Bayesian neural network for predicting disruptions using EFIT and diagnostic data in KSTAR

[1]JinSu Kim, [2]JeongWon Lee, [3]Jaemin Seo, [4]Young-chul Ghim and *[1]Yong-Su Na
asdwlstn@snu.ac.kr, *ysna@snu.ac.kr
[1]*Department of Nuclear Engineering, Seoul National University, Seoul, Korea*
[2]*Korean Institute of Fusion Energy*
[3]*Department of Physics, Chung-Ang University, Korea*
[4]*Department of Nuclear and Quantum Engineering, Korea Advanced Institute of Science and Technology, Korea*

Plasma Laboratory for Advanced REsearch

2023.10.27

Contents

Introduction – Disruption

- **Definition and general process**
  - ❑ **Definition**
    - Global and sudden losses of plasma with large amount of energy loss.
    - 4 phases: Pre-precursor phase → Precursor phase → Fast phase → Current phase
  - ❑ **Process of disruption**
    - The **evolution of an unstable current profile →** growth of a tearing mode
    - A **sudden relaxation of the equilibrium**: current profile flatten + loss of confinement + collapse of T_{plasma} → **Thermal quench (TQ)**
    - The **total current decays** → **Current quench (CQ)**
    - E_{ϕ} ↑ associated with Z_{plasma} ↑: generates **runaway electrons → Large current**
    - Loss of plasma energy + current decay

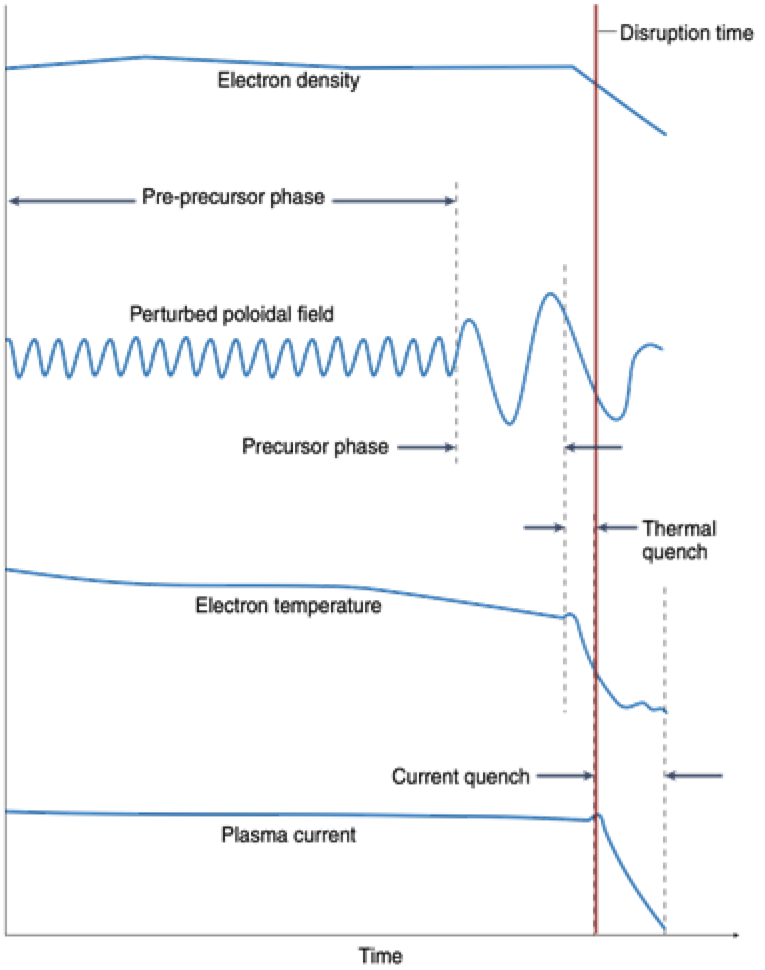
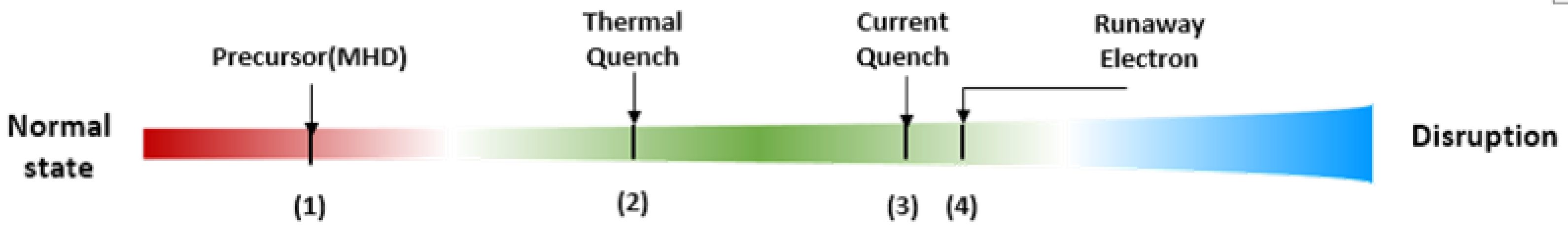


Fig 1, J.Vega et al, 2022, Nature Physics



Introduction – Disruption Prediction

- **Importance of disruption prediction**

❑ **Severeness of the disruption in tokamak device**

- Disruptions→ erosion / melting / structural damage in Tokamak device
- **Predicting disruption well in advance** is important **to mitigate** and **to avoid disruptions**.

❑ **Related work**

- Physics-based approaches by MHD theory and simulation: DECAF (2020)
- Data-driven approaches (ML/DL) can be alternative for disruption predictor.

❑ **Various attempts based on data-driven approach**

- Kates et al (2019): Fusion recurrent neural networks in JET and D3D
- Croonen et al (2020): SVM, RF, GBT → Ensemble learning
- Ferreiral et al (2020): CNN models with Plasma tomography (image) in JET
- R.M.Churchil et al (2021): Dilated TCN with ECE profiles in D3D
- E.Aymerich et al (2022): CNN with plasma profile(Bolometer diagnostic, Thomson scattering) in JET

Introduction – Disruption Prediction

- **Related work: Deep learning application for multimodal data in disruption prediction**

❑ **Disruption Prediction and Analysis through Multimodal Deep Learning in KSTAR [Jinsu Kim et al., FED, submitted)**

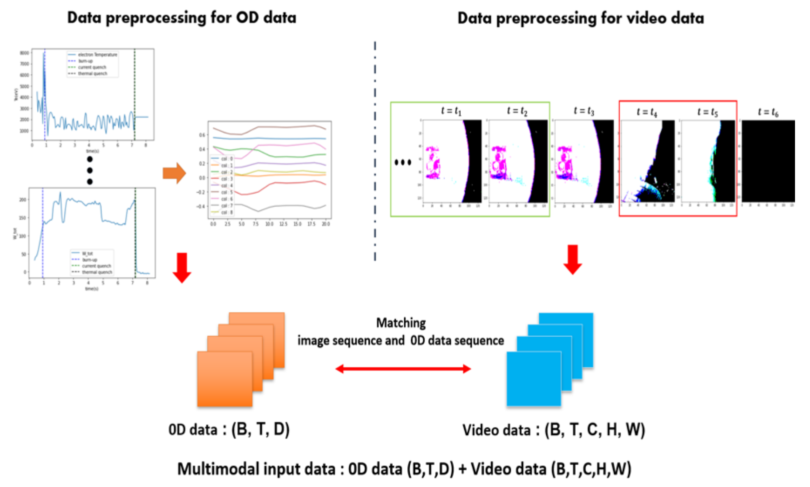
- Multimodal learning: Meta-learning for 2 or more different modalities of data (e.g. Video data + time-series data)

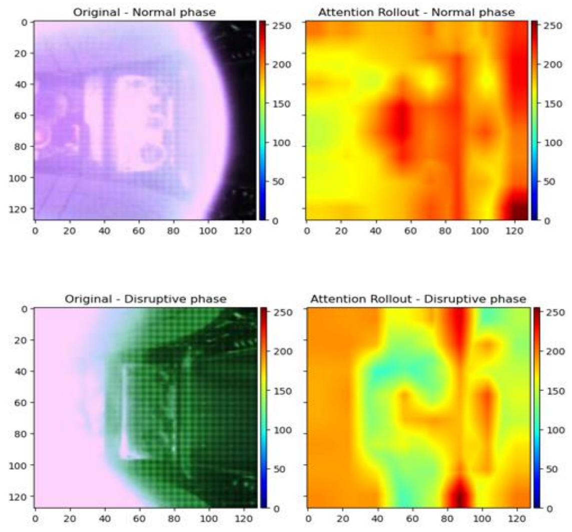
  → Improved capabilities + Robustness to data noise + Improved accuracy by multi-modalities

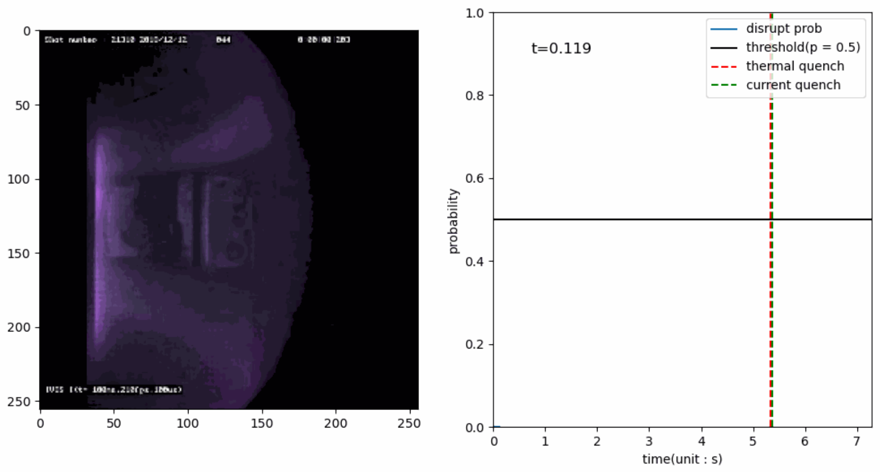
- Data structure: Video (IVIS Image sequence) + Time-series (0D parameters)

  → Video data: Spatial-temporal information including time-varying position and shape of plasma

  → 0D parameters: Physical attributed features for state of plasma







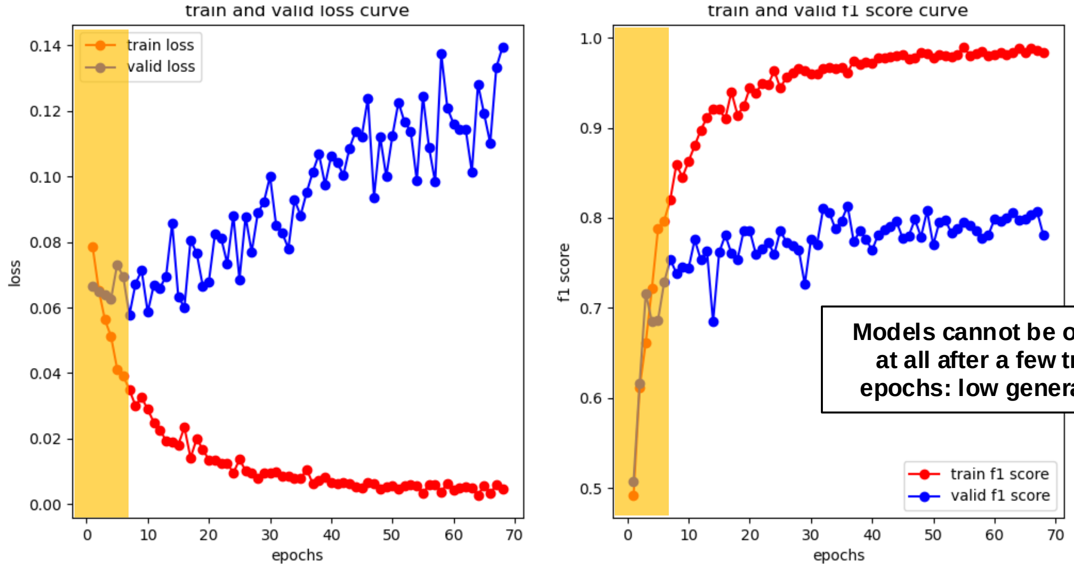
Introduction – Disruption Prediction

- **Issues on disruption predictions using deep learning : Overfitting**

❑ Overfitting

- Low generalization due to the **tendency of fitting closer to the training data** than to the underlying distribution
- ↑ model complexity or few data compared to model complexity → generalization error ↑
- Generalization: how to discover general patterns from given data
- Underfitting: limiting the reduction of training error due to the low complexity or small data
- **↑ modalities** → ↑ capacities + representation but sub-optimal + overfitting causes due to different generalization rates

