Fully Convolutional Spatio-Temporal Models for Disruption Predictions

NSTX-U / Magnetic Fusion Science Meeting

September 21, 2020

Ge Dong (PPPL/PU)

Kyle Felker (ANL/PU), Alexey Svyatkovskiy (Microsoft Inc./PU), William Tang (PPPL/PU), Julian Kates-Harbeck (Kernel Inc./PPPL)

Outline

- Background and Introduction
- Model AI/Deep Learning Architecture
- Training and Prediction Results
- Computational Performance
- Ongoing & Future FRNN Development
- Summary

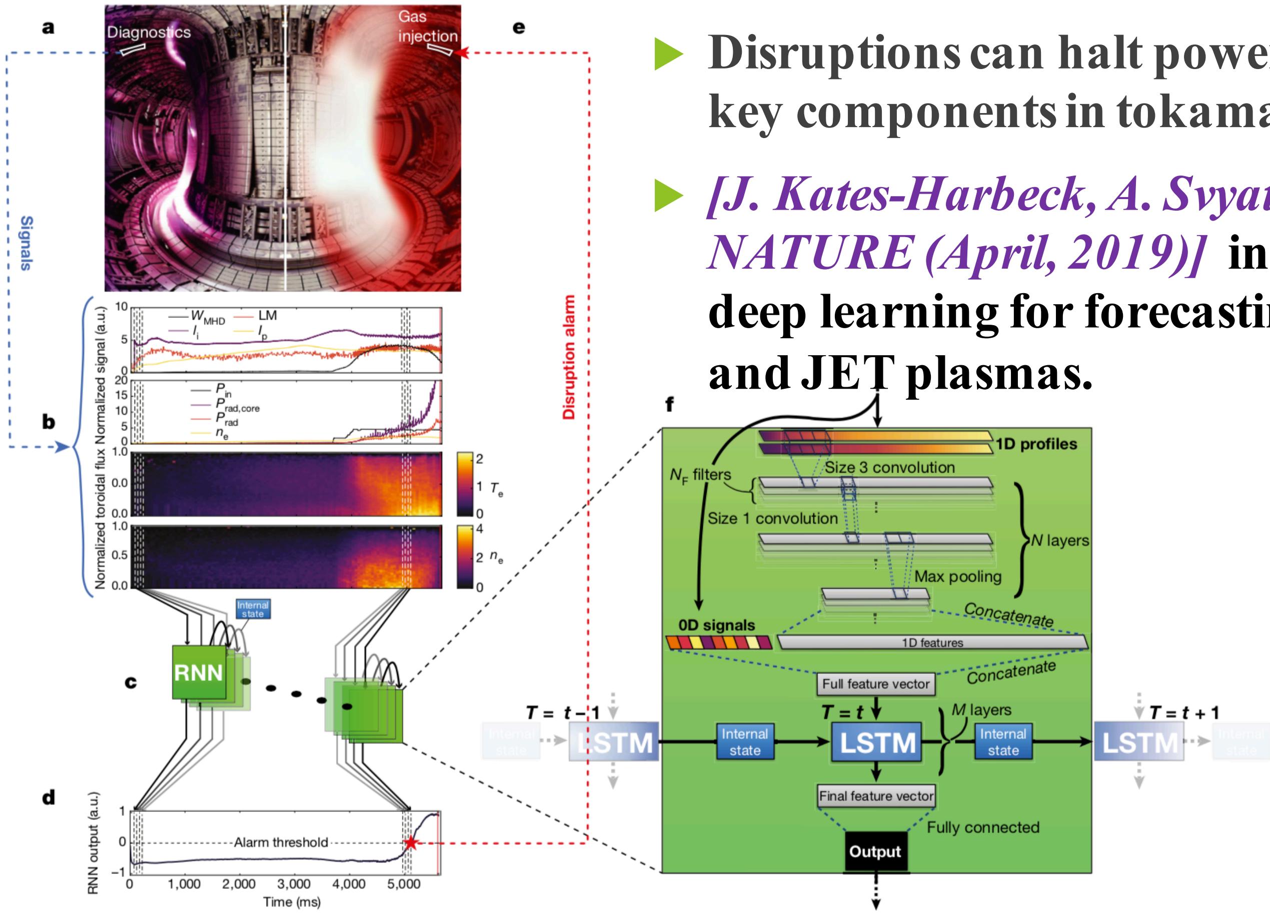
Introduction: Background

- **Deep Learning** – An increasingly important & prominent Artificial Intelligence (AI)-enabled methodology for the analysis and interpretation of challenging spatio-temporal data in modern scientific areas.

- **Present Context** -- Application of deep learning models for addressing disruption prediction & control in tokamaks is a high-profile exemplar problem.

