Deep Learning Acceleration of Progress Toward Delivery of Fusion Energy

William M. Tang

Princeton University/Princeton Plasma Physics Laboratory (PPPL)

THEORY & SIMULATION OF DISRUPTIONS WORKSHOP (TSDW-2017)

Princeton Plasma Physics Laboratory
Princeton, NJ

July 17-18, 2017

Co-authors: **Julian Kates-Harbeck, Alexey Svyatkovskiy, Kyle Felker, Eliot Feibush, Michael Churchill**

SITUATION ANALYSIS

Most critical problem for MFE: *avoid/mitigate large-scale major disruptions*

•Approach: Use of big-data-driven statistical/machine-learning (ML) predictions for the occurrence of disruptions in EUROFUSION facility "Joint European Torus (JET)"

•Current Status: ~ 8 years of R&D results (led by JET) using Support Vector Machine (SVM) ML on *zero-D* time trace data executed on CPU clusters yielding ~ *reported success rates in mid-80% range for JET 30 ms before disruptions , BUT > 95% with false alarm rate < 3% actually needed for ITER* (Reference – P. DeVries, et al. (2015))

•**Princeton Team Goals include:**

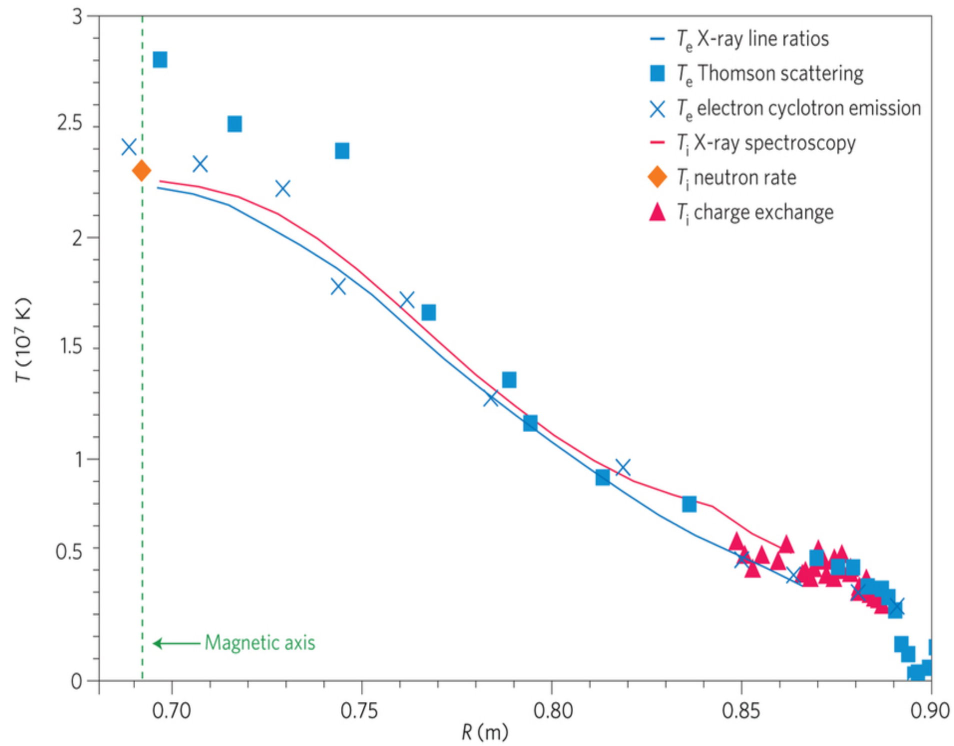
(i)improve physics fidelity via development of new *ML **multi-D**, time-dependent software including better classifiers;*

(ii)develop ***"portable"*** (cross-machine) predictive software beyond JET to other devices and eventually ITER; and

