DEEL

DEpendable & Explainable Learning

# Hands-on Conformal Prediction with PUNCC

Mouhcine MENDIL

IVADO

IRT SAINT EXUPÉRY

CRIAQ

Consortium de recherche et d'innovation en aérospatiale au Québec

ANITI

Porté par l'Université de Toulouse

Institut intelligence et données

# Hands-on Conformal Prediction with **PUNCC**

DEEL PUNCC

A Conformal Prediction Library

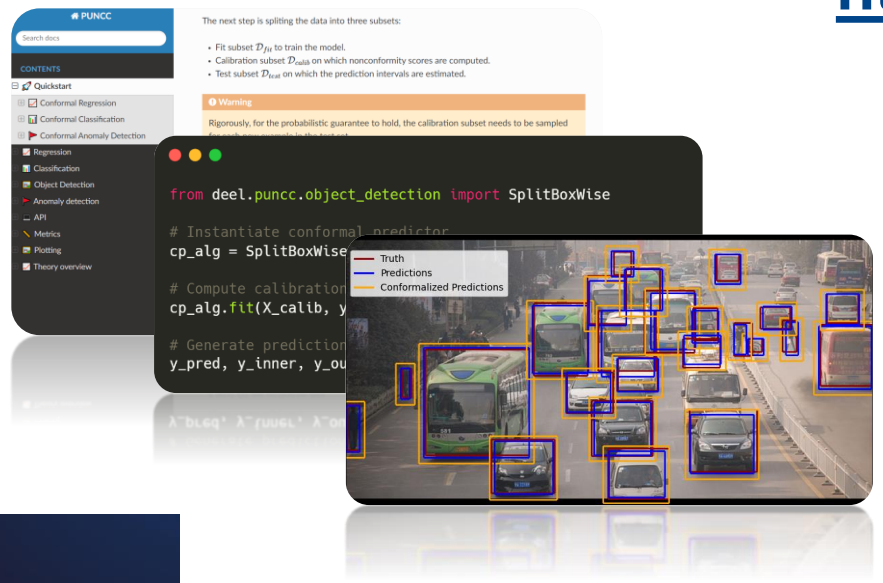
Python 3.8 + | License MIT | pylint passing | tox passing

***Puncc*** (short for Predictive **u**ncertainty **c**alibration and **c**onformalization) is an open-source Python library. It seamlessly integrates a collection of state-of-the-art conformal prediction algorithms and associated techniques for diverse machine learning tasks, including regression, classification and anomaly detection. ***Puncc*** can be used with any predictive model to provide rigorous uncertainty estimations. Under data exchangeability (or *i.i.d*), the generated prediction sets are guaranteed to cover the true outputs within a user-defined error $\alpha$.

Documentation is available online.

## Table of contents






https://github.com/deel-ai/puncc

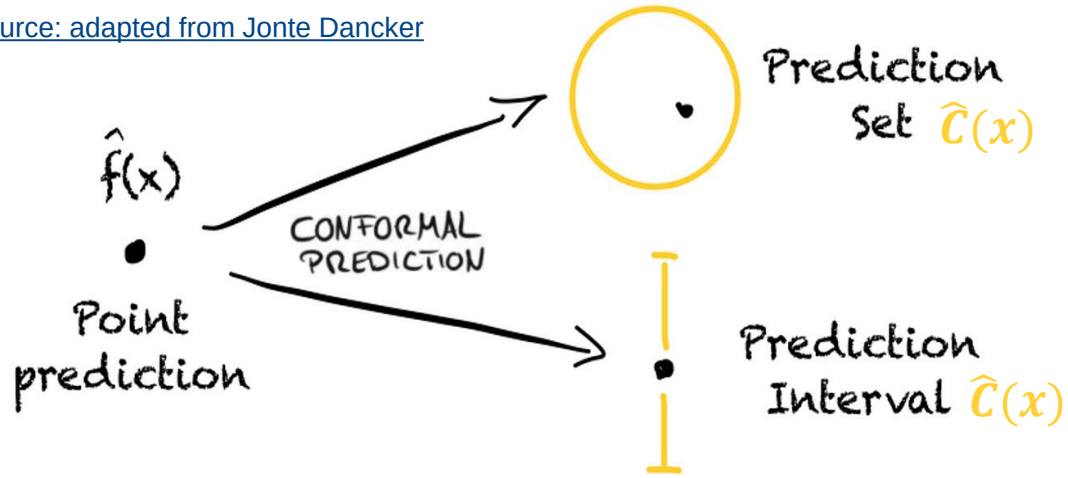
Proceedings of Machine Learning Research 204:1–20, 2023 Conformal and Probabilistic Prediction with Applications