Introduction: FRNN
[“Fusion Recurrent Neural Network” (*RNN*)] Software

- **Disruptions can halt power production and damage key components in tokamaks.**
- ***[J. Kates-Harbeck, A. Svyatkovskiy and W. Tang, NATURE (April, 2019)]*** **introduced method based on deep learning for forecasting disruptions in DIII-D and JET plasmas.**



Introduction: FRNN

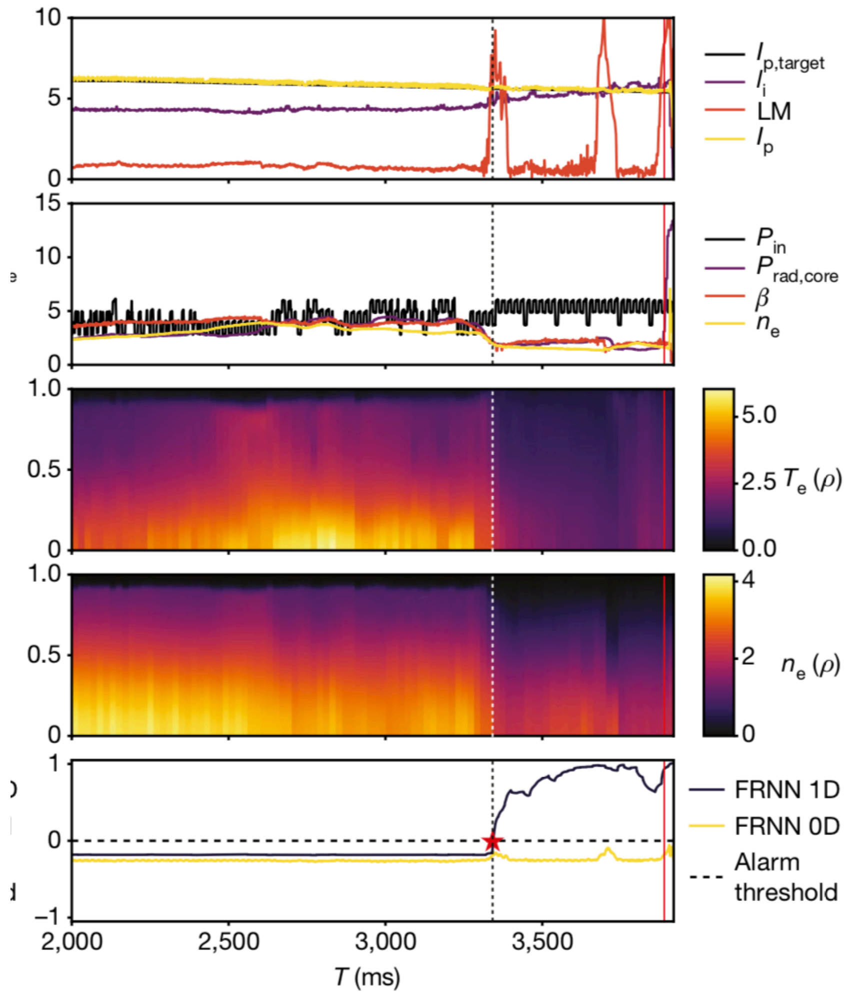
- Deep learning models <u>first to demonstrate efficient cross-machine predictive capabilities</u>

Table 1 | Prediction results

	Single machine		Cross-machine		Cross-machine with 'glimpse'
Training set	DIII-D	JET (CW)	JET (CW)	DIII-D	DIII-D $+ \delta$
Testing set	DIII-D	JET (ILW)	DIII-D	JET (ILW)	JET (ILW) $- \delta$
Best classical model	**0.937**	0.893	0.636	0.616	0.851
FRNN 0D	0.890	**0.952**	**0.761**	0.817	0.879
FRNN 1D	0.922	–	–	**0.836**	**0.911**

Introduction: FRNN

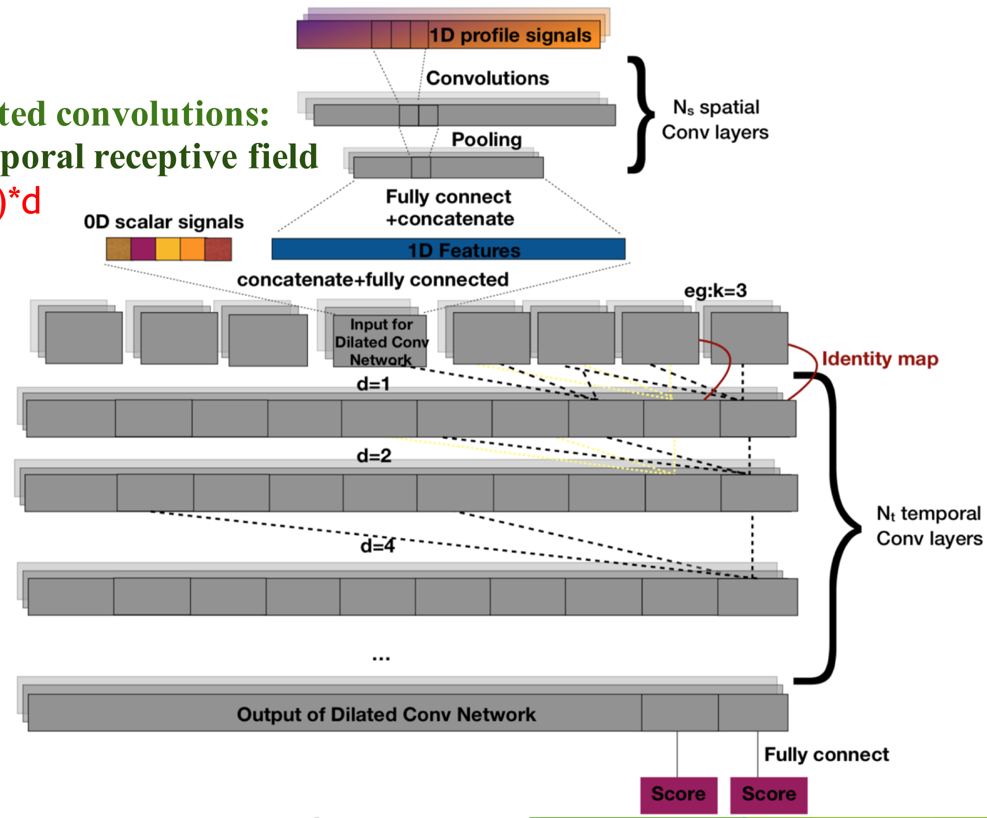
- D3-D Tokamak Shot #159,593: Illustration of unique capability of the deep-learning model to use 1D information (black) to correctly predict the oncoming disruption.