(iii)enhance execution speed of disruption analysis for very large datasets

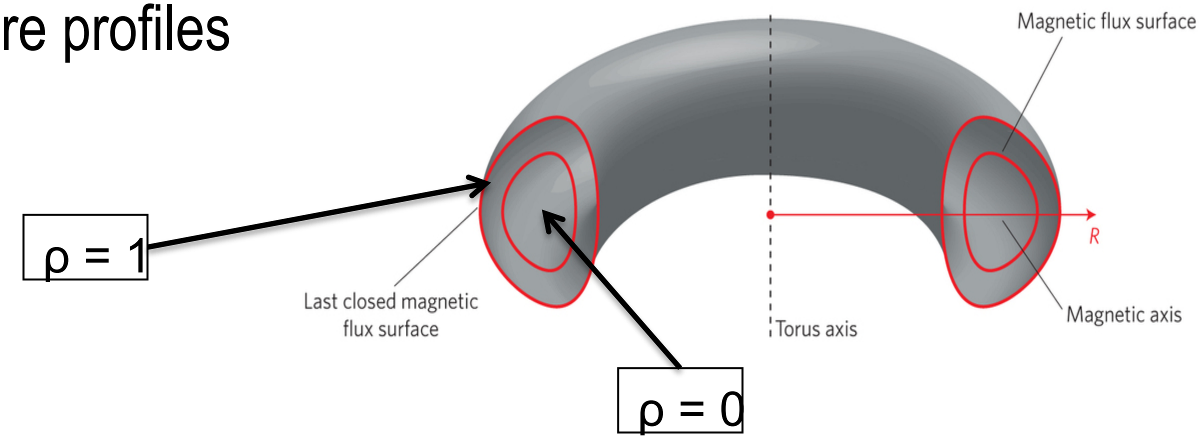
→ *development & deployment of advanced ML software via **Deep Learning Recurrent Neural Networks***

Challenges & Opportunities



Higher Dimensional Signals

- At each timestep: **arrays** instead of **scalars**
- All as a function of ***ρ*** (normalized flux surface)
- Examples:
  - 1D Current profiles
  - 1D Electron temperature profiles
  - 1D Radiation profiles



Mazon, Didier, Christel Fenzi, and Roland Sabot. "As hot as it gets." Nature Physics 12.1 (2016): 14-17.

Machine Learning Workflow

Identify Signals
• Classifiers

Preprocessing and feature extraction

Normalization

Train model, Hyper parameter tuning

Use model for prediction

Princeton/PPPL DL predictions now advancing to *multi-D time trace signals* (*beyond zero-D*)

Measured sequential data arranged in patches of equal length for training

All data placed on appropriate numerical scale ~ O(1)
e.g., Data-based with all signals divided by their standard deviation

• All available data analyzed;
• Train LSTM (Long Short Term Memory Network) iteratively;
• Evaluate using ROC (Receiver Operating Characteristics) and cross-validation loss for every epoch (equivalent of entire data set for each iteration)

Apply ML/DL software on new data

JET Disruption Data

# Shots	Disruptive	Nondisruptive	Totals
Carbon Wall	324	4029	4353
Beryllium Wall (ILW)	185	1036	1221
Totals	509	5065	5574

Sample 7 Signals of zero-D time traces (07)	Data Size (GB)
Plasma Current	1.8
Mode Lock Amplitude	1.8
Plasma Density	7.8
Radiated Power	30.0
Total Input Power	3.0
d/dt Stored Diamagnetic Energy	2.9
Plasma Internal Inductance	3.0

JET produces ~ Terabyte (TB) of data per day

~55 GB data collected from each JET shot

➔***Well over 350 TB total amount with multi-dimensional data yet to be analyzed***

Deep Recurrent Neural Networks (RNNs): Basic Description

- **“Deep”**
  - **Hierarchical representation** of complex data, building up salient features automatically
  - Obviating the need for hand tuning, feature engineering, and feature selection
- **“Recurrent”**
  - Natural notion of time and memory → i.e., at every time-step, the output depends on
    - Last Internal state “s(t-1)” ***Recurrence!***
    - Current input x(t)
  - The internal state can act as memory and accumulate information of what has happened in the past

