

The NIST logo consists of the letters "NIST" in a bold, white, sans-serif font. The "N" and "I" are connected, and the "S" is a simple, blocky shape.

US PET's Lab

Making Privacy Technologies Accessible Throughout
Government

9.25.24

Curtis Mitchell, Mike Walton (xD/Census)

Gary Howarth (NIST)



AGENDA

- Project Mission and Goals
- Phase 1 Walkthrough
- Phase 2 Progress
- Future Directions



OVERVIEW

Project Mission & Goals



To produce a proof-of-concept research environment that allows users to explore documentation, use cases, and “hands on” examples of PETs deployed in a sandbox environment.



US PETs LAB PHASE 1

Alpha Build of the US PETs Lab



Goal – Demonstrate Functionality

Objective – The solution provides a simple web application inspired by the Privacy Engineering Collaboration Space.

Detail – Implement a user-facing web application with areas describing PETs tools, use cases and interactive code examples. Demonstrate a particular joint use case using a single PET (e.g., federated learning).



CONCEPT

Two Parts:

Web Application – Documentation and Links to Resources

Coding Environment – Place to experiment with PET software
and processes



APPLICATION

Two Parts:

Web Application – Jekyll static site application using USWDS

Coding Environment – Python-based Jupyter Notebooks with pre-loaded PETs libraries



Documentation

Applications: Opportunities for privacy enhancement and use-cases

PETs: Technical and non-technical documentation of various PETs

Threats: Detailing threat models and best practices for selecting de-risking strategies using PETs

Datasets: Available and open data documentation



APPLICATION

Frontend Web Application

Welcome to the United States Privacy-Enhancing Technologies Laboratory

From fundamentals to advanced topics, this laboratory is intended to help developers and data scientists as well as legal and policy professionals experiment with and evaluate PETs for their applications

[Get Started](#)

Get in Touch

Do you have questions about PETs or the features in this laboratory? Do you have a case study that you'd like to highlight? Or feedback for how the PETs Lab might better meet your needs? We'd love to hear from you. Email us at inquiries@xd.gov or open pull requests and issues on [Github](#).

[Email Us](#)



APPLICATION

Sandbox Jupyter Notebooks

The screenshot shows a Jupyter Notebook window titled 'PPFL.ipynb'. The interface includes a top menu bar (File, Edit, View, Run, Kernel, Tabs, Settings, Help), a left sidebar with a file browser showing a 'dataset' folder and the current notebook 'PPFL.ipynb', and a main code editor area. The code is written in Python and is divided into two sections: 'Federated Training' and 'Define Flower Client'. The 'Federated Training' section contains two functions: `get_parameters` and `set_parameters`. The `get_parameters` function returns a list of numpy arrays for each parameter in the network's state dictionary. The `set_parameters` function takes a list of numpy arrays and updates the network's state dictionary. The 'Define Flower Client' section contains a `FlowerClient` class that inherits from `fl.client.NumPyClient`. The class has methods for `__init__`, `get_parameters`, `fit`, and `evaluate`. The `fit` method trains the model for one epoch and returns the parameters and accuracy. The `evaluate` method evaluates the model and returns the loss and accuracy.

```
File Edit View Run Kernel Tabs Settings Help
PPFL.ipynb
Python 3 (ipykernel)

Filter files by name
Name Last Modified
dataset 15 days ago
PPFL.ipynb 2 days ago

Federated Training

[11]: def get_parameters(net) -> List[np.ndarray]:
      return [val.cpu().numpy() for _, val in net.state_dict().items()]

      def set_parameters(net, parameters: List[np.ndarray]):
          params_dict = zip(net.state_dict().keys(), parameters)
          state_dict = OrderedDict((k: torch.Tensor(v) for k, v in params_dict))
          net.load_state_dict(state_dict, strict=True)

Define Flower Client

[12]: class FlowerClient(fl.client.NumPyClient):
      def __init__(self, net, trainloader, valloader):
          self.net = net
          self.trainloader = trainloader
          self.valloader = valloader

      def get_parameters(self, config):
          return get_parameters(self.net)

      def fit(self, parameters, config):
          set_parameters(self.net, parameters)
          train(self.net, self.trainloader, epochs=1)
          return get_parameters(self.net), len(self.trainloader), {}

      def evaluate(self, parameters, config):
          set_parameters(self.net, parameters)
          loss, accuracy = test(self.net, self.valloader)
```



US PETS LAB PHASE 2

Focusing on Privacy-Preserving Federated Learning



Ideation: Pivoting to Focus on Privacy-Preserving Federated Learning for Medical Genomic Data

Planning:

- Iterated project plan with NIST
- Presented early-stage ideas to US/UK Gov Health Groups



Version Control: Github-hosted repository

Deployment:

- Started with cloud.gov in phase 1, but free tier proved inadequate (memory limits)
- Pivoting to NIST-managed AWS Environment
- Project conducted through the National Cybersecurity Center of Excellence (NCCOE) to facilitate collaboration



Reference: Paper Using Soybean Genomic Data

Deciding factors:

- Notebooks hosted on Github
- Publicly available datasets
- Responsive authors

Gill et al. *BMC Plant Biology* (2022) 22:180
https://doi.org/10.1186/s12870-022-03559-z

BMC Plant Biology

RESEARCH Open Access

Machine learning models outperform deep learning models, provide interpretation and facilitate feature selection for soybean trait prediction

Mitchell Gill¹, Robyn Anderson¹, Haifei Hu¹, Mohammed Bennamoun², Jakob Peterleit¹, Babu Valliyodan^{3,4}, Henry T. Nguyen¹, Jacqueline Batley¹, Philipp E. Bayer¹ and David Edwards^{1*}

Abstract
Recent growth in crop genomic and trait data have opened opportunities for the application of novel approaches to accelerate crop improvement. Machine learning and deep learning are at the forefront of prediction-based data analysis. However, few approaches for genotype to phenotype prediction compare machine learning with deep learning and further interpret the models that support the predictions. This study uses genome wide molecular markers and traits across 1110 soybean individuals to develop accurate prediction models. For 13/14 sets of predictions, XGBoost or random forest outperformed deep learning models in prediction performance. Top ranked SNPs by F-score were identified from XGBoost, and with further investigation found overlap with significantly associated loci identified from GWAS and previous literature. Feature importance rankings were used to reduce marker input by up to 90%, and subsequent models maintained or improved their prediction performance. These findings support interpretable machine learning as an approach for genomic based prediction of traits in soybean and other crops.

Keywords: Machine learning, XGBoost, Interpretable models, Feature selection, Genomic selection, Soybean



DEVELOPMENT

Current Work:

- Recreating example notebooks for updated Python environment
- Creating federated instance of soybean feature prediction models using Flower FL framework





PPFL Evaluation Plans:

- User community feedback
 - SME input for PPFL
 - SME input for genetics use case
- Threat modeling
 - Developing Privacy Threat Modeling Toolkit with MITRE
 - Cybersecurity threat modeling
- Empirical privacy evaluation
 - Developing software tools to estimate privacy risks
- Privacy red teaming



DEVELOPMENT

Next Steps:

- Add privacy mechanisms, e.g., differential privacy, secure multiparty computation aggregation, to federated learning workflow
- Develop/adapt fidelity, utility, and privacy metrics



NEXT STEPS

Continuing Development of the US PETs Lab



NEXT STEPS

Near-Term Steps:

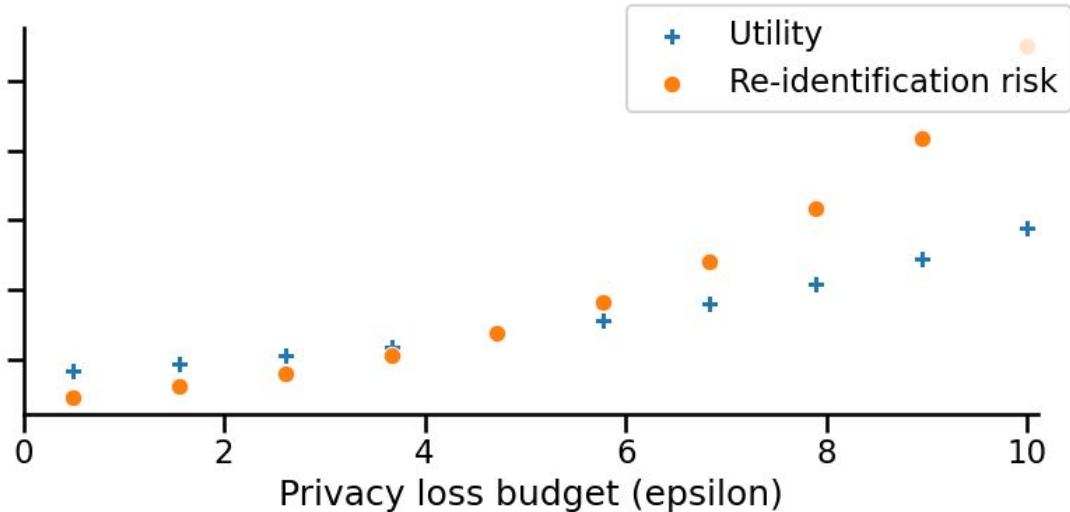
- Build out examples of different ML/DL models used with PPFL
- Study effect of privacy-preserving parameters on empirical privacy risks
- Compare performance of different PET combinations with federated learning
- Rerun models using alternate data



NEXT STEPS

Near-Term Steps:

- Study effect of privacy-preserving parameters on empirical privacy risks





NEXT STEPS

Medium-Term Steps:

- Resume documentation for other PETs
- Gather additional use cases and datasets
- Determine sandbox access paradigm



NEXT STEPS

PETs to Add in Future:

- Differential Privacy
- Zero-Knowledge Proofs
- Homomorphic Encryption
- Others?



Q & A



gary.howarth@nist.gov

curtis.l.mitchell@census.gov

michael.w.walton@census.gov

NIST



United States™
Census
Bureau