**PUNCC: a Python Library for Predictive Uncertainty Calibration and Conformalization**


**Mouhcine Mendil** MOUHCINE.MENDIL@IRT-SAINTEXUPERY.COM
*IRT Saint Exupéry, Toulouse, France*
**Luca Mossina** LUCA.MOSSINA@IRT-SAINTEXUPERY.COM
*IRT Saint Exupéry, Toulouse, France*
**David Vigouroux** DAVID.VIGOUROUX@IRT-SAINTEXUPERY.COM
*IRT Saint Exupéry, Toulouse, France*

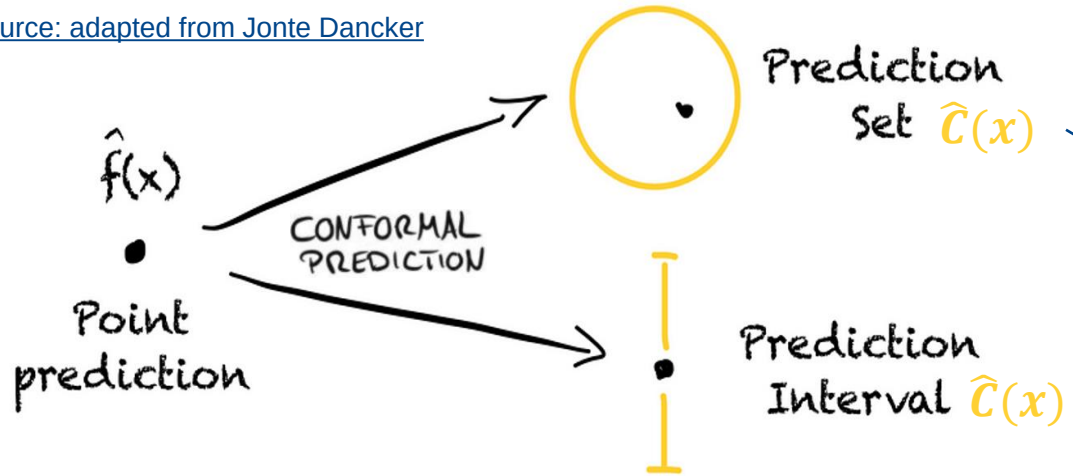
# Conformal Prediction: Recap

Source: adapted from Jonte Dancker



# Conformal Prediction: Recap

Source: adapted from Jonte Dancker