Internal State (“memory/ context”)

$$s_t = A(W_{in}x_t + W_{recurrent}s_{t-1} + b)$$
$$h_t = W_{out}s_t$$

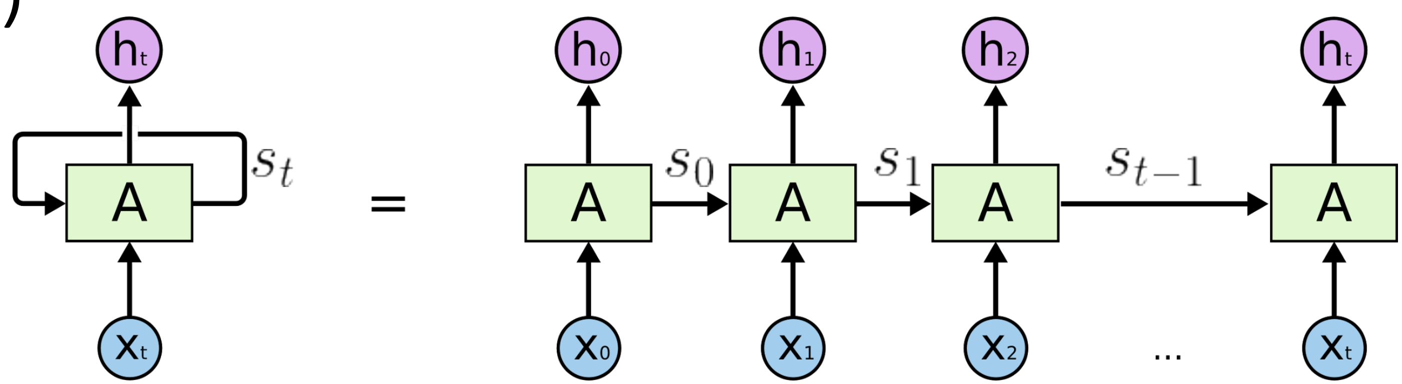
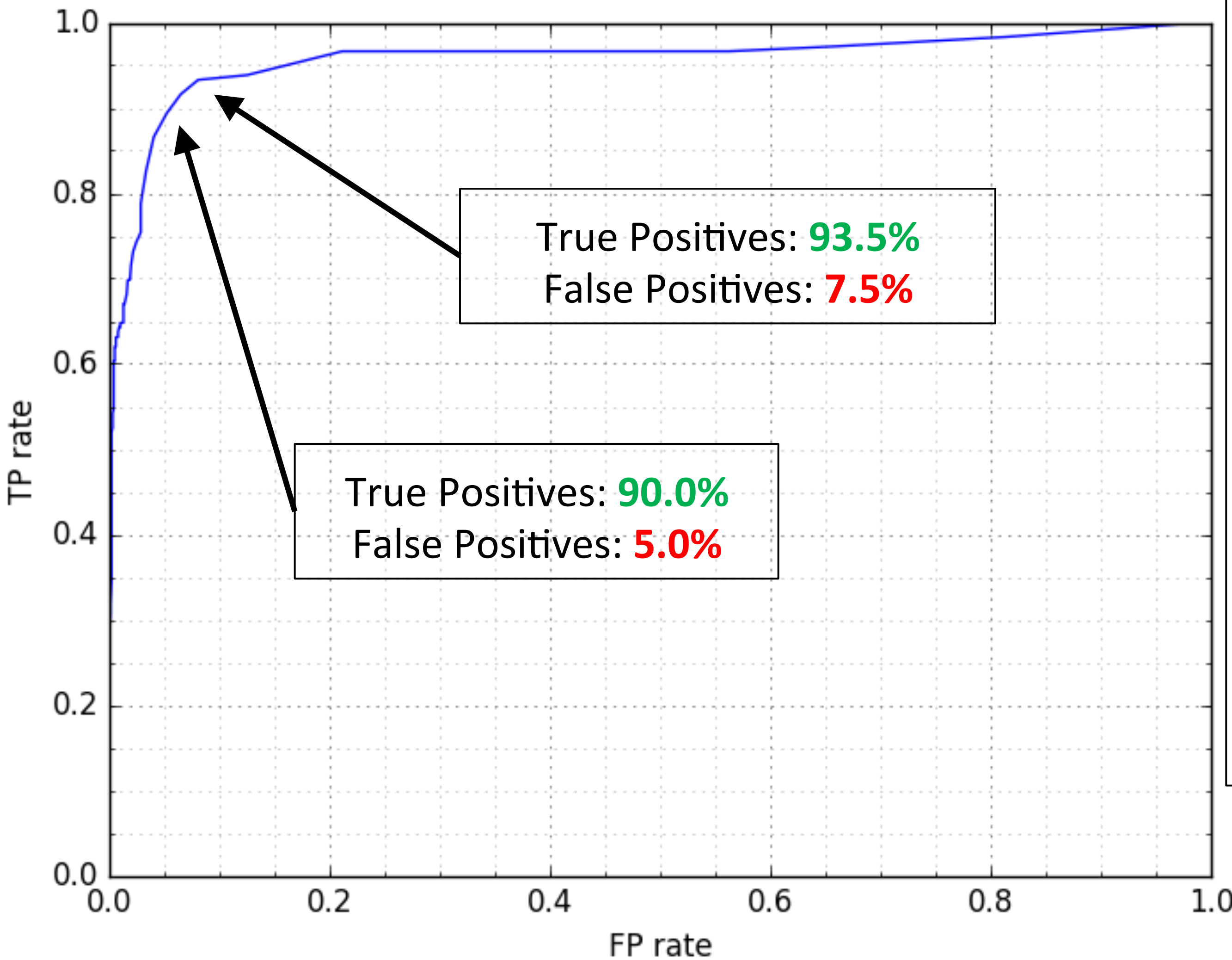