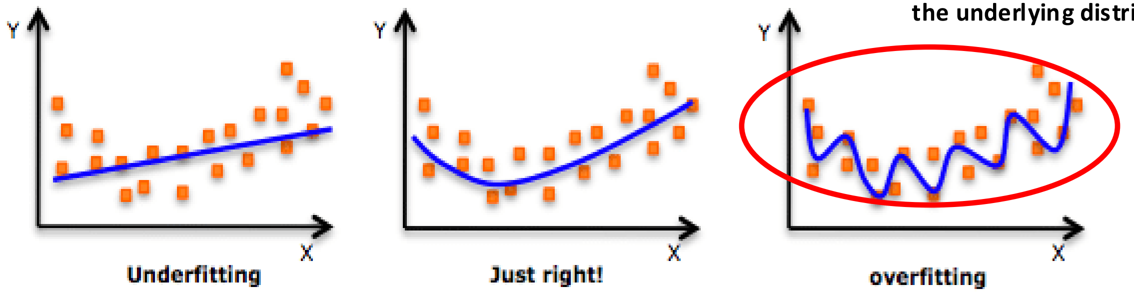
Training loss curve for Transformer models with EFIT data only



Models cannot be optimized at all after a few training epochs: low generalization

Overfitting

Fitting to the data noise rather than the underlying distribution



Pictures from https://www.analyticsvidhya.com/blog/2021/06/complete-guide-to-prevent-overfitting-in-neural-networks-part-1/

PLARE

Introduction – Disruption Prediction

▪ Issues on disruption predictions using deep learning : Overconfidence

❑ Overconfidence

- Overconfident prediction when neural networks provide a confidence interval
- ReLU networks susceptible to O.O.D examples (Guo et al., 2017), always overconfident far away from the data (Hein et al., 2019)
- **The neural networks can not be aware of their predictions' uncertainty based on general approach**

False alarms: overconfidence of models

Multi-modality can be one of the solutions, but too expensive

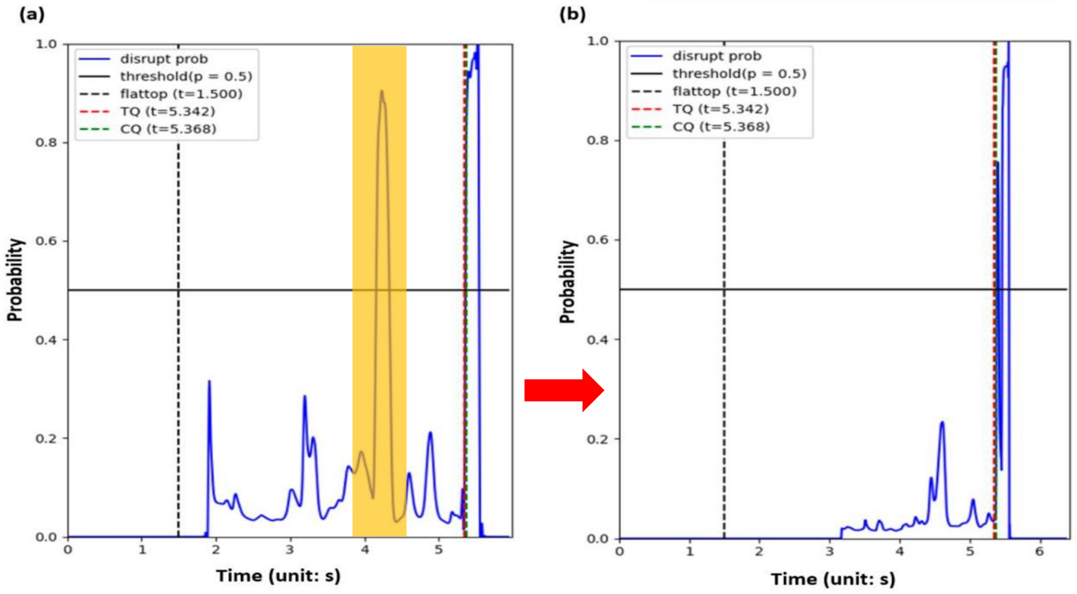
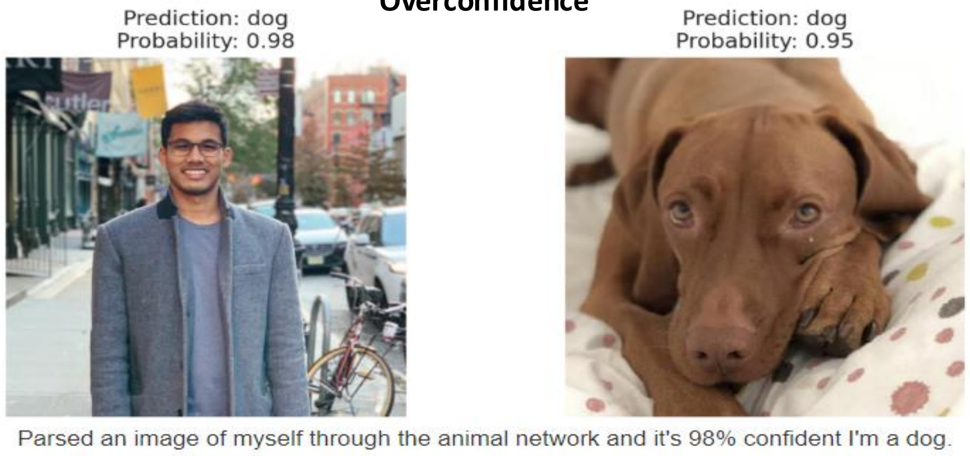


Fig 17. Result of continuous disruption prediction of shot 21310 in KSTAR with (a) Transformer and (b) Multimodal model with a prediction time of 95.20 ms

Overconfidence



Pictures from https://jramkiss.github.io/2020/07/29/overconfident-nn/

Introduction – Bayesian Neural Network

- **How to handle these issues**
  - Learning uncertainties is needed → **Bayesian approach should be applied!**
    - **Stochastic neural networks trained by variational inference: computation of the uncertainties + scalability + small dataset**
    - **Maximize a Posterior (MAP):** Maximize Likelihood Estimation (MLE) + Regularization, **robustness to overfitting**
  - Conventional approach (Frequentist view): weights of the neural networks are trained by maximum likelihood estimation (MLE)
  - Weights as random variables: Finding the optimal weights = Maximum a posteriori (MAP) weights

$$W^{MLE} = argmax_W \log P(D|W) \qquad W^{MAP} = argmax_W \log P(W|D) = argmax_W \log P(D|W) + logP(W)$$

  - Bayesian by Backpropagation (Charles Blundell et al, 2015)

$$\theta^* = argmin_\theta KL[q(w|\theta)||P(w)] - E_q[logP(D|W)] = argmin_\theta F(D,\theta)$$

  - Variational inference: intractable in general cases, but variational approximation by MC sampling can reduce the computational cost and handle intractability.

$$F(D,\theta) = \sum \log q(w|\theta) - logP(w) - logP(D|w)$$

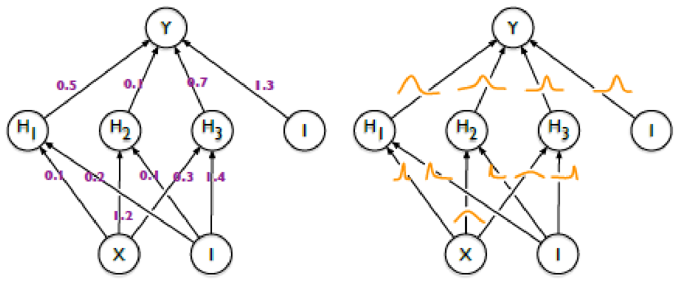


Figure 1. Left: each weight has a fixed value, as provided by classical backpropagation. Right: each weight is assigned a distribution, as provided by Bayes by Backprop.

Charles Blundell et al, 2015

Introduction - uncertainties

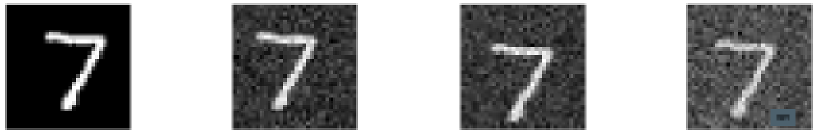
▪ Computation of the uncertainty

- **Aleatoric uncertainty vs Epistemic uncertainty**
- **Aleatoric uncertainty**: data uncertainty, due to the random nature of the physical systems
- **Epistemic uncertainty**: model uncertainty, related to the probabilistic distribution of the model weights, due to the lack of knowledge of the systems (a low generalization of the model)

$$Var_q[P(y^*|x^*)] = E_q\left[y^*y^{*T}\right] - E_q[y^*]E_q[y^*]^T = \underbrace{\boxed{\int [diag\left[E_p[y^*]\right] - E_p[y^*]E_p[y^*]^T q_\theta(w|D)dw}}_{\textbf{Aleatoric uncertainty}} + \underbrace{\boxed{\int \left[E_p[y^*] - E_q[y^*]\right]\left[E_p[y^*] - E_q[y^*]\right]^T q_\theta[w|D]dw}}_{\textbf{Epistemic uncertainty}}$$

- Aleatoric uncertainty decreases with the increase of dataset, however epistemic uncertainty requests to refine the model
- We can calculate epistemic uncertainty from the Bayesian approach, thus we can get more accurate and reliable disruption prediction with considering Epistemic uncertainty.

$\gamma =$	0	0.1	0.2	0.3
aleatoric =	0.00099897	0.00099885	0.00099875	0.00099838



Aleatoric uncertainty computed by Bayesian VGG on MNIST dataset, Kumar Shridhar et al, 2019

$$\mathrm{Var}_q\big(p(y^*|x^*)\big) = \underbrace{\frac{1}{T}\sum_{t=1}^{T}\mathrm{diag}(\hat{p}_t) - \hat{p}_t\,\hat{p}_t^T}_{\text{aleatoric}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}(\hat{p}_t - \bar{p})(\hat{p}_t - \bar{p})^T}_{\text{epistemic}}$$

where $\bar{p} = \frac{1}{T}\sum_{t=1}^{T}\hat{p}_t$ and $\hat{p}_t = \mathrm{Sofplus_n}\big(f_{w_t}(x^*)\big)$.