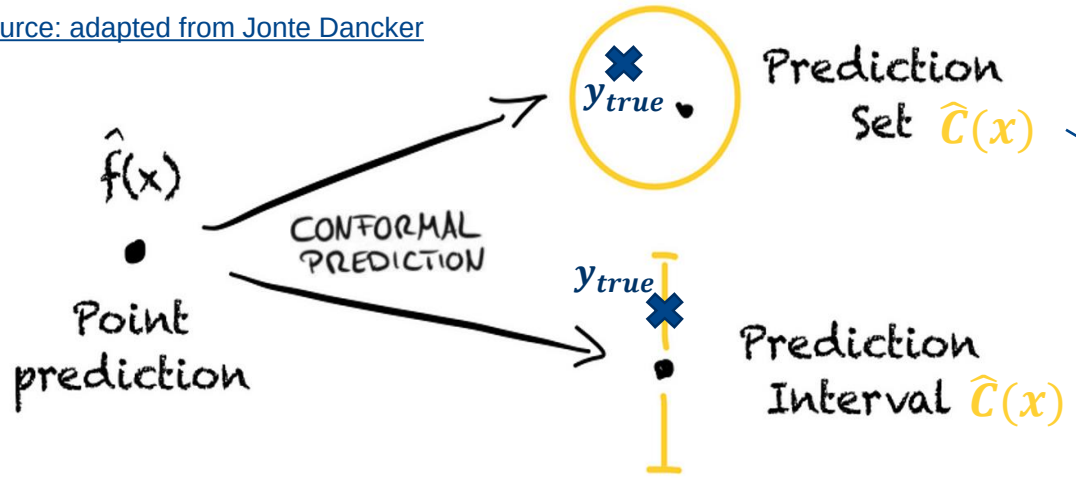

$$\hat{C}(x) = \{y \in \mathcal{Y}, \quad s(x, y) \leq \delta_\alpha\}$$

Nonconformity score

$\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$ quantile of calibration scores

# Conformal Prediction: Recap

Source: adapted from Jonte Dancker




Nonconformity score

$$\hat{C}(x) = \{y \in \mathcal{Y}, \quad s(x, y) \leq \delta_\alpha\}$$

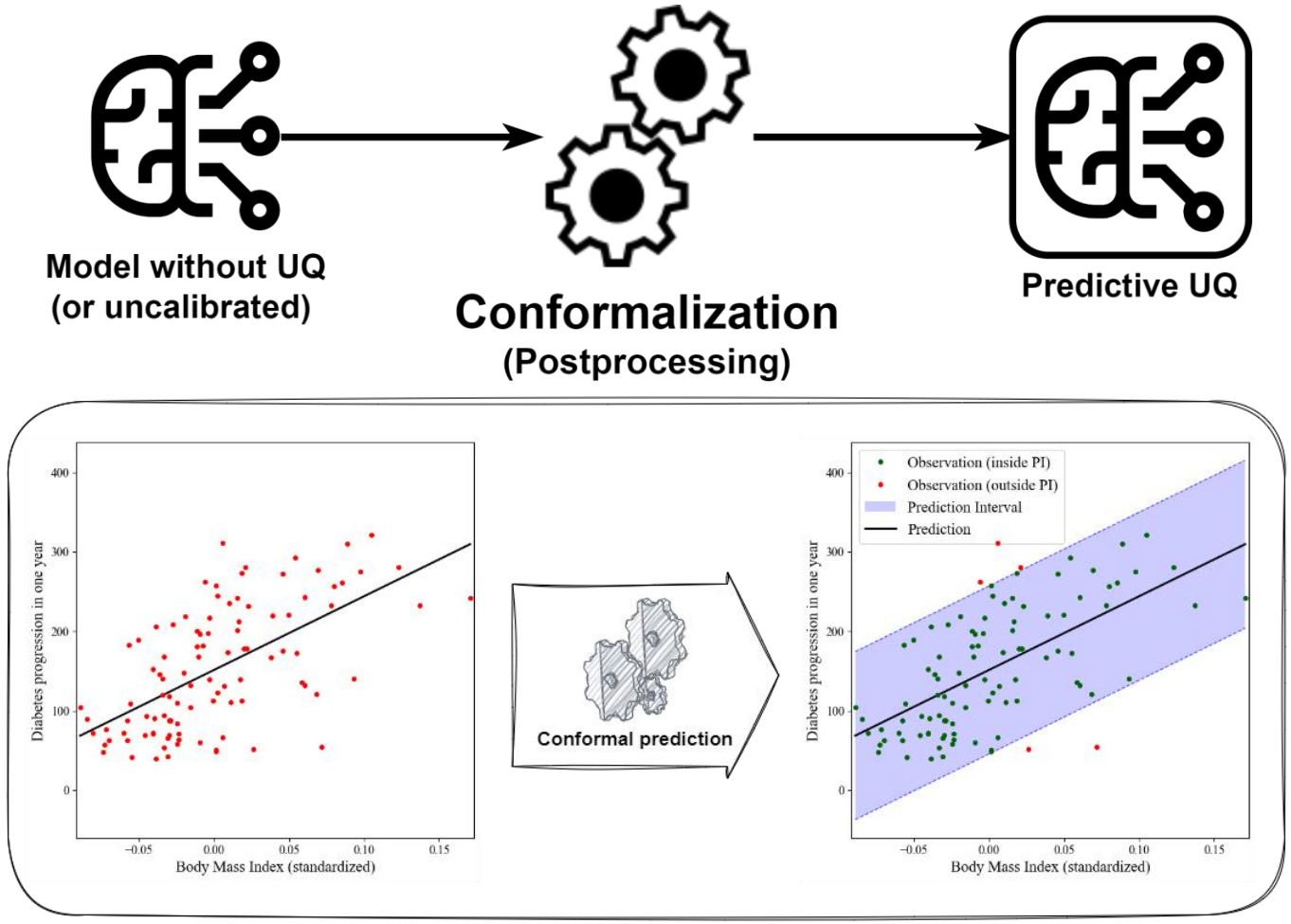
$\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$ quantile of calibration scores