Model Architecture - Temporal Convolutional Neural Network (TCN)

- Schematic of TCN-based model
- **Advantages:**
  - TCN directly fetches historical information through a time series (Note: *RNN can still lose distant information through operations on the cell state that carries long term memory*)
  - Easy model parallelism — "feedforward" network does not incorporate gated functions or recurrent connections

Dilated convolutions:
Temporal receptive field
(k-1)*d

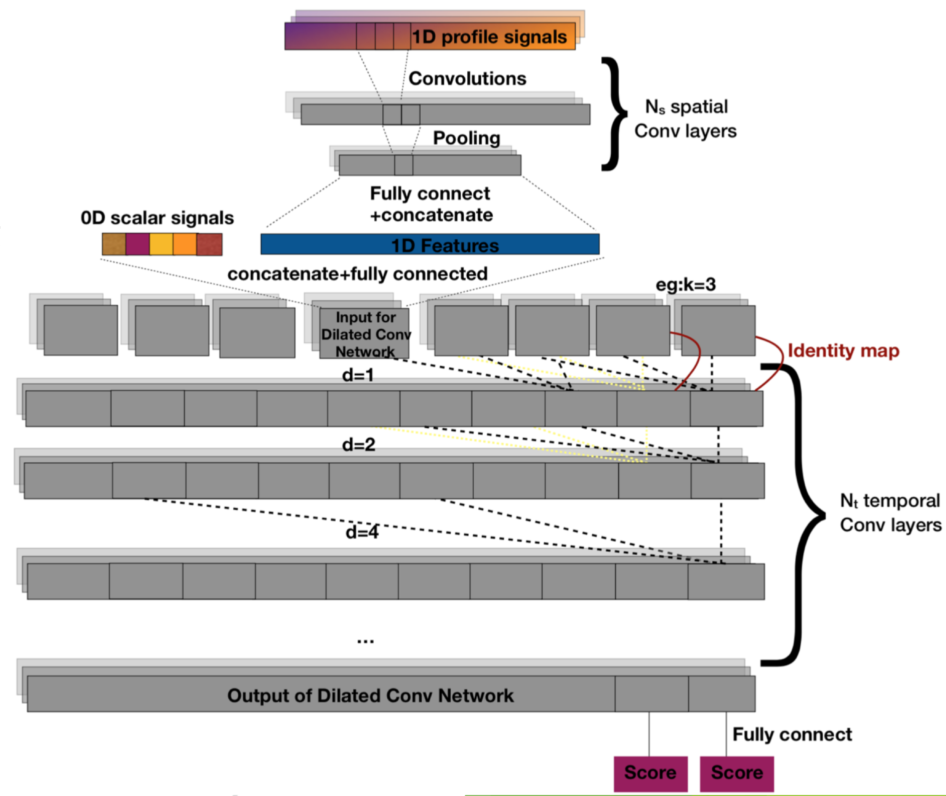


Model Architecture - Inputs

Signal description	Numerical scale DIII-D	Numerical scale JET
Plasma current	3.8 e-1 MA	5.03 e-1 MA
Plasma current target	3.9 e-1 MA	Not available on JET
Plasma current error	3.1 e-2 MA	Not available on JET
Plasma current direction	1.0	Not available on JET
Internal Inductance	2.02 e-1	1.51 e-1
Plasma density	1.19 e19 m^{-3}	4.69 e19 m^{-3}
Input power (beam for DIII-D)	1.85 e6 W	4.47 e6 W
Radiated power core	4.58 e2 W	4.05 e4 W
Radiated power edge	4.94 e2 W	2.72 e4 W
Stored energy	2.79 e5 J	1.2 e6 J
Locked mode amplitude	1.14 e-6 T	5.72 e-5 T
Safety factor q95	1.0	1.0
Normalized plasma pressure	6.91 e-3	Not available on JET
Input beam torque	1.47 Nm	Not available on JET
Electron temperature profile	9.53 e-1 keV	1.53 keV
Electron density profile	1.47 e19 m-3	2.98 e19 m^{-3}