Simple computation of aleatoric uncertainty and epistemic uncertainty, Kumar Shridhar et al, 2019

Introduction - strategies

Aims of this research

- **Prediction**: Forecasting before thermal quench (minimum: 40ms)
- **High accuracy**: Minimizing false alarm rates and missing alarm rates
- **Cause estimation**: Direct input feature importance computation for inferring causes

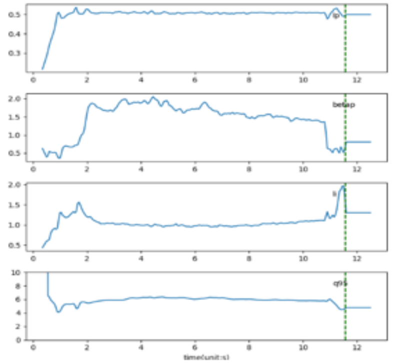
Key concepts

- **Bayesian neural network**: Stochastic neural network for covering overconfidence
- **Integrated Gradients**: Gradient-based feature importance computation algorithm
- **Dilated TCN**: Model architecture for handling multi-time scale data

Introduction - strategies

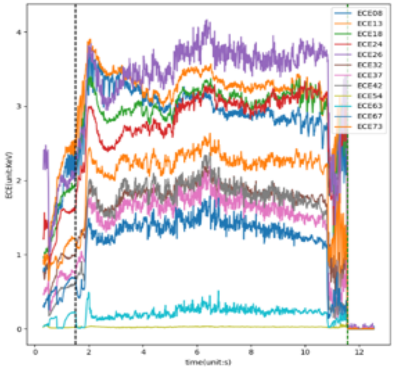
Dataset with multi-modalities containing multi-time scale signals

EFIT data



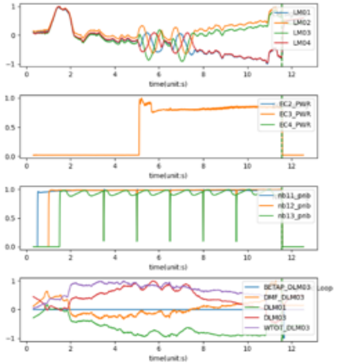
Time interval: 50 ms
Data point: 10

ECE data



Time interval: 10 ms
Data point: 50

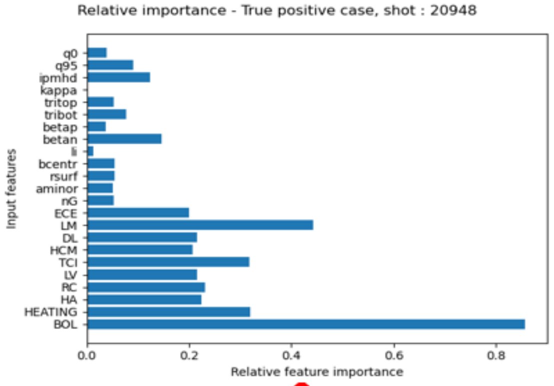
Diagnostic

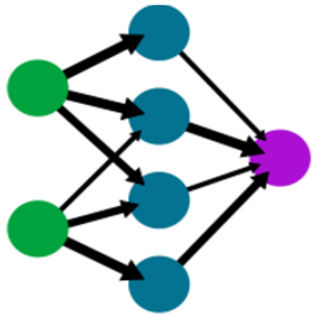


Time interval: 10 ms
Data point: 50

Forward computation

Direct feature importance computation (Integrated gradient)



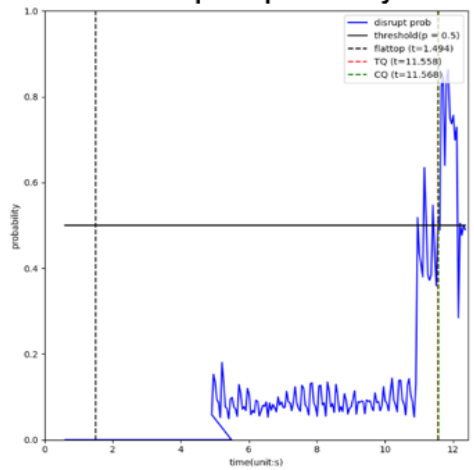


Neural Network (Disruption prediction model)

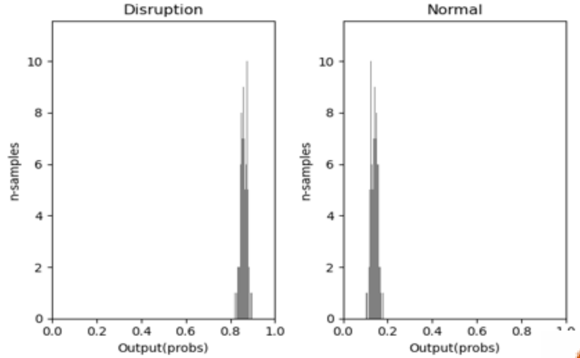
Forward computation

Figuring out the causes and classification of disruptions in advances

Disruption probability



Uncertainty of prediction



Covering false alarm cases by aleatoric uncertainties

Development – Dataset Construction

- **Expansion of the signals for enhancing the prediction accuracy**
  - **Causes of disruption (J.A.Wesson, Nucl.Fusion, 1989)**
    - **Density limit disruption**: Radiative contraction + Precursor instabilities + Energy quench + Current decay
    - **Low q-limit disruption**: Fast rise of n=1 perturbed magnetic field + Mode lock
    - **Current rise disruption**: Peaking of the current profile + Sawtooth + Density limit disruption
    - **Vertical instability disruption**: Elongated plasma forced by strong Lorentz force due to halo current and disrupted
  - **Disruption classification**
    - **Mode lock: LM signals**
    - **H/L transition: H89**
    - **Density limit: Radiation / density profiles**
    - **High radiated power: H-alpha**
    - **Internal transport barrier: 1D-profiles of temperature / density**
    - **Vertical displacement: Error fields, HCM(Halo current)**

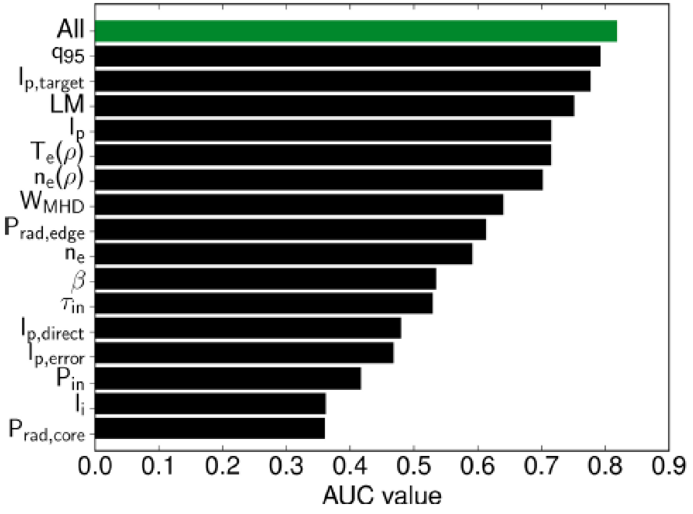
- plasma current [A]
- locked mode amplitude [T]
- radiated power [W]
- plasma density [m^{-3}]
- input power [W]
- internal inductance
- stored energy derivative [$J\ s^{-1}$]
- safety factor
- poloidal beta
- plasma centroid vertical position [m].

Input feature for disruption predictor from B.Cannas et al, 2006

PLARE

Development – Dataset Construction

- **Expansion of the signals for enhancing the prediction accuracy**
  - **Input features generally used in ML / DL**
    - EFIT: q95, Ip, P, li, B-field, R, a, ….
    - Diagnostic signals: Mirnov coil, LM amplitude,…



Julian Kates et al, 2019

TABLE I. Diagnostic dataset features.

S. No.	Signal description	Acronym	Units
1	Total input power	P_{total}	W
2	Plasma current	I_{plasma}	A
3	Plasma density	*Dens*	m^{-3}
4	Line integrated density	L_{Dens}	m^{-2}
5	Electron temperature	*Te_Probe*	T (eV)
6	Safety factor q_{95}	q_{95}	s^{-1}
7	Greenwald density	$ne_{Greenwald}$	D
8	Toroidal magnetic field	B_t	T
9	Vertical plasma position	*PVP*	Z_m
10	Horizontal plasma position	*PHP*	Z_m
11–14	Mirnov coil (four coils)	*Rad*	W

Jayakumar Chandrasekaran et al, 2022

name	description
I_{pla}	plasma current
MLA	mode lock amplitude
l	plasma internal inductance
W_{dia}	diamagnetic energy
$\dot{W}_{dia}$	time derivative of the diamagnetic energy
n_e	electron density
P_{out}	radiated output power
P_{in}	input power: sum of ICRH and NBI power
q_{95}	edge safety factor
B_ϕ	toroidal magnetic field strength

J.Croonen et al, 2020

- **Disruption classification: Tabular dataset is enough** $X \in R^D$

KSTAR environment = partially observable system
→ Sequential data

- **Disruption prediction: Tabular dataset is not enough, Time-series dataset is needed (Sequential data)** $X \in R^{TxD}$

Development – Dataset Construction

- **Expansion of the signals for enhancing the prediction accuracy**

- **EFIT**

Description	Variable
Plasma current	I_p
Normalized beta	β_n
Poloidal beta	β_p
Elongation	κ
Safety factor (edge)	q_{95}
Safety factor (core)	q_0
Major radius	R_C
Minor radius	a
Internal inductance	li
Triangularity – top	δ_{top}
Triangularity - bottom	δ_{bottom}
Toroidal magnetic field	$B_{toroidal}$

Time intervals: 50 ms
Data points (sequence length): 10

- **ECE**

Description	Variable
	ECE08
	ECE13
	ECE18
	ECE24
	ECE26
ECE with different channels	ECE32
	ECE37
	ECE42
	ECE54
	ECE63
	ECE67
	ECE72

Time intervals: 10 ms
Data points (sequence length): 50

- **Diagnostic data**

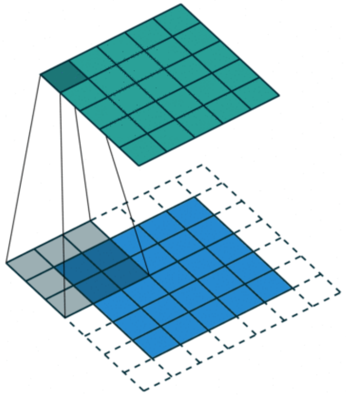
Description	Variable
Lock mode signals	LM01 ~ LM04
	HCMIL01 ~ HCMIL16
Halo current monitoring signals	HCMID01 ~ HCMID08
	HCMCD01 ~ HCMCD16
	HCMOD01 ~ HCMOD08
	Betap-DLM03
Diamagnetic loop	W_{tot}
	DMF-DLM03
	DLM01-DLM03
TCI	ne-tci01 ~ ne-tci05
Loop voltage	LV01-LV45
H alpha	TOR-HA01 ~ POL-HA10
EC heating	EC2-PWR ~ EC4-PWR
NB heating	NB11 – NB13
	RC01 ~ RC03
Rogowski coil	VCM01 ~ VCM03
	RCPPU1, RCPPL1

Time intervals: 10 ms
Data points (sequence length): 50

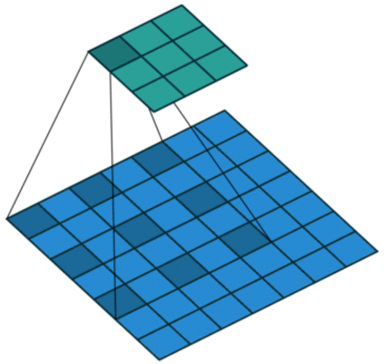
Development – Dilated temporal convolution network