User-defined error-rate

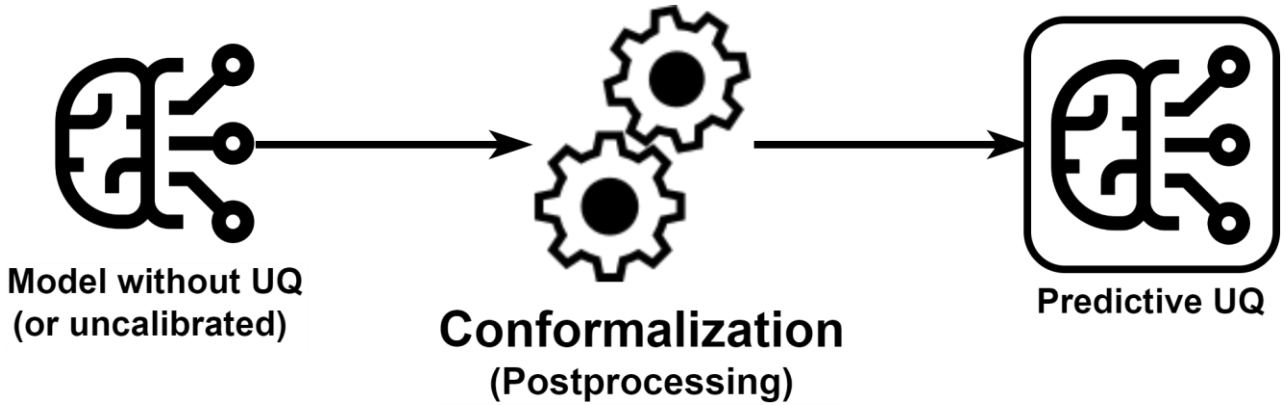
$$P\left(y_{true} \in \hat{C}(x)\right) \geq 1 - \alpha$$
$$\Leftrightarrow P\left(y_{true} \notin \hat{C}(x)\right) < \alpha$$