Model Architecture - Outputs

- Output of the dilated convolutional layer blocks → final fully-connected layer → **disruption score**
- **<u>Disruptive shot:</u>** If disruption score is greater than (>) "alarm threshold" *<u>before the "warning time"</u>* → **TP (true positive) result**
- **<u>Clean shot:</u>** If the disruption score exceeds (>) the "alarm threshold" at any time → **FP (false positive) result**
- **<u>Results Summary: ROC Curve</u>**: Changing the 'alarm threshold' produces "receiver operating characteristic" results – plotted as "AUROC" Area Under Receiver Operating Curve)



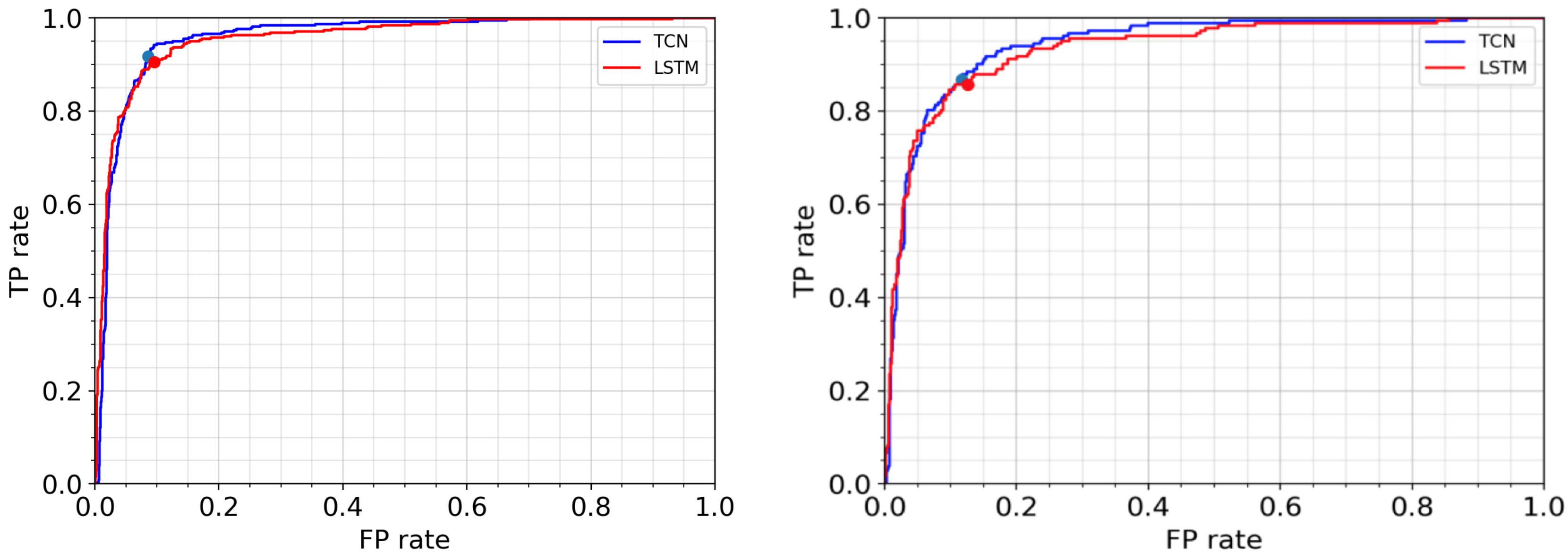
Model Architecture - Hyperparameters

Hyperparameter	EXPLANATION	REPRESENTATIVE VALUE
η	Learning rate	9.08 e-5
γ	Learning rate decay per epoch	0.99
N_{batch}	Training batch size	
$T_{warning}$	Warning time for target function, which becomes positive at $T_{warning}$	20
Target	Type of target function	ttd (function linear in time to disruption)
Nt	Number of causal temporal convolutional layers	8
Ns	Number of spatial convolutional layers	2
λ	Weighting factor for positive examples	16
K_t	Size of temporal convolutional filters	11
K_s	Size of spatial convolutional filters	7
N_{Tstack}	Number of stacks of temporal convolutional blocks	2
n_{tf}	Number of temporal convolutional filters	60
n_{sf}	Number of spatial convolutional filters	20
Dropout	Dropout probability	0.05

Results – AUROC (“Area Under Receiver Operating Curve”)