- **Effective model architectures for multi-scale time series data**
  - Deep convolutional neural networks with dilated convolutions
    - Convolution layer: A layer that computes convolution products with different filters to extract the feature maps
    - Dilated convolution: extends the receptive field by adding zero padding (spacing) in kernel filters

*Receptive field: A size of input neurons' space that affects one neuron of the output layer

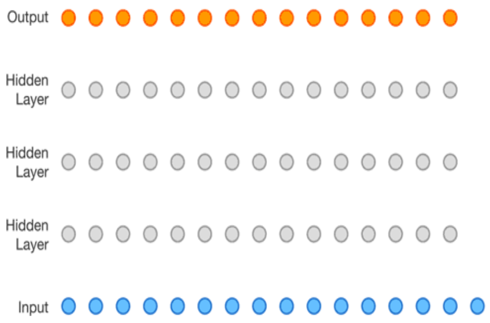


Convolution process with dilated rate = 1



Convolution process with dilated rate = 2

Adding zero padding to extend the receptive field



$$y[n] = \sum_i w[i]x[n - d \cdot i]$$

w: 1D dilated convolution filter of length k, x: input

- Temporal convolution networks with dilated convolutions can effectively separate out structures in multi-scale data.
- Long sequences can be covered by the increase of receptive fields due to dilated convolutions

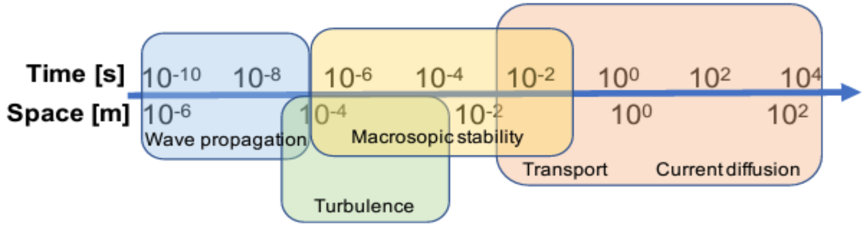


Figure 1 from R.M. Churchill, 2019

PLARE

Development – Integrated Gradients

▪ Gradient-based feature selection algorithm

- **Integrated Gradients**
  - The problem of **attributing the prediction** of a neural network to its **input features** can be approximated by **Integrated Gradients.**
  - Gradients of the output with respect to the input = a natural analog of the model coefficient, but breaks sensitivity

$$\mathrm{Grads}_i^{\gamma}(x) = \int \frac{\partial F(\gamma(a))}{\partial \gamma(a)} \frac{\partial \gamma_i(a)}{\partial a} da$$ **: Path integrated gradients**

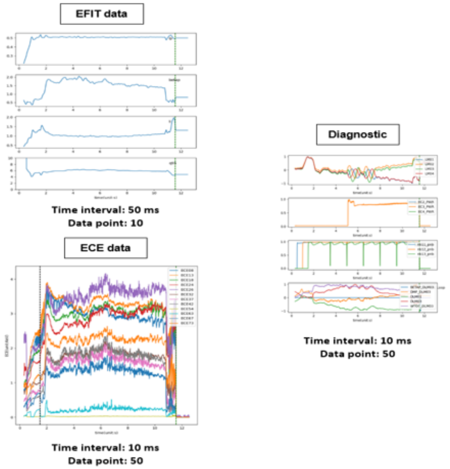
$$\frac{\partial F(x)}{\partial x_i}$$ **: Gradient of F along the i-th dimension at x (i-th feature)**

$$\boxed{(x_i - x_i') \times \frac{1}{m} \sum \frac{\partial F\left(x' + \frac{k}{m} \times (x - x')\right)}{\partial x_i}}$$ **: Approximation of path integrated gradients**

Sensitivity related to changes in features = Feature importance

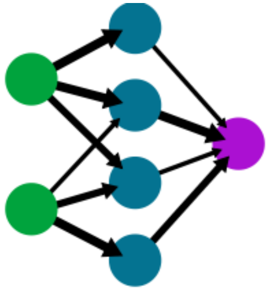
K-steps computation for enhancing the approximation

Input data



$$x' + \frac{k}{m}(x - x')$$

Differentiate input



Neural Network
(Disruption prediction model)

Forward computation

Backward computation

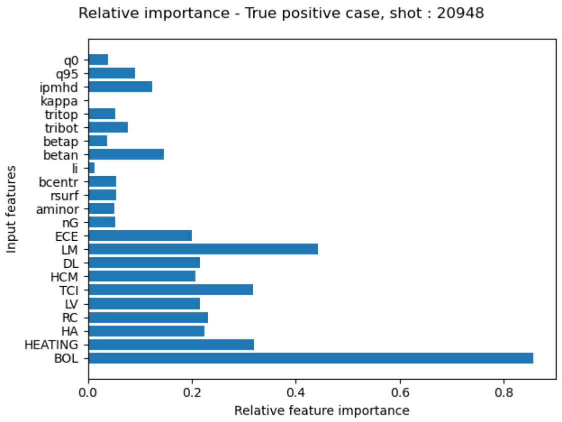
Output

$$y_i = \frac{\exp(o_j)}{\sum \exp(O_j)} : P(y_i | x_i, \theta, D)$$

$$\frac{\partial y(x)}{\partial x_i}$$

Gradient related to input feature i

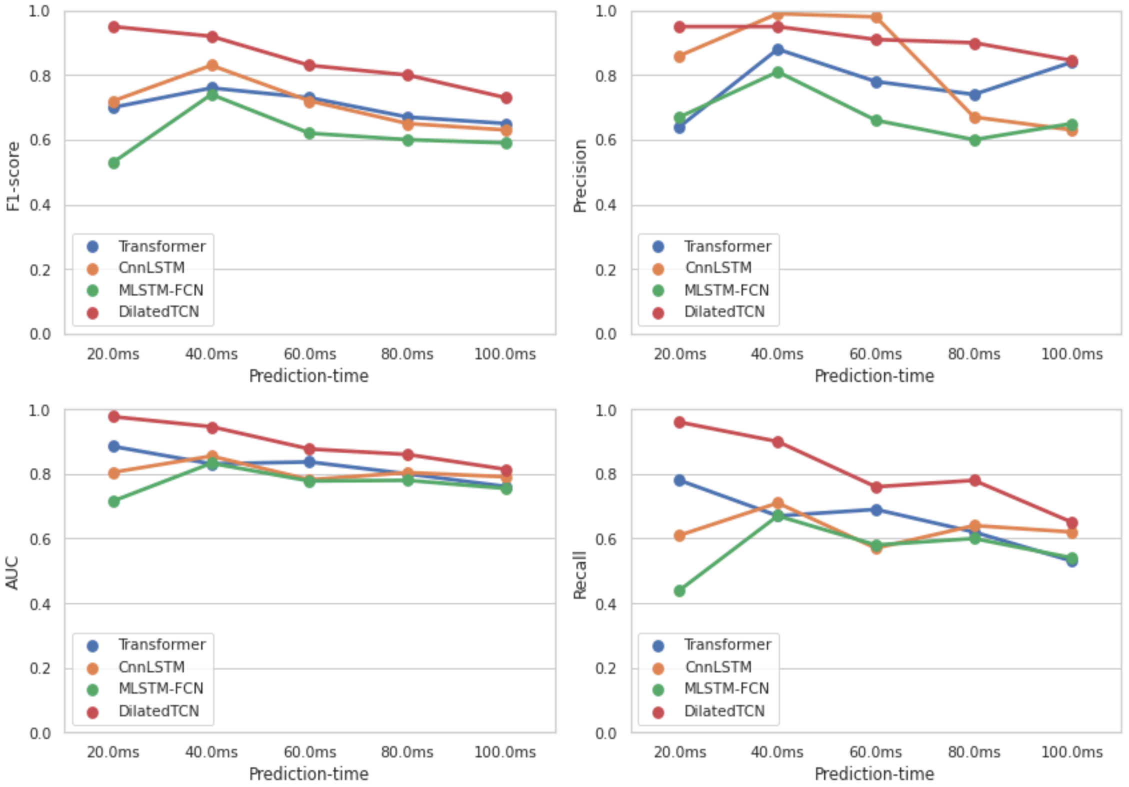
$$\boxed{(x_i - x_i') \times \frac{1}{m} \sum \frac{\partial F\left(x' + \frac{k}{m} \times (x - x')\right)}{\partial x_i}}$$

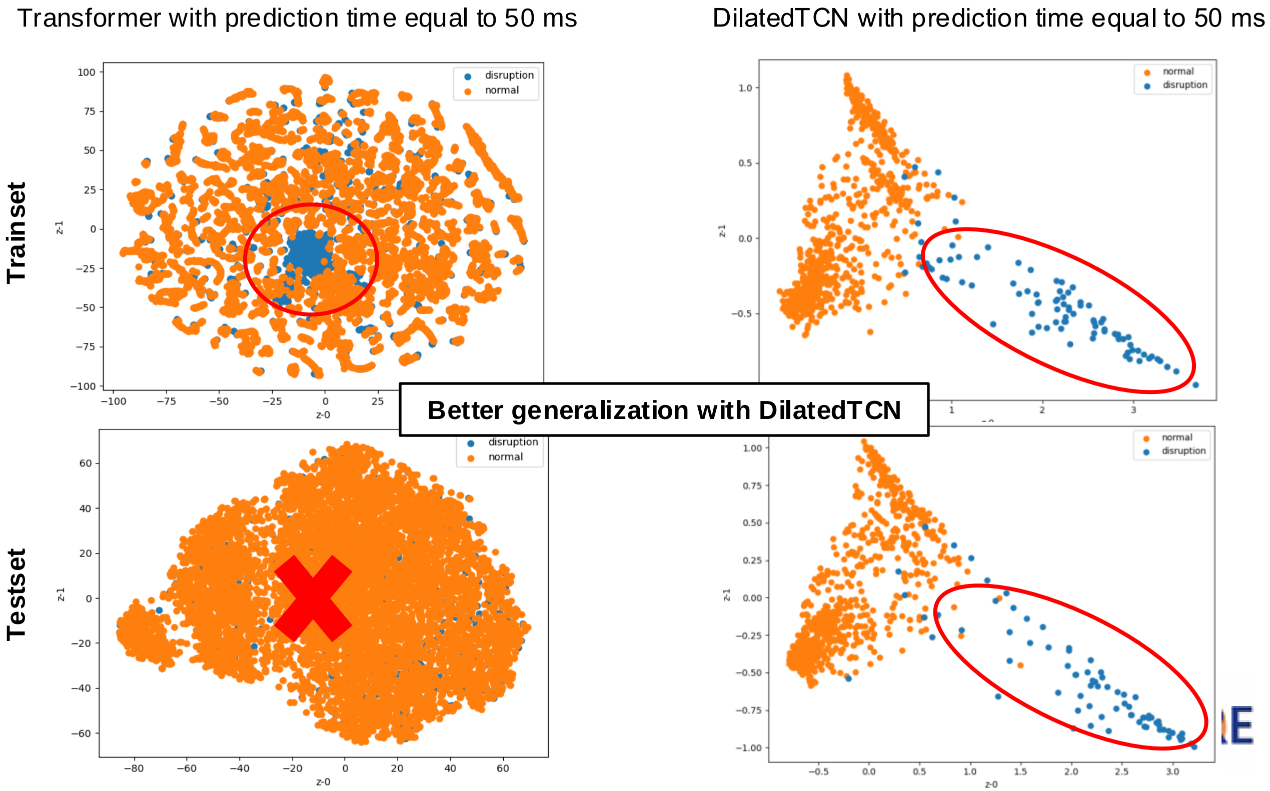


Results – Overall model performance

- **Model performance with different prediction times compared with previous research**
  - **Evaluation of disruption prediction models in advance of the current quench**
    - Models: Transformer, CNN-LSTM, MLSTM-FCN, DilatedTCN
    - Input features: EFIT + Diagnostic data (same as previous slides)
    - Data configuration (Features, sequence length, train-test set) and training strategies (Focal Loss + Deferred Re-weighting): equivalent