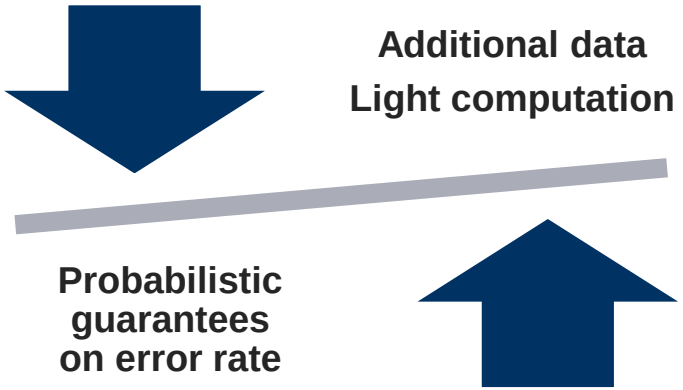
If $\mathcal{D}_{calib}$ and $(x, y_{true})$ are i.i.d

# Conformal Prediction: Procedure



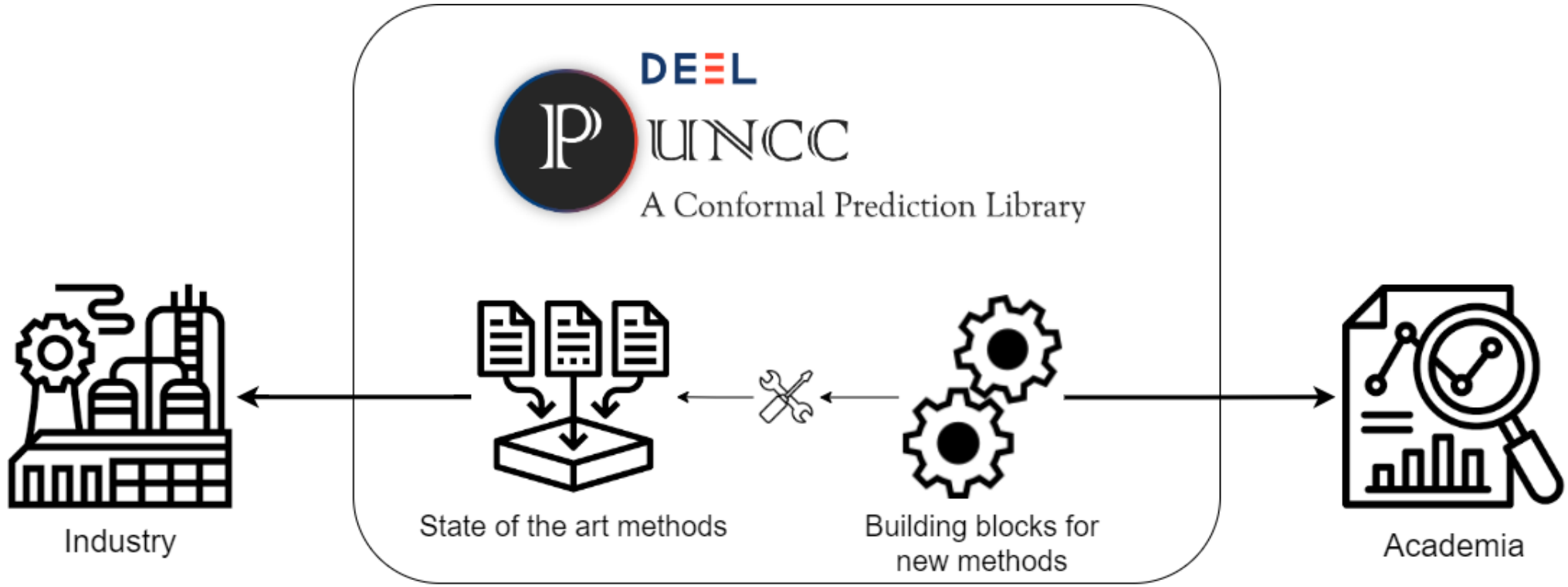