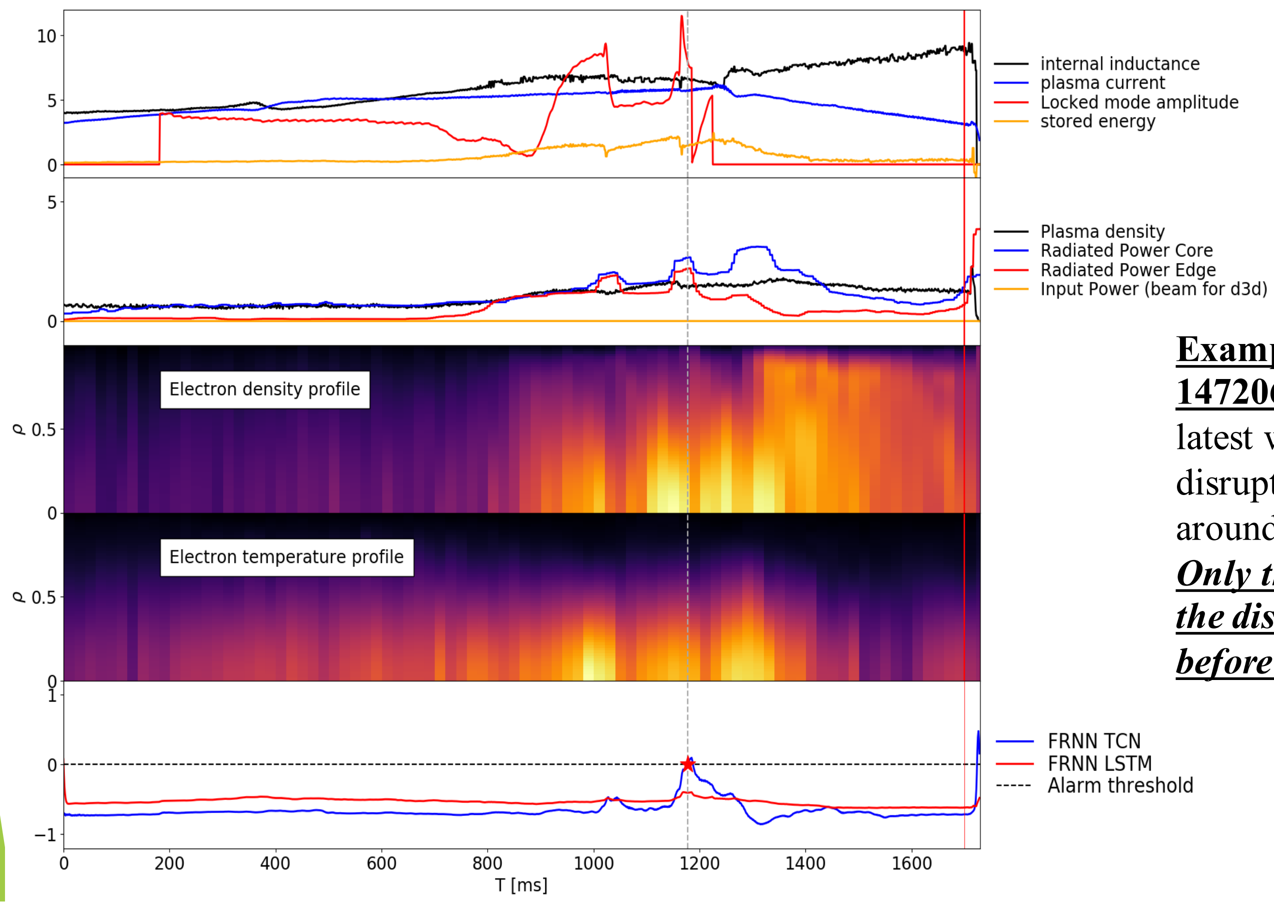
	Single machine				Cross Machine	
Training (#shots)	DIII-D (1702)			JET-CW(1956)	DIII-D (2268)	JET-CW(1956)
Validation (#shots)	DIII-D (837)			JET-CW(962)	DIII-D (1117)	JET-CW(962)
Testing (#shots)	DIII-D (846)			JET-ILW(1133)	JET-ILW(1133)	DIII-D(846)
Warning time	30ms	0.2s	1s	30 ms	30ms	30ms
FRNN 0D-LSTM	0.93	0.90	0 .72	**0.95**	0.81	**0.76**
FRNN 0D-TCN	0.93	0.90	0.74	0.95	**0.91**	0.73
FRNN 1D-LSTM	0.93	0.89	**0.80**		0.84	
FRNN 1D-TCN	**0.93**	**0.91**	0.79		0.89	

Results: ROC for Single Experimental Facility (D3-D)



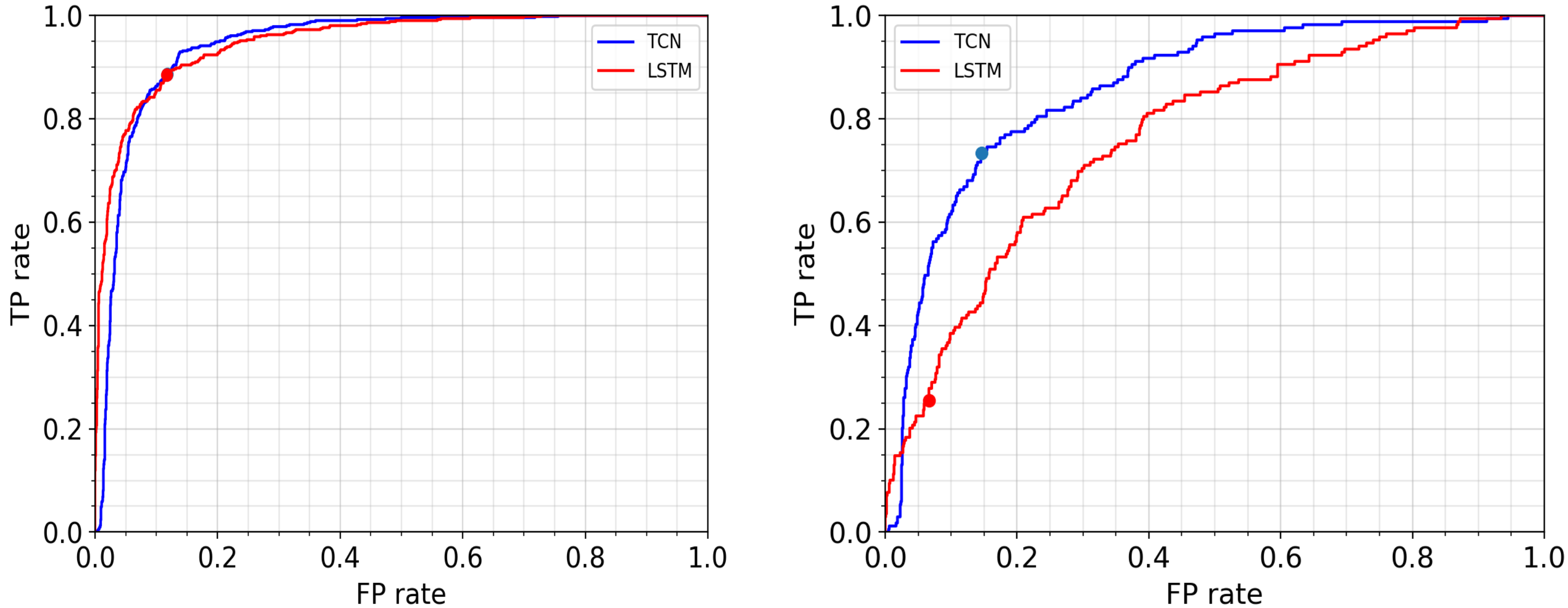
Comparison of ROC curves on the DIII-D training (left panel) and test (right panel) dataset with 0.2s warning time for the optimal FRNN 1D models, based on the TCN (blue) and LSTM (red) architectures. The solid dots indicate model performance at the optimal alarm threshold determined on the validation set.

Results: Single Experimental Facility (DIII-D) Example



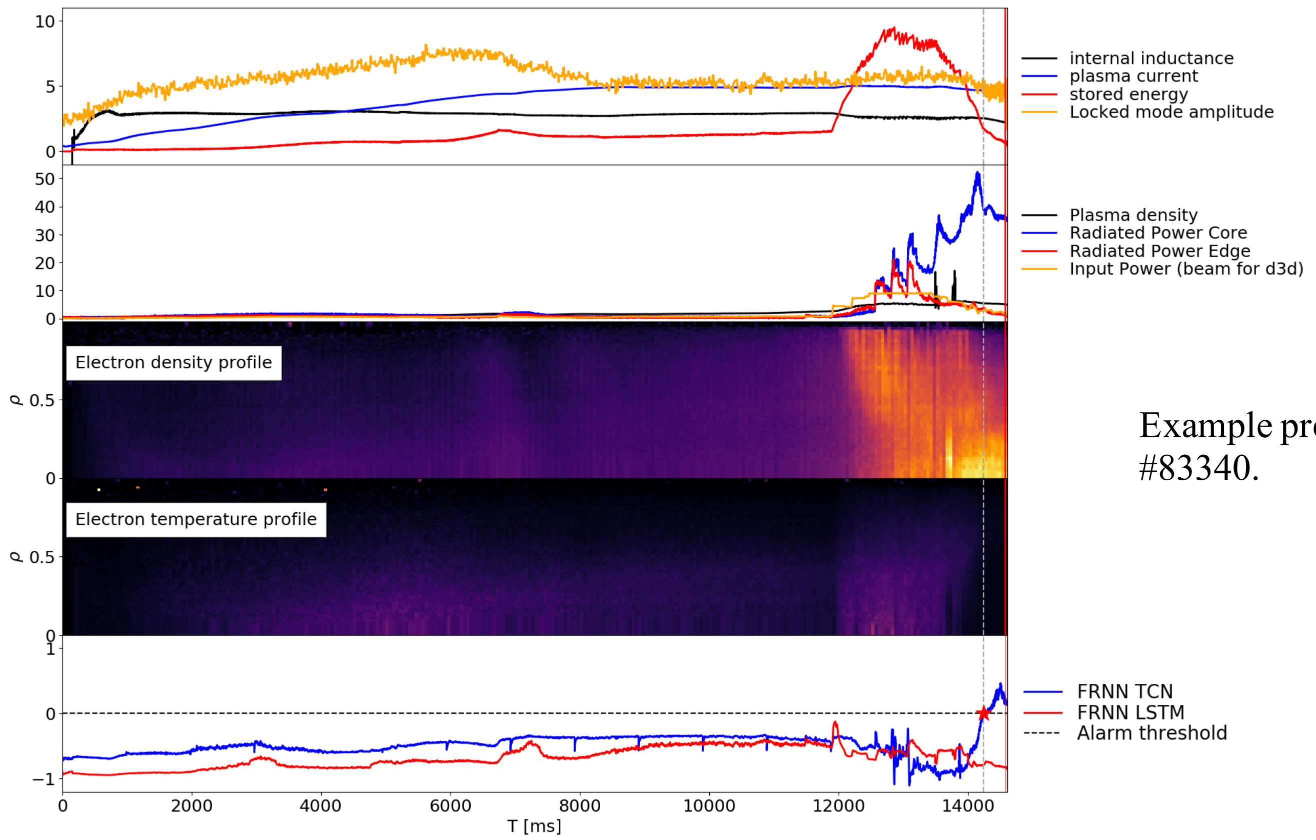
<u>**Example prediction on DIII-D shot # 147206:**</u> The solid vertical red line shows the latest warning time (30ms before the disruption). Both models respond noticeably around the indicated disruption alarm time. <u>***Only the TCN based model correctly triggers the disruption alarm around 0.5 second before the actual disruption.***</u>

Results: Cross-Device (DIII-D and JET) Example



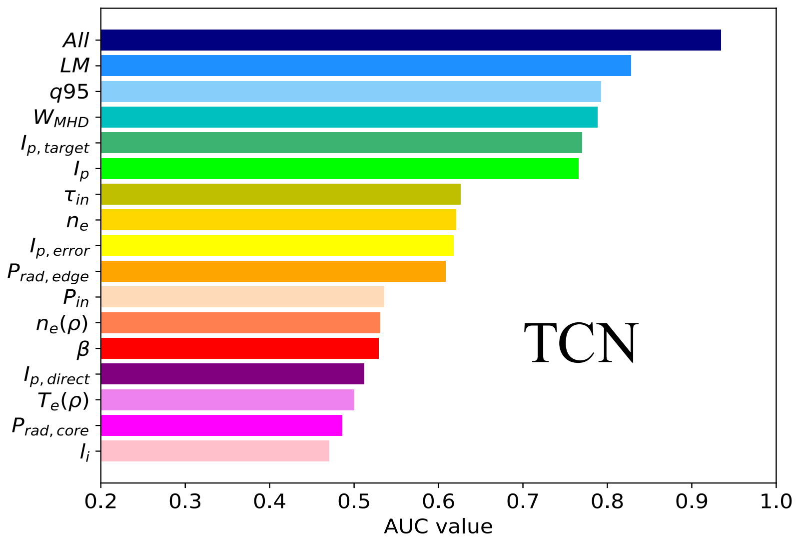
Comparison of ROC curves on the DIII-D training (left panel) and JET test (right panel) dataset for the optimal FRNN 1D models, based on the TCN (blue) and LSTM (red) architectures. **The solid dots indicate model performance at the optimal alarm threshold determined on the validation set.**

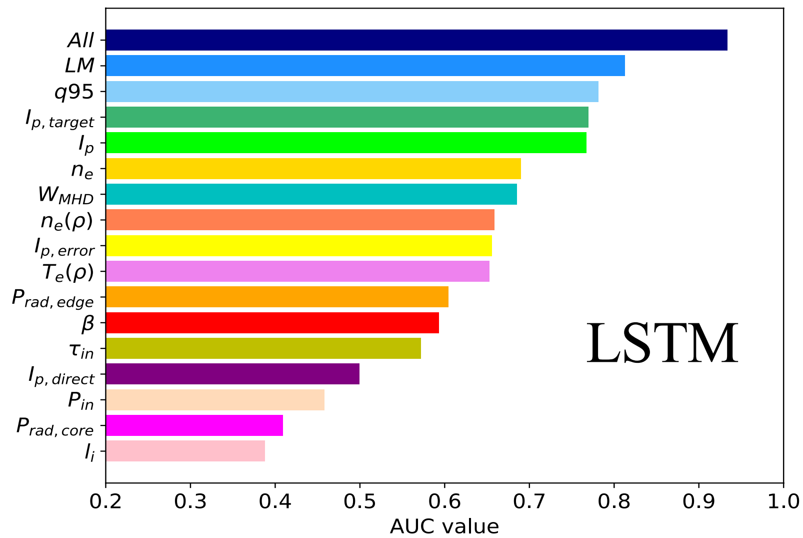
Results: Cross-Device Example



Example prediction on JET shot #83340.