Model	Number of parameters
Transformer	1,447,170
MLSTM-FCN	1,898,370
CNN-LSTM	1,439,234
Dilated TCN	252,768

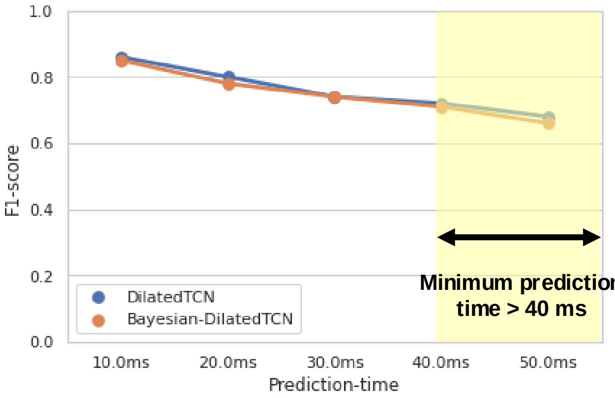


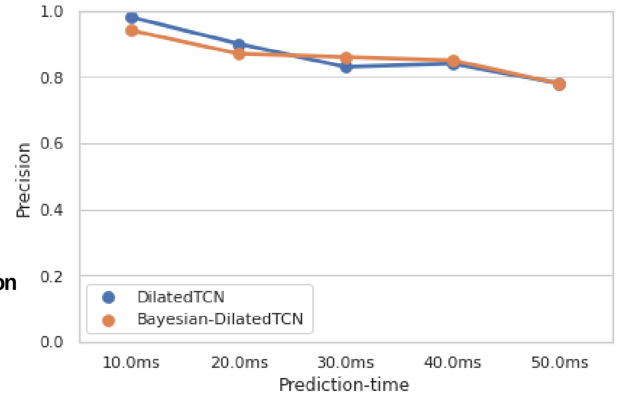


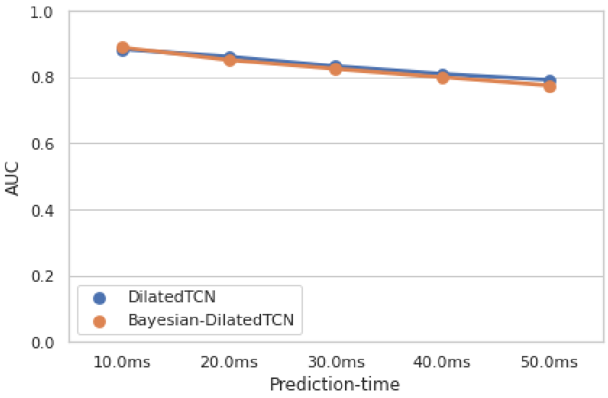
Results – Overall model performance

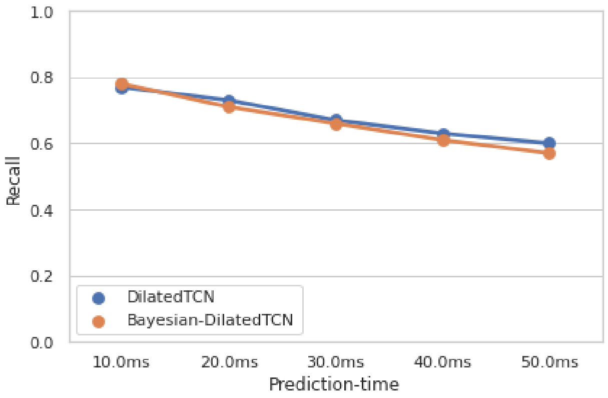
- **Model performance with different prediction times compared with Bayesian approach**
  - **Evaluation of disruption prediction models in advance to the thermal quench**
    - Models: DilatedTCN, Bayesian-DilatedTCN
    - Input features and data configuration: same as previous slide

No significant difference of model capabilities





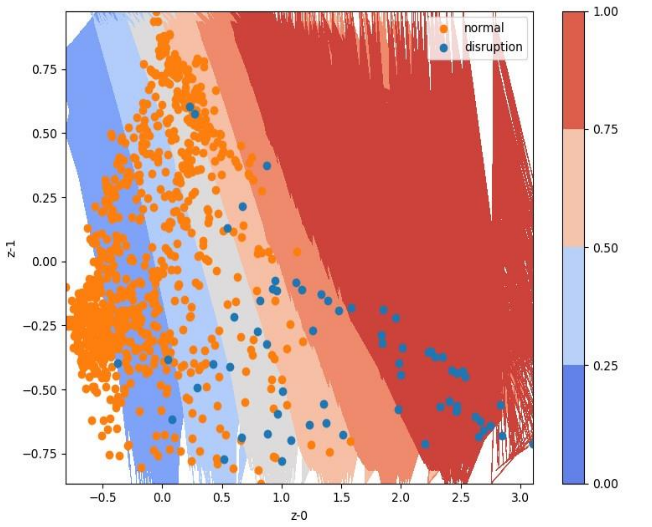


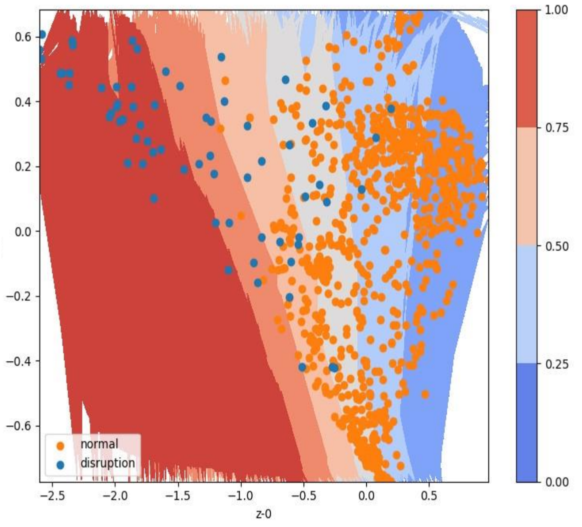


Dilated TCN with 40 ms

Testset

Bayesian Dilated TCN with 40 ms



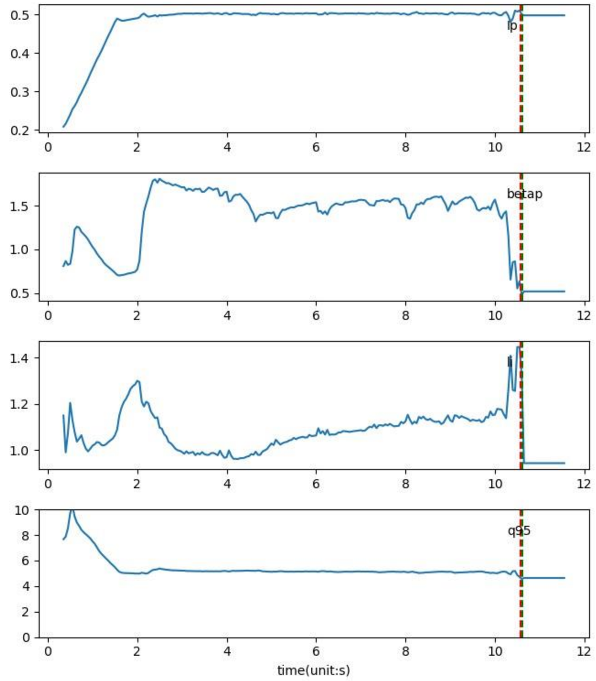


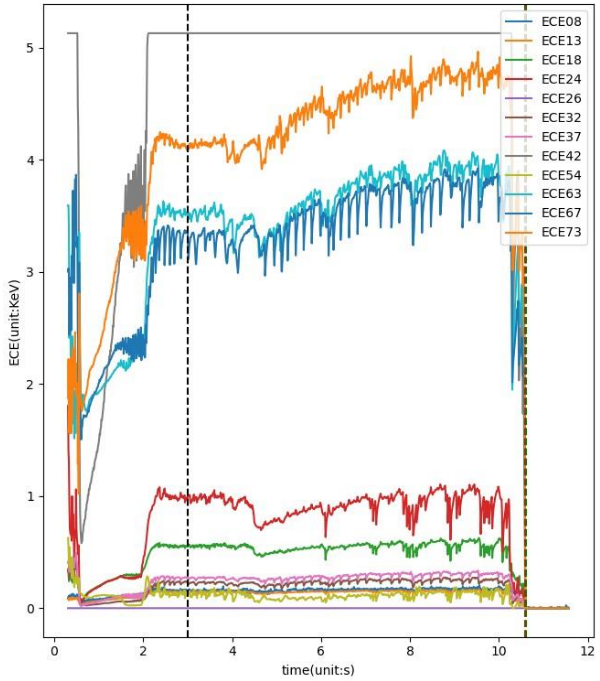
Is Bayesian not useful?

PLARE

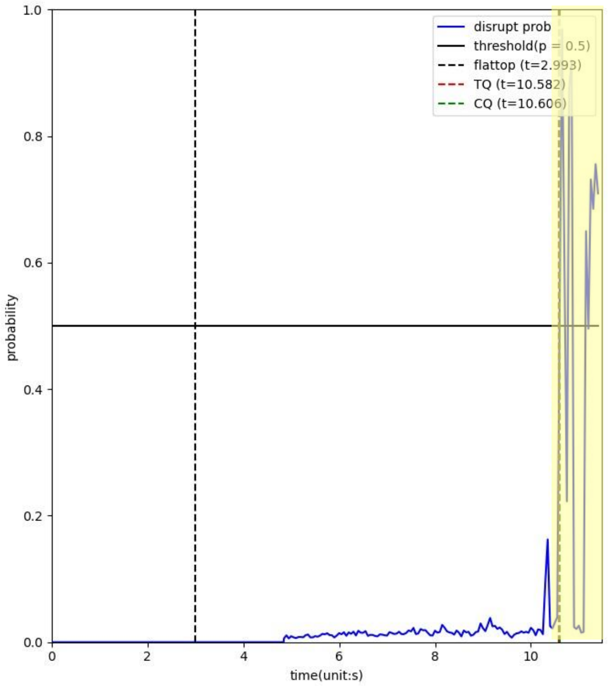
Results – Simulations for continuous disruption predictions

- **Simulation result for predicting disruptions in shot 20948 (1)**
  - **Predicting occurrences in advance to thermal quench and detecting indirect causes from LM disrupted experiment**
    - Setting: Bayesian Dilated TCN + prediction time 40ms + Locked mode plasmas (shot 20948)
    - Successful prediction + q95 and LM signals show high feature importance: detecting causes also possible



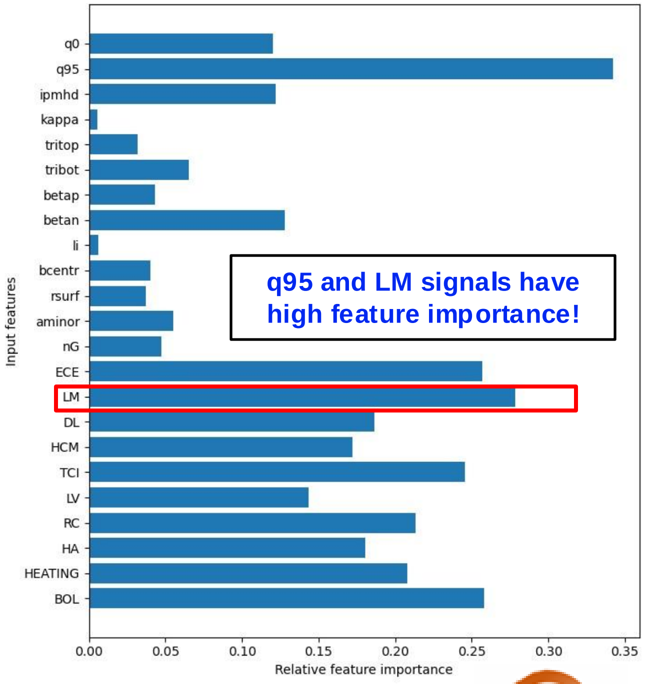


Disruption prediction with shot : 20948



Disruption was successfully predicted in advance to thermal quench with 40 ms.