Image adapted from: colah.github.io

FRNN ("Fusion Recurrent Neural Net") Code Performance (ROC Plot)

Performance Tradeoff: Tune True Positives (good: correctly caught disruption) **vs. False Positives** (bad: safe shot incorrectly labeled disruptive).



RNN Data:

- Testing **1200 shots** from Jet ILW campaigns (C28-C30)
- **All shots used**, no signal filtering or removal of shots

Jet SVM* work:

- **990 shots** from same campaigns
- **Filtering** of signals, **ad hoc removal of shots** with abnormal signals
- **TP 80 to 90%, FP 5%**

*Vega, Jesús, et al. "Results of the JET real-time disruption predictor in the ITER-like wall campaigns." *Fusion Engineering and Design* 88.6 (2013): 1228-1231.

RNNs: HPC Innovations Engaged

GPU training

- Neural networks use dense tensor manipulations, efficient use of GPU FLOPS
- Over 10x speedup better than multicore node training (CPU's)

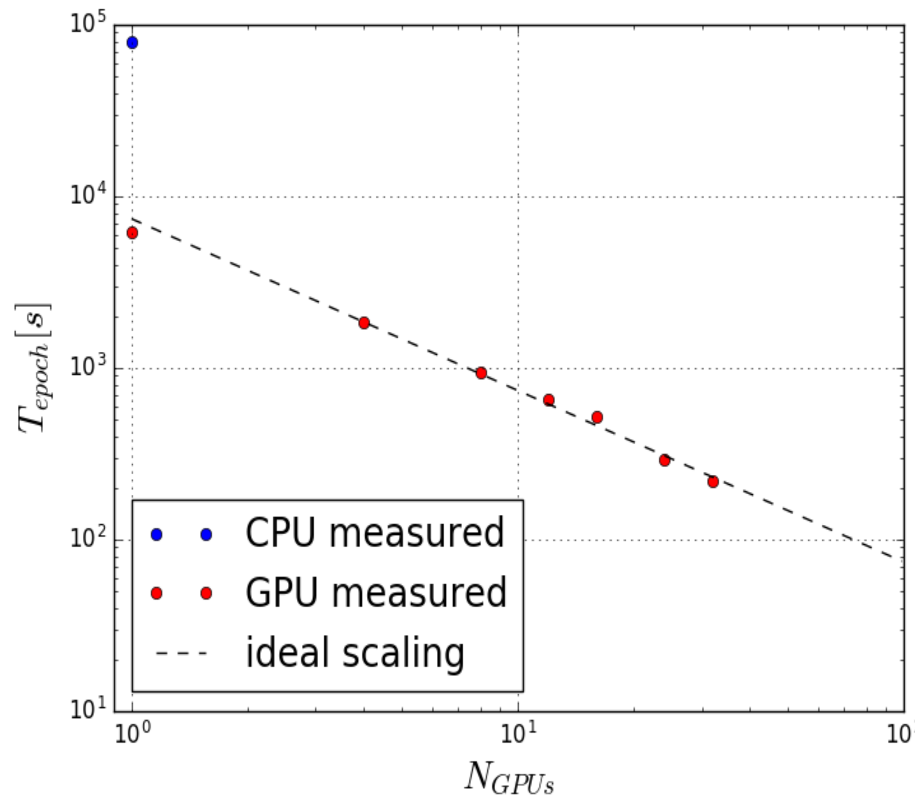
Distributed Training via MPI

Linear scaling:

- Key benchmark of "time to accuracy": we can train a **model that achieves the same results nearly N times faster with N GPUs**

Scalable

- **to 100s or >1000's of GPU's on Leadership Class Facilities**
- **TB's of data and more**
- Example: Best model training time on full dataset (~40GB, 4500 shots) of 0D signals training
  - **SVM** (JET) **: > 24hrs**
  - **RNN** (20 GPU's) : **~40min**



Fusion Recurrent Neural Net (FRNN) Description

- **Python deep learning code** for disruption prediction in fusion (tokamak) experiments
  - Reference: https://github.com/PPPLDeepLearning/plasma-python
- Implements distributed data parallel synchronous RNN training
  - *Tensorflow & Theano backends with MPI for communication*
  - FRNN code workflow is characteristic of typical distributed deep learning software
  - Core modules:
    - **Models:** Python classes necessary to construct, train, and optimize deep RNN models.
    - **Pre-process:** arrange data into patches for stateful training; normalize
    - **Primitives:** Python objects for key plasma physics abstractions
    - **Utils:** a set of auxiliary functions for pre-processing, performance evaluation, and learning curves analysis

Scaling Summary

Communication: each batch of data requires time for synchronization

$$T_{sync} \sim log\left(N_{workers}\right)$$

Runtime: computation time

$$T \sim \frac{1}{N}\left(A + B\ log(N)\right) = O\left(\frac{log(N)}{N}\right)$$

