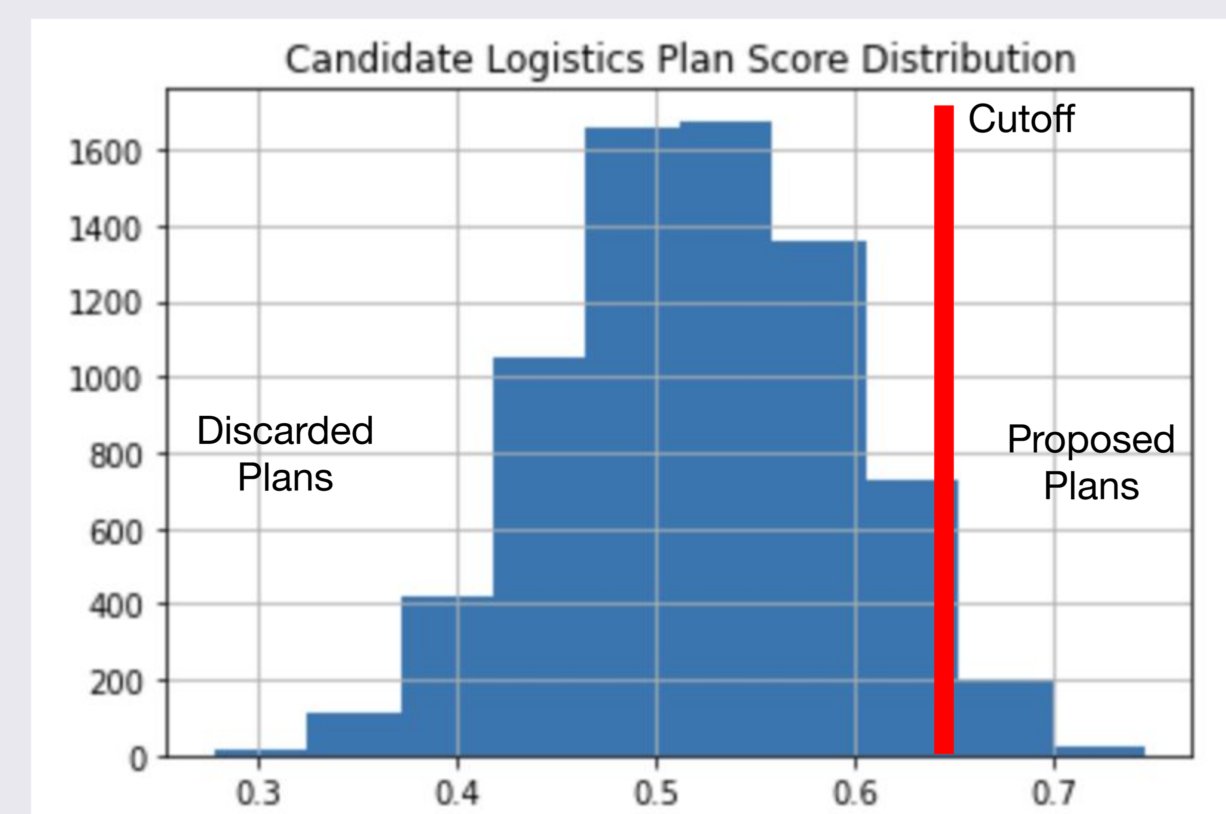

Figure 8: Histogram of candidate logistics plan scores (cutoff for an "acceptable" plan is set at the 95th percentile)

FUTURE DIRECTIONS

To further investigate, we will improve the network by adding edges to lower the impact of risk, integrating different types of supply, and re-working the node/edge relationships to ensure that we can achieve higher scores for flow plans (currently, the maximum score corresponding to a flow object in our dataset was roughly 76%). We will also be incorporating node throughput and storage constraints in the generation of randomized flow plans.

To increase the usability of the network, we will also add a more formalized UI to help us clearly visualize relationships between nodes and edges.