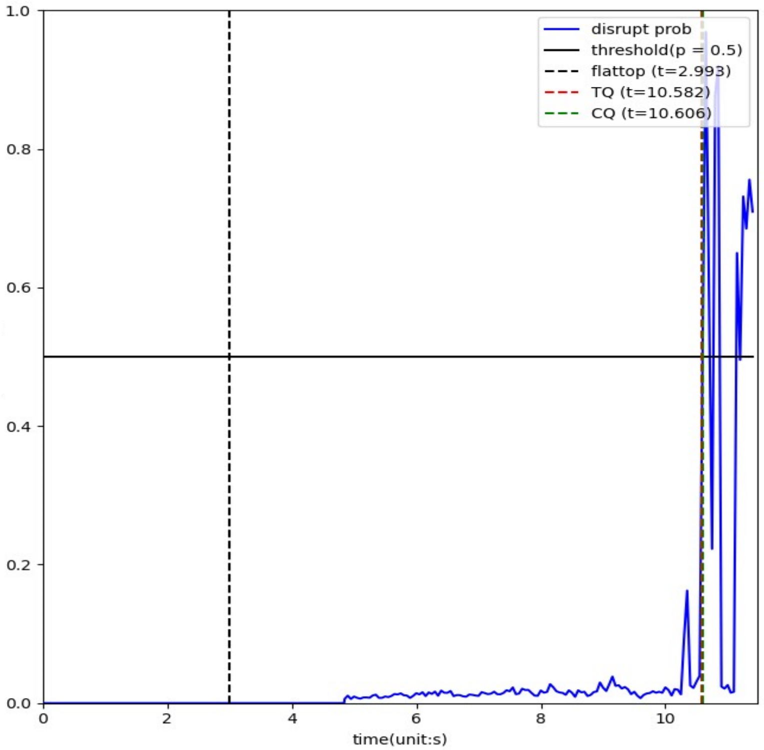
Input signal importance can be computed directly within the disruption predictions

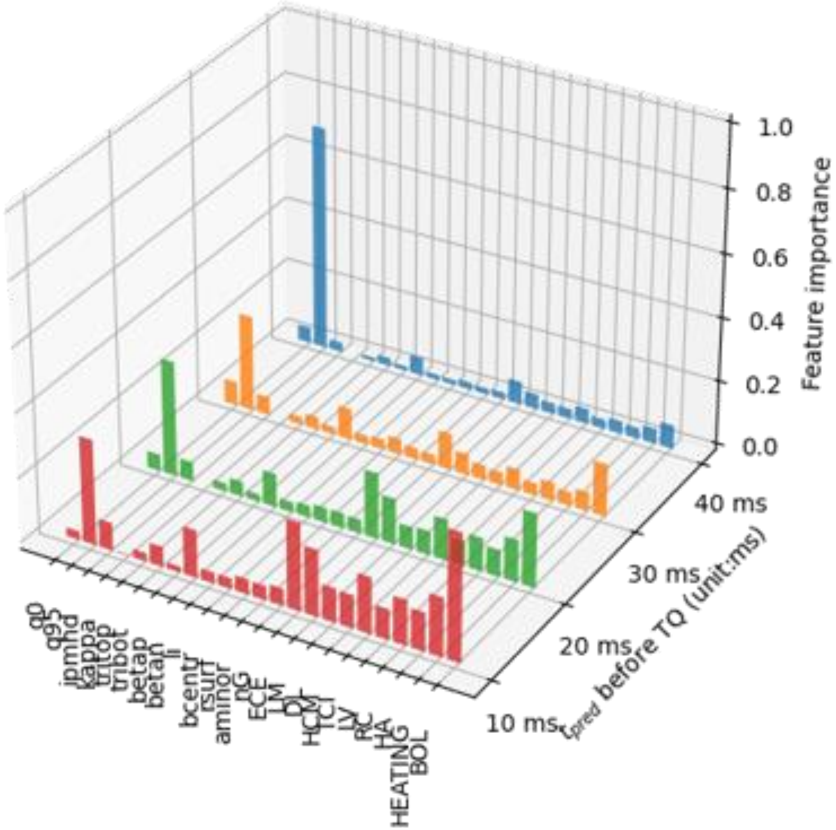


PLARE

Results – Simulations for continuous disruption predictions

- **Simulation result for predicting disruptions in shot 20948 (2)**
  - **Disruption probability curve and time-varying feature importance near the disruptive phase – thermal quench**
    - Input features with high importance: q95 (40ms) → q95, ECE, BOL (30ms) → q95, ECE, BOL, LM (20ms) → q95, ECE, BOL, LM (10ms)
    - Probable situation: Precursor (locking) → Current profile affected → Te decrease → radiation increase → thermal quench





TQ – 40ms — **Main feature: q95**

Current profile flattening

Te drastically decrease (ECE)

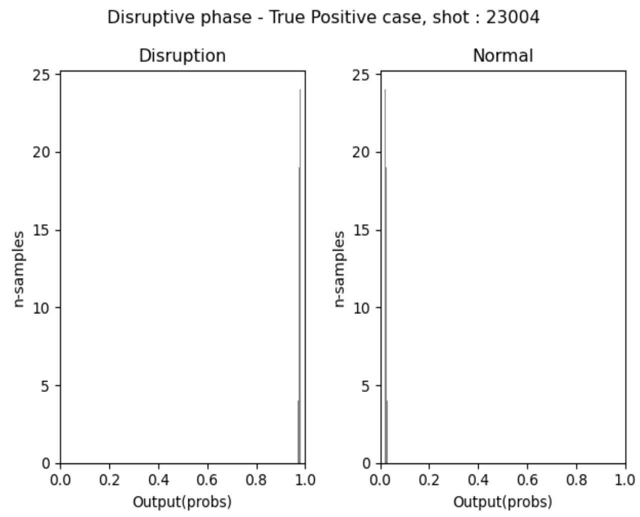
TQ – 10ms — **Main feature: LM, BOL**

ECE(Te) + Locked mode + Radiation

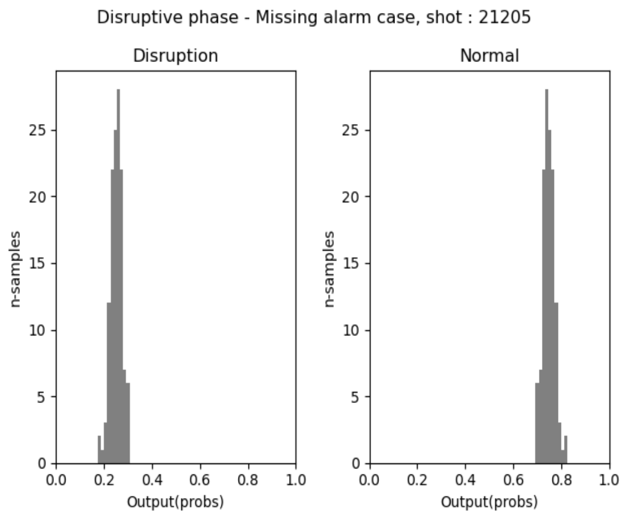
Thermal quench

Discussion – Disruption prediction and uncertainty computation

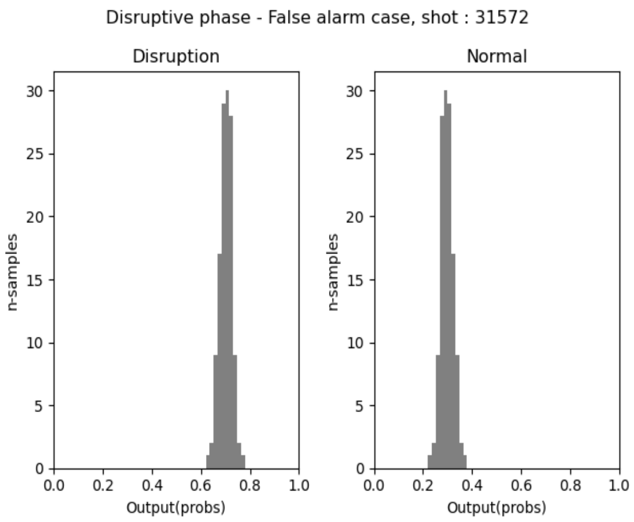
- **Uncertainty computations for several cases: True alarms, False alarms, Missing alarms**
  - **Aleatoric and Epistemic uncertainties of disruption predictions for feasible cases**
    - True positive case (True alarms): Low uncertainty + High average probability, good generalization with True positive data
    - False positive case (False alarms): the rate can decrease by constraining the upper limit of uncertainties
    - False negative case (Missing alarms): completely misunderstanding the way to predict disruptions → extending input signals or other relevant features should be used.



Low uncertainty + High average probability



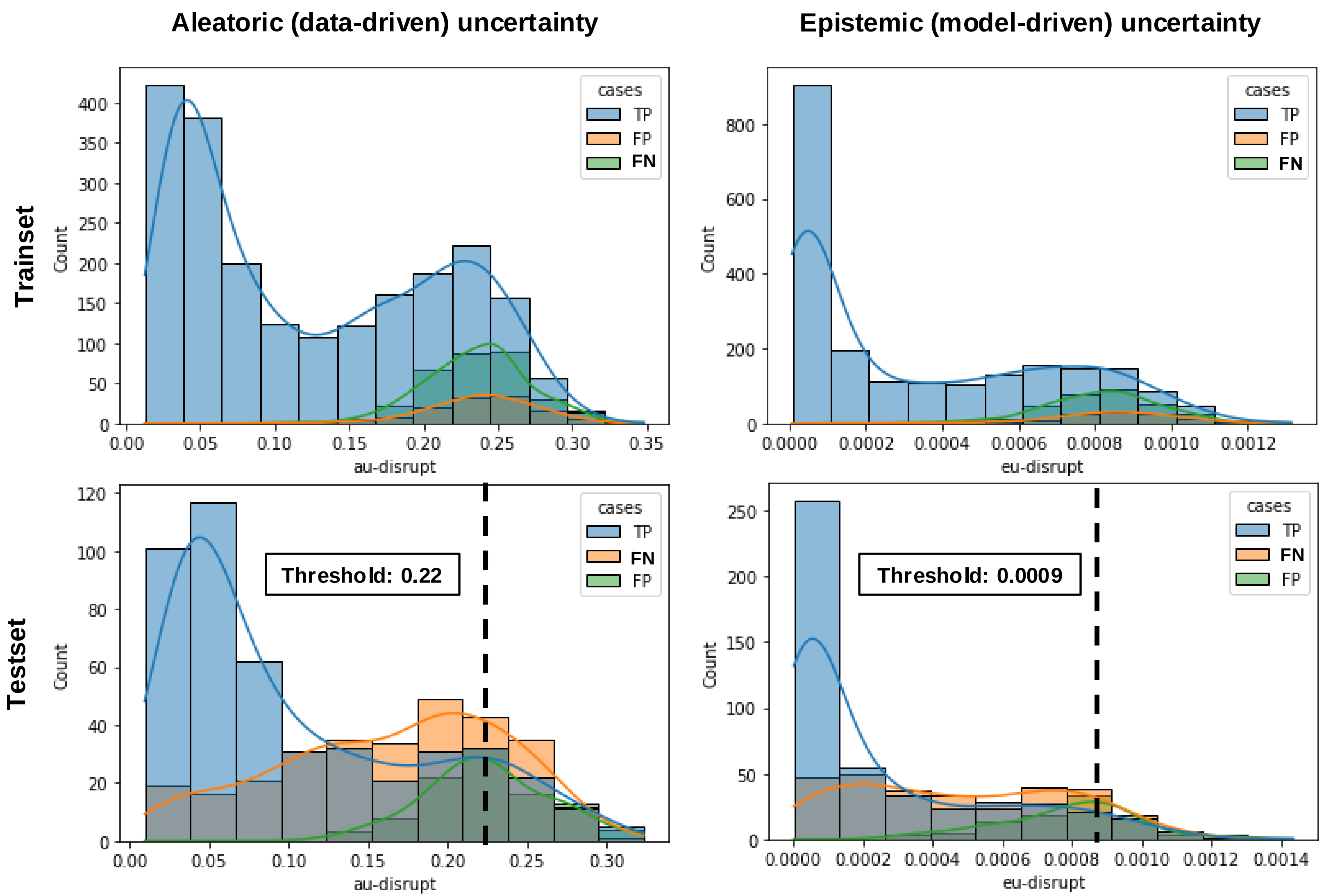
High uncertainty + High average probability