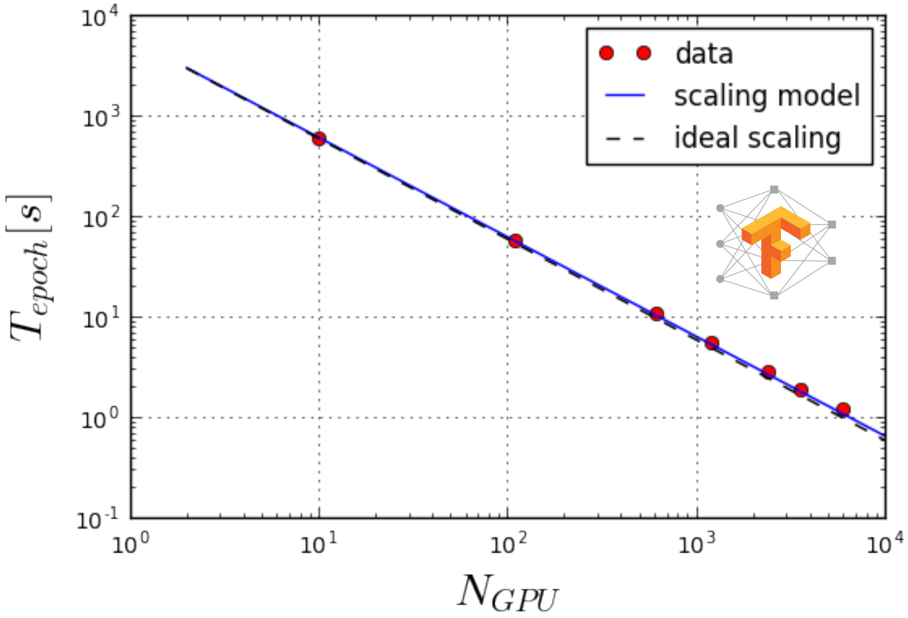
Parallel Efficiency

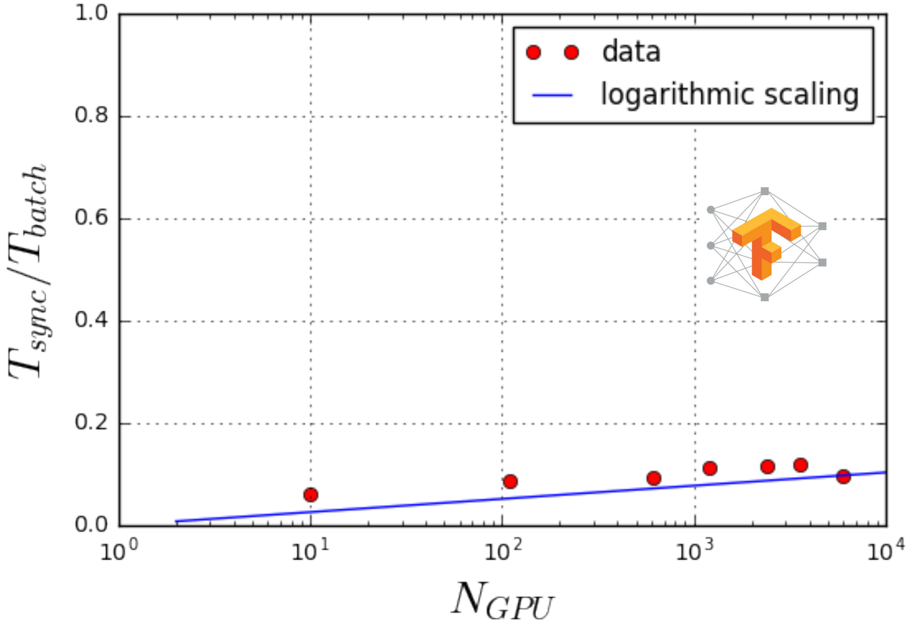
$$\text{Parallel Efficiency} \sim \frac{A + B}{A + B\ log(N)} = o\left(\frac{1}{log(N)}\right)$$

FRNN Scaling Results on GPU's

- Tests on OLCF Titan CRAY supercomputer
  - **OLCF DD AWARD**: *Enabled Scaling Studies on Titan currently up to 6000 GPU's*
  - Total ~ 18.7K Tesla K20X Kepler GPUs

Tensorflow+MPI





CURRENT PERSPECTIVE

Forecasting disruptions using machine learning is an important application of a **general idea:**

→ Use multi outcome prediction to **distinguish disruption types/scenarios**

→ Beginning now to move from **prediction** to **active control** (including new collaborations on DIII-D – R. Buttery, T. Strait, N. Logan, R. Nazikian, …..)

→ Increasingly large and diverse data sets require building *scalable systems to take advantage of leadership class computing facilities*

Fusion Deep Learning (FRNN) Technical Summary

• FRNN → a distributed data-parallel approach to train deep neural networks (stacked LSTM's);

• Replica of the model is kept on each "worker" → *processing different mini-batches of the training dataset in parallel;*

• Results on each worker are combined after each *epoch* using MPI;

• Model parameters are synchronized via parameter averaging → *with learning rate adjusted after each epoch to improve convergence*

• Stochastic gradient descent (SGD) used for large-scale optimization with parallelization via mini-batch training to reduce communication cost.

→ **Challenge**: scaling studies to examine if *convergence rate saturates/ decreases with increasing mini-batch size (to thousands of GPU's).*

→ **Targeted Large HPC Systems with P-100's for Performance Scaling Studies:** (1) "PIZ-DAINT" Cray XC50 @ CSCS (Switzerland) with > 4K GPU'S; (2) "SATURN V" @ NVIDIA with ~ 1K GPU's; (3) "TSUBAME 3" @ TITECH with ~ 3K GPU's; & (4) "SUMMIT-DEV" @ OLCF.

Fusion Big Data ML/DL Application Summary

• **Fusion Energy Mission**:

-- *Accelerate demonstration of the scientific & technical feasiblity of delivering Fusion Power*

-- *Most critical associated problem is to avoid/mitigate large-scale major disruptions.*

• **ML Relevance to HPC:**

-- ***Rapid Advances*** *on development of predictive methods via large-data-driven* ***"machine-learning" statistical methods***

-- ***Approach Focus:*** ***Deep Learning/Recurrent Neural Nets (RNNs)***

-- ***Significance:*** *Exciting alternative predictive approach to "hypothesis-driven/first principles" exascale predictive methods*

-- ***Complementarity:*** *Physics-centric path-to-exascale HPC needed to introduce/establish improved Supervised ML Classifiers with associated features*

• **Associated Challenge:**

→ *Improvements over zero-D SVM-based machine-learning needed to achieve > 95% success rate, <5% false positives at least 30 ms before disruptions -- with portability of software to ITER via enhanced physics fidelity (capturing multi-D) with improvement in execution time enabled by access to advanced HPC hardware (e.g., large GPU systems).*