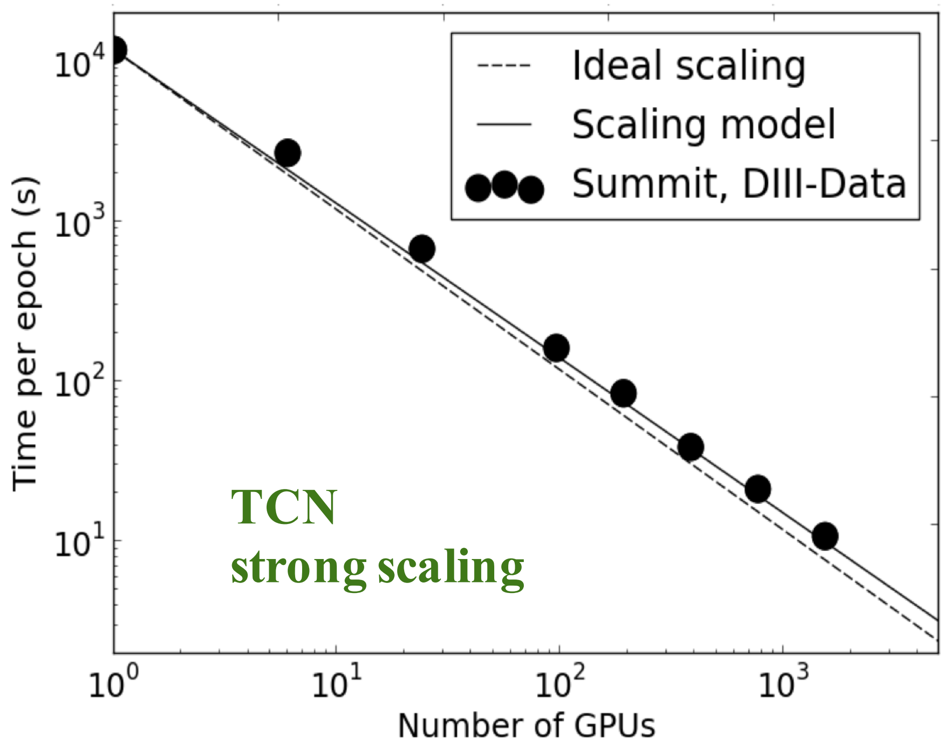
Results: Signal Importance Studies





Each bar represents the test set AUC values achieved by a model trained on the single labeled signal.

Computational Performance:
— strong scaling



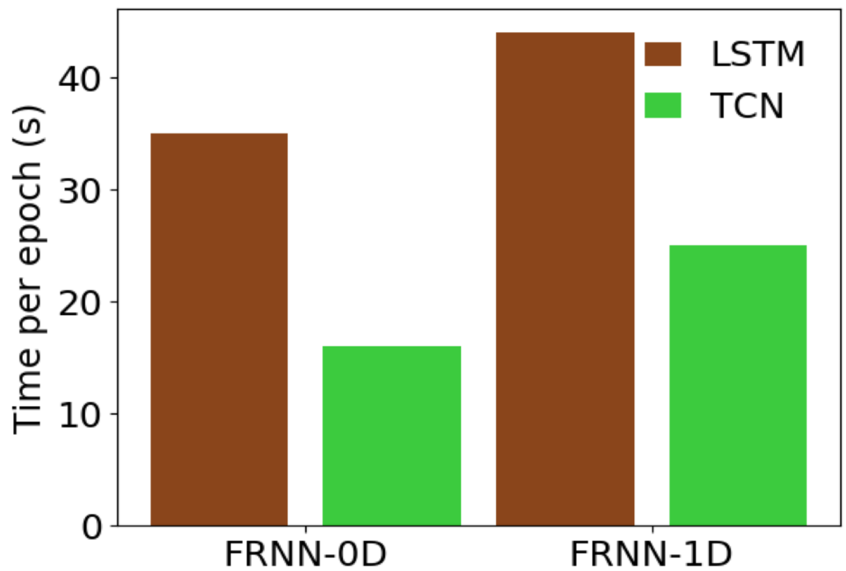
Computational Performance:
— models comparison of "time per epoch"

Time per epoch (i.e., the time required to complete one pass over the entire training dataset) during training:

-- 4 Tesla V100 processors used for FRNN-0D and FRNN-1D, for LSTM (brown), and TCN (green) architectures, respectively.

-- Lower values correspond to better computational performance.

-- The 4 models here correspond to the best performing ones from the studies of D3-D single machine disruption predictions with 30ms warning time.



Ongoing & Future Development of the FRNN Software Suite

FRNN [NATURE (April, 2019)]

AI/Deep Learning Model

- Keras API
- LSTM based models

Input

- 0D+1D data

Output

- Disruption score

FRNN [2020]

AI/Deep Learning Model

- Keras API / Pytorch API
- LSTM / TCN / TTLSTM based models

Input

- 0D+1D + 2D data

Output

- Disruption score + real time sensitivity score
- Physics-based signals

Summary

- We have trained a **fully convolutional spatio-temporal deep-learning model** for the exemplar application of tokamak disruption predictions.
- In addition to the LSTM capability for addressing temporal data information in FRNN, ***we have now implemented and applied the "temporal convolutional neural network (TCN)" architecture to comprehensively process the time-dependent input signals.***
- This **advance allows convolution operations to carry the majority of the computational load of training**, thus ***enabling a reduction in training time, and further optimizing the effective use of high-performance computing (HPC) resources.***
- The TCN based architecture achieves **equal or better predictive performance** when compared with the LSTM architecture for a large, representative fusion database.
- A new **deep learning predictive platform** is introduced here ***with flexible architecture selection options, capable of being tuned for an increasing variety of challenging tasks.***