High uncertainty + Relatively low probability

Discussion – Disruption prediction and uncertainty computation

- **Analysis for aleatoric and epistemic uncertainty distribution for TP, FP, and FN cases**



- True & False alarms: X diff btw train & test
- Missing alarms: both uncertainties ↑

Low confidence about classifying non-disruptive state

Low generalization for missing alarm cases

Question: How to reduce FN and FP without a decrease in TP?

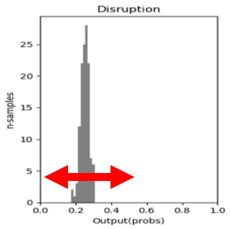
Option 1. FP↓ and FN↑, but TP↓

Option 2. FN↓ and FP↑, as TP↑

Discussion – Disruption prediction and uncertainty computation

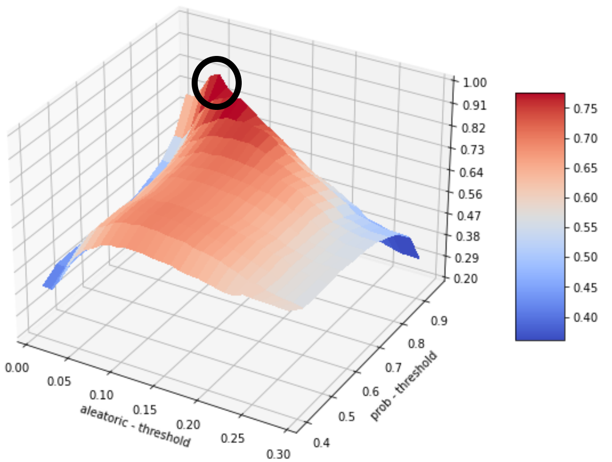
- **Improvement by fine-tuning thresholds of models**
  - **Fine-tuning thresholds of model output and aleatoric uncertainty to maximize F1 score**
    - Setting: evaluation on test dataset + Bayesian Dilated TCN + TQ 40 ms
    - Finding the optimal thresholds for model output and aleatoric uncertainty → Increase of F1, Precision, and Recall

Au-threshold

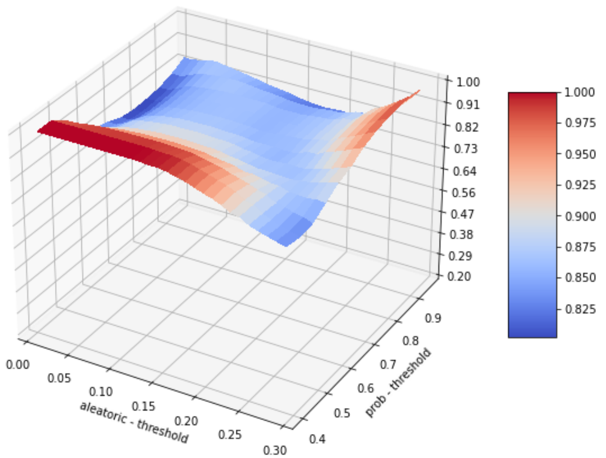


Model output > threshold: disruptive → **Model output > threshold & aleatoric uncertainty < au-threshold: disruptive**
Model output <= threshold & aleatoric uncertainty >= au-threshold: disruptive

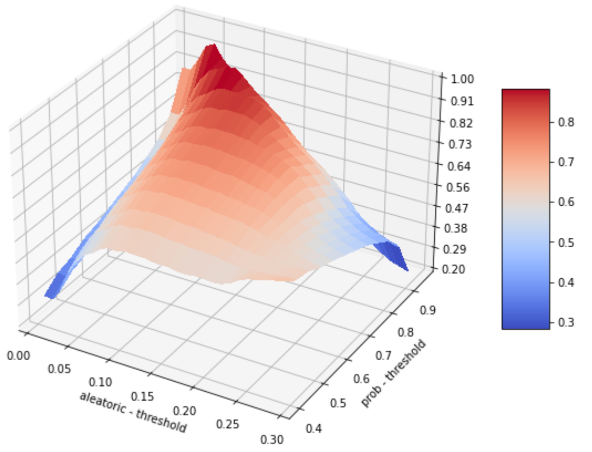
Threshold (output)	Threshold (aleatoric uncertainty)	F1 score	Precision	Recall
0.5	-	0.588	0.837	0.619
0.95	0.05	0.823	0.873	0.936



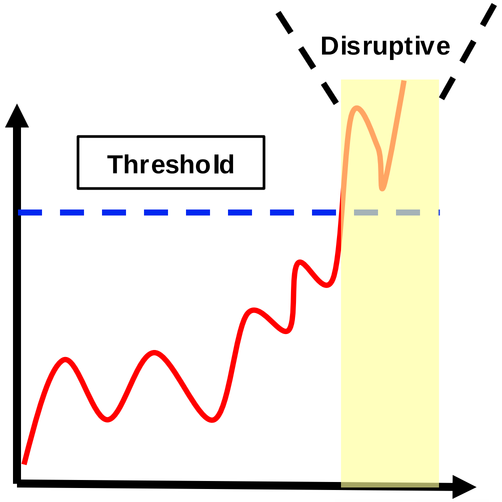
F1 score



Precision

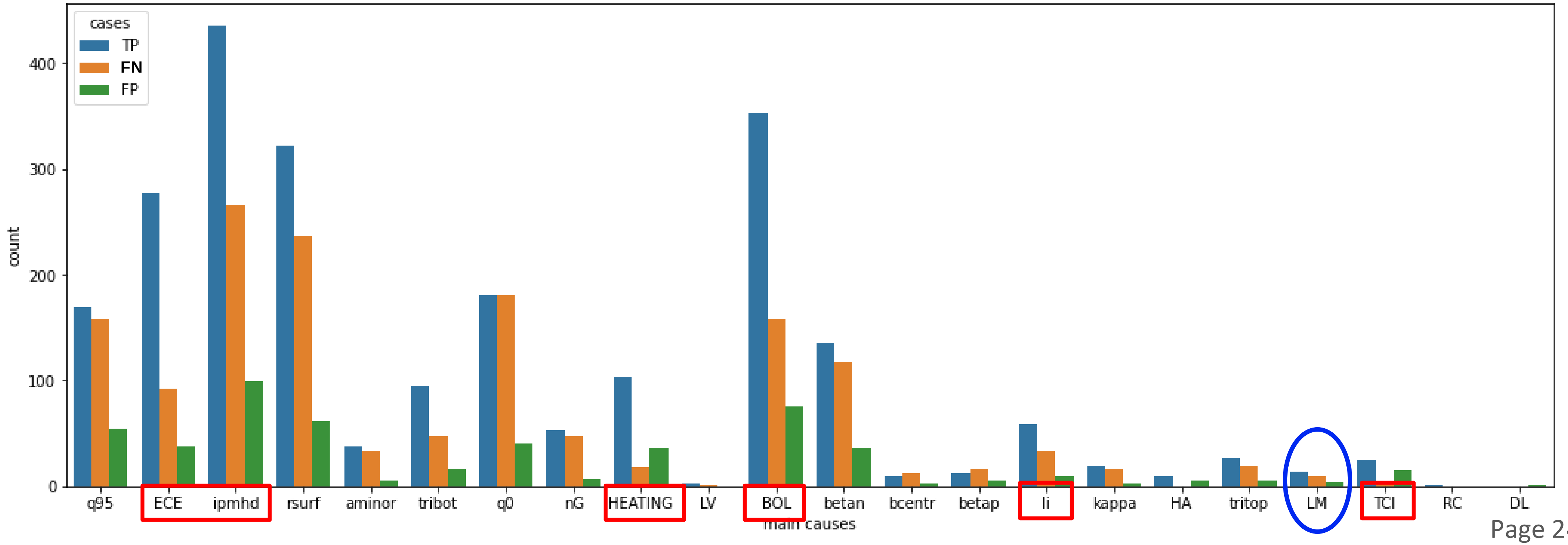


Recall



Discussion – Feature importance and main signals for predictions

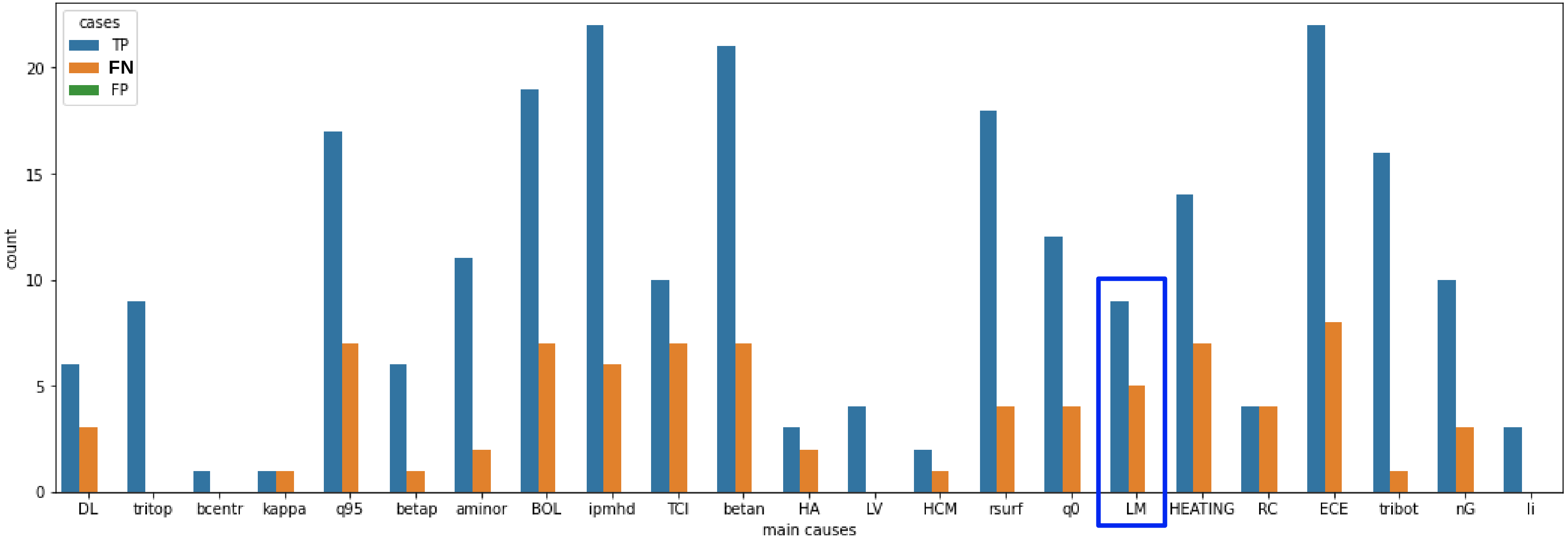
- **Analysis of feature importance: estimation of main signals for predicting disruptions**
  - **Estimation of main signals related to disruption prediction from integrated gradients for TP, FP, and TN cases**
    - The top 5 main signals with the highest integrated gradients were selected for all predictions.
    - Main signals: **plasma current, Bolometers, Major radius, ECE profiles, q0, q95**
    - Input signals with high feature importance for true alarms and low feature importance for missing alarms are imperative.