# Conformal Prediction: Procedure



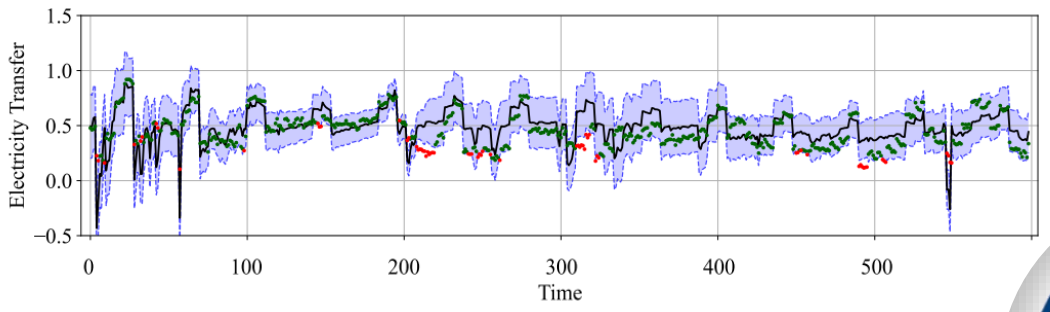



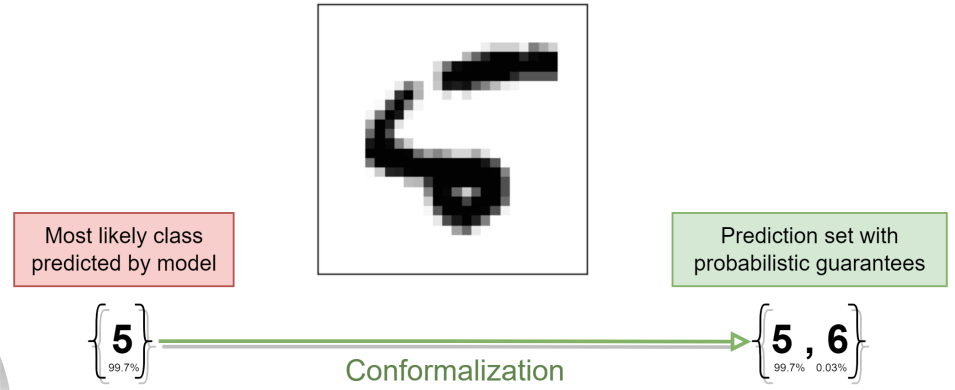
# PUNCC Library (Predictive UNcertainty Calibration and Conformalization)

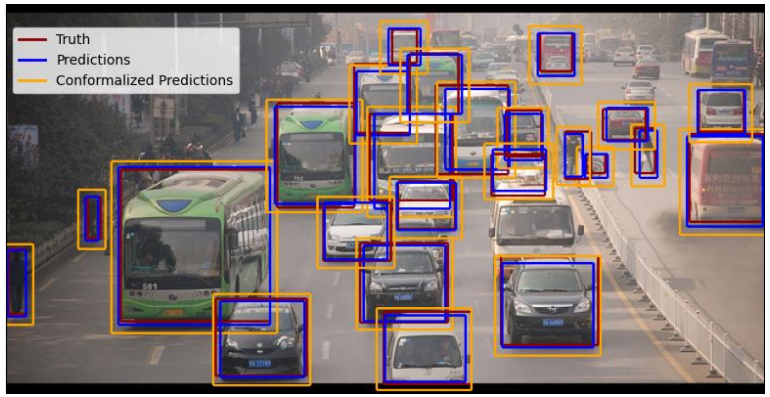
DEEL
DEpendable & Explainable Learning

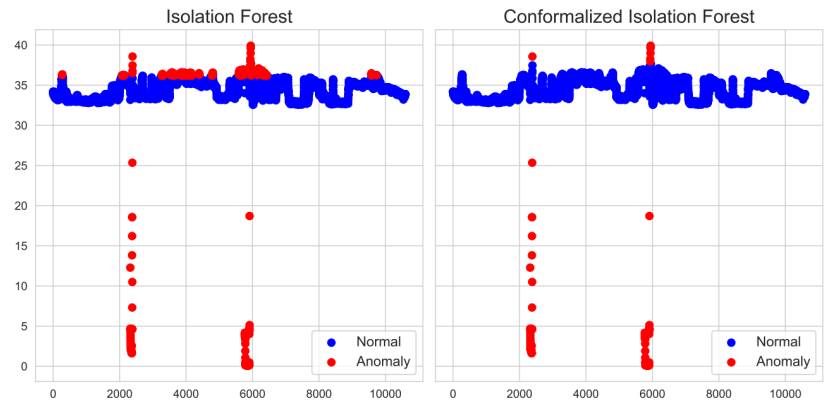


# PUNCC for different ML Tasks

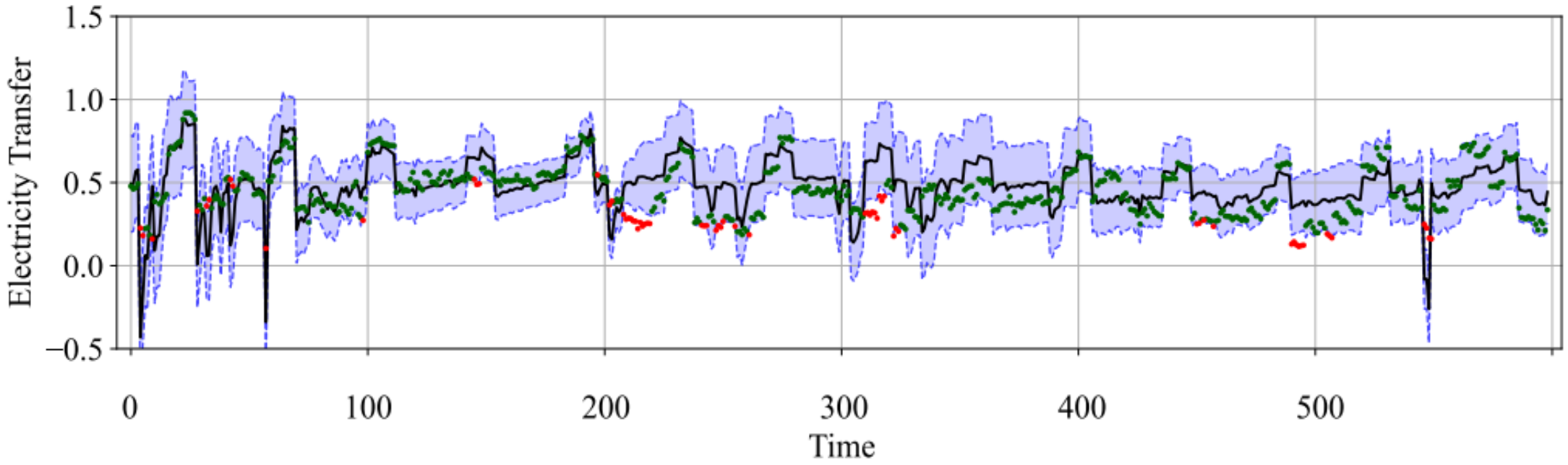


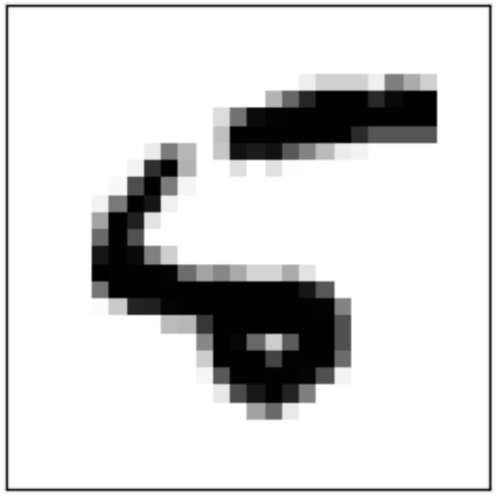



# PUNCC: Regression



# PUNCC: Classification



Most likely class predicted by model

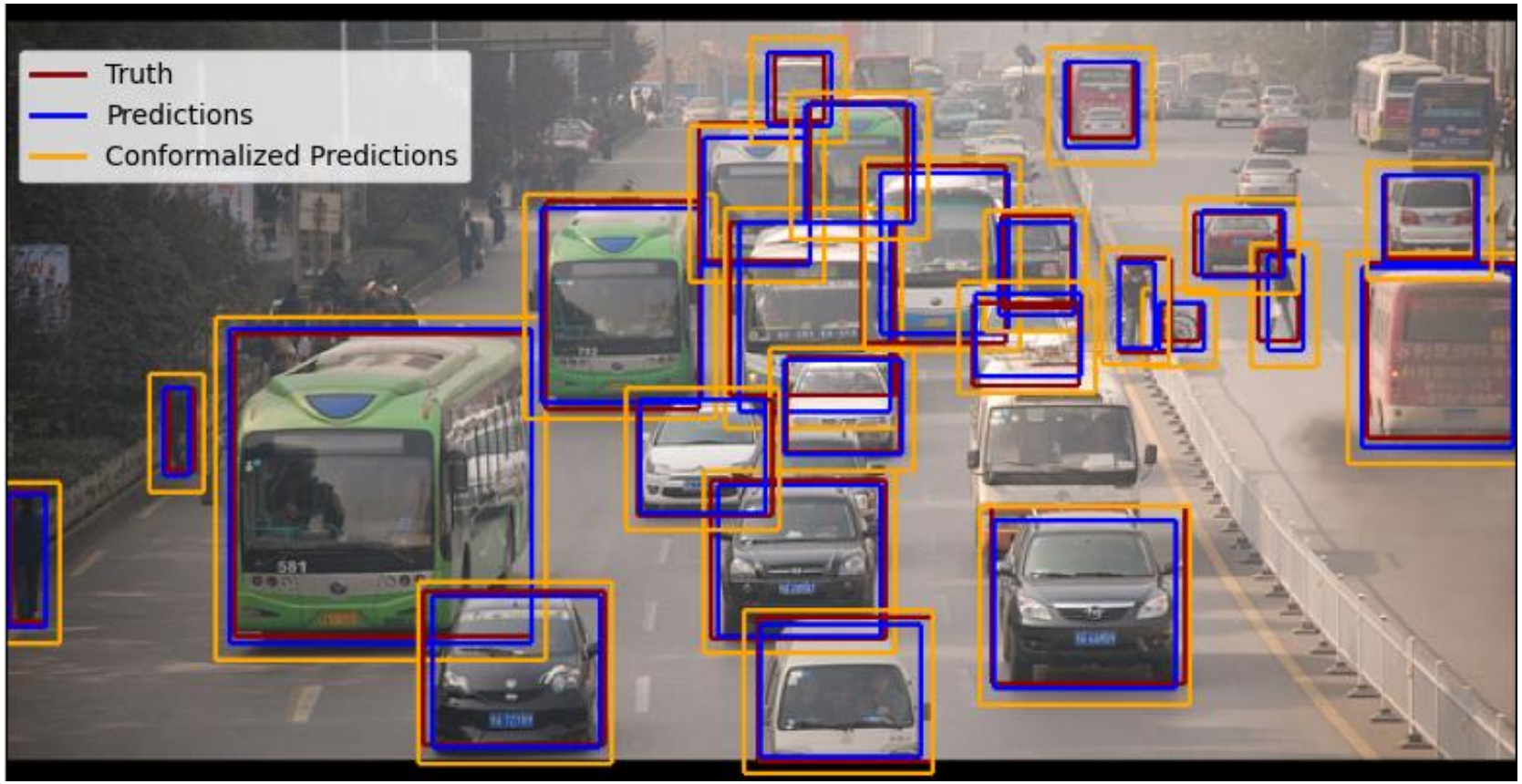
{ 5 } 99.7%

Conformalization