PLARE

Discussion – Feature importance and main signals for predictions

- **Case study: Locked mode disruption shot**
  - **Top 5 main signals for the experiments with a special case: Locked mode disruption case**
    - Shot list: 20941, 20945, 20947, 20948, 20949, 20951, 20975, 20977
    - No false alarms observed + Profile information (ECE + q95, q0) and triangularity (top and bottom) important
    - Bayesian Dilated TCN predicts some shots (20830, 20904, 20948, 20949, 20951, 20975, 20977, 20978, 20980, ..)
    - Possibility of estimating the indirect causes of disruptions



Conclusion

- Not only current quench, but thermal quench can now be predicted via multiple diagnostic signals and Dilated TCN.
- Bayesian neural networks can provide aleatoric and epistemic uncertainty that enhance models' precisions: False alarm rates can decrease with a rule-based approach utilizing uncertainties.
- Direct feature importance computation with integrated gradients allows the model to detect the indirect causes of disruptions within predicting disruptions.
- Analysis of causes of disruptions estimated by Bayesian models with specific experiments will be conducted.

  → An arose question: Can the Bayesian model map the relation between the causes (signals) and precursors of disruptions?

Shot														Factors
20826	2018	4	14.595	14.5664	14.5655	14.5664		0	0	8	1	0	0	500 MHD
20830	2018	3.999	14.574	14.5495	14.5524	14.5524		0	0	8	1	1	1	500 MHD
20897	2018	1.5	5.098	5.072	5.0732	5.0732		0	0	8	1	0	0	700 ELM
20899	2018	1.5	5.286	5.2581	5.2593	5.2593		0	0	8	1	1	1	700 NC
20900	2018	1.5	4.785	4.7571	4.7586	4.7586		0	0	8	1	0	0	700 NC
20904	2018	3.7	5.738	5.7141	5.7176	5.7176		0	0	8	1	1	1	500 NC
20925	2018	3.7	8.536	8.5119	8.5129	8.5129		0	0	8	1	0	0	500 NC
20938	2018	1.9	8.773	8.7432	8.7441	8.7441		1	0	8	1	1	1	600 NC
20941	2018	2.997	5.789	5.7637	5.7339	5.7637		1	0	8	1	0	0	539 LM3D
20945	2018	2.997	7.955	7.9328	7.934	7.934		1	0	8	1	0	0	540 LM3D
20947	2018	2.997	7.035	7.011	7.0013	7.011		1	0	8	1	1	1	550 LM3D
20948	2018	2.993	10.606	10.5841	10.5816	10.5816		1	0	8	1	1	1	519 LM3D
20949	2018	4.995	7.876	7.8472	7.846	7.8472		1	0	8	1	1	1	560 LM3D
20951	2018	2.993	9.675	9.6496	9.6461	9.6461		1	0	8	1	1	1	519 LM3D
20975	2018	2	7.351	7.316	7.3165	7.3165		1	0	0	1	0	0	850 LM3D
20977	2018	2	8.971	8.9341	8.9354	8.9354	8.9015	1	0	8	1	1	1	850 LM3D
20978	2018	2	8.625	8.5896	8.5892	8.5896	8.5608	0	0	8	1	0	0	850 MHD
20980	2018	2	8.62	8.5842	8.5852	8.5852	7.4189	0	0	8	1	1	1	850 MHD
21031	2018	1.497	6.941	6.9235	6.9282	6.9235	6.2842	0	0	8	1	0	0	519 ETC
21033	2018	1.5	5.998	5.9788	5.9812	5.9812	5.6742	0	0	8	1	0	0	500 NC

Thank You

Appendix – Bayesian Neural Network

▪ Bayesian Neural Network

- Conventional approach (Frequentist view): weights of the neural networks are trained by maximum likelihood estimation (MLE)
- Weights as random variables: Finding the optimal weights = Maximum a posteriori (MAP) weights

$$W^{MLE} = argmax_W \log P(D|W) \qquad W^{MAP} = argmax_W \log P(W|D) = argmax_W \log P(D|W) + logP(W)$$

- Bayesian by Backpropagation (Charles Blundell et al, 2015)

$$\theta^* = argmin_\theta KL[q(w|\theta)||P(w)] - E_q[logP(D|W)] = argmin_\theta F(D,\theta)$$

- Variational inference: intractable in general cases, but variational approximation by MC sampling can reduce the computational cost and handle intractability.

$$F(D,\theta) = \sum \log q(w|\theta) - logP(w) - logP(D|w)$$

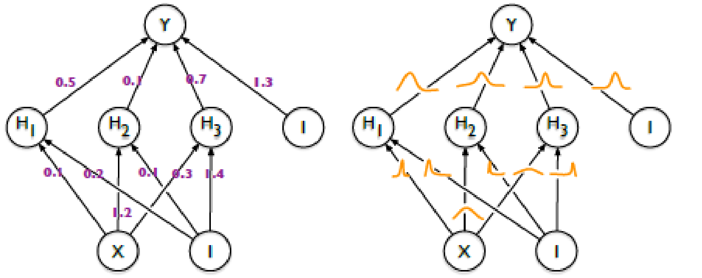
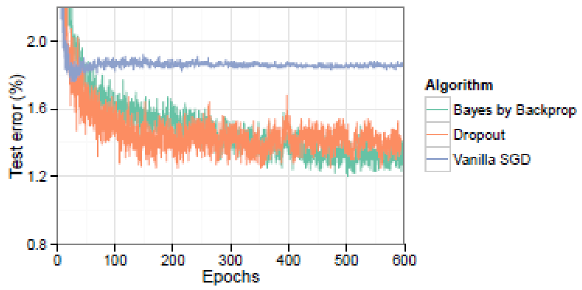
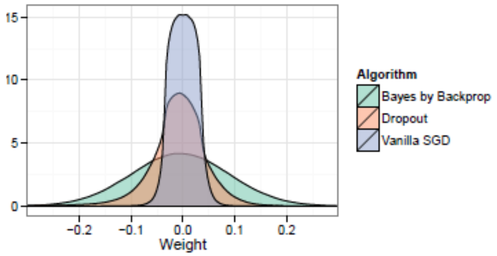


Figure 1. Left: each weight has a fixed value, as provided by classical backpropagation. Right: each weight is assigned a distribution, as provided by Bayes by Backprop.

Charles Blundell et al, 2015



Test error on MNIST as training progresses, Charles Blundell et al, 2015



Trained weights of the neural networks, Charles Blundell et al, 2015

Gaussian variational posterior

1. Sample $\epsilon \sim \mathcal{N}(0, I)$.
2. Let $\mathrm{w} = \mu + \log(1 + \exp(\rho)) \circ \epsilon$.
3. Let $\theta = (\mu, \rho)$.
4. Let $f(\mathrm{w}, \theta) = \log q(\mathrm{w}|\theta) - \log P(\mathrm{w})P(\mathcal{D}|\mathrm{w})$.
5. Calculate the gradient with respect to the mean

$$\Delta_\mu = \frac{\partial f(\mathrm{w}, \theta)}{\partial \mathrm{w}} + \frac{\partial f(\mathrm{w}, \theta)}{\partial \mu}. \tag{3}$$

6. Calculate the gradient with respect to the standard deviation parameter ρ

$$\Delta_\rho = \frac{\partial f(\mathrm{w}, \theta)}{\partial \mathrm{w}} \frac{\epsilon}{1 + \exp(-\rho)} + \frac{\partial f(\mathrm{w}, \theta)}{\partial \rho}. \tag{4}$$

7. Update the variational parameters:

$$\mu \leftarrow \mu - \alpha \Delta_\mu \tag{5}$$

$$\rho \leftarrow \rho - \alpha \Delta_\rho. \tag{6}$$

$$P(\mathrm{w}) = \prod_j \pi \mathcal{N}(\mathrm{w}_j | 0, \sigma_1^2) + (1 - \pi) \mathcal{N}(\mathrm{w}_j | 0, \sigma_2^2), \tag{7}$$

Appendix - uncertainties

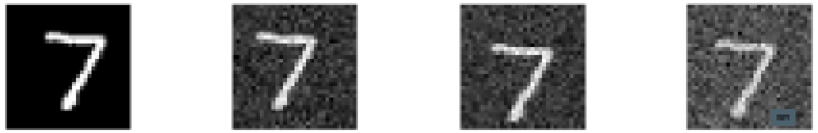
- ## Computation of the uncertainty

  - **Aleatoric uncertainty vs Epistemic uncertainty**
  - **Aleatoric uncertainty**: uncertainty induced by the data noise, due to the random nature of the physical systems
  - **Epistemic uncertainty**: uncertainty induced by model weights, related to the probabilistic distribution of the model weights, due to the lack of knowledge of the systems (a low generalization of the model)

$$Var_q[P(y^*|x^*)] = E_q\left[y^* y^{*T}\right] - E_q[y^*]E_q[y^*]^T = \underbrace{\boxed{\int [diag\left[E_p[y^*]\right] - E_p[y^*]E_p[y^*]^T q_\theta(w|D)dw}}_{\text{Aleatoric uncertainty}} + \underbrace{\boxed{\int \left[E_p[y^*] - E_q[y^*]\right]\left[E_p[y^*] - E_q[y^*]\right]^T q_\theta[w|D]dw}}_{\text{Epistemic uncertainty}}$$

  - Aleatoric uncertainty decreases with the increase of dataset, however epistemic uncertainty requests to refine the model
  - We can calculate epistemic uncertainty from the Bayesian approach, thus we can get more accurate and reliable disruption prediction with considering Epistemic uncertainty.

$\gamma =$	0	0.1	0.2	0.3
aleatoric $=$	0.00099897	0.00099885	0.00099875	0.00099838



Aleatoric uncertainty computed by Bayesian VGG on MNIST dataset, Kumar Shridhar et al, 2019

$$\mathrm{Var}_q\big(p(y^*|x^*)\big) = \underbrace{\frac{1}{T}\sum_{t=1}^{T} \mathrm{diag}(\hat{p}_t) - \hat{p}_t\,\hat{p}_t^T}_{\text{aleatoric}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}(\hat{p}_t - \bar{p})(\hat{p}_t - \bar{p})^T}_{\text{epistemic}}$$

where $\bar{p} = \frac{1}{T}\sum_{t=1}^{T} \hat{p}_t$ and $\hat{p}_t = \mathrm{Sofplus}_n\big(f_{w_t}(x^*)\big)$.

Simple computation of aleatoric uncertainty and epistemic uncertainty, Kumar Shridhar et al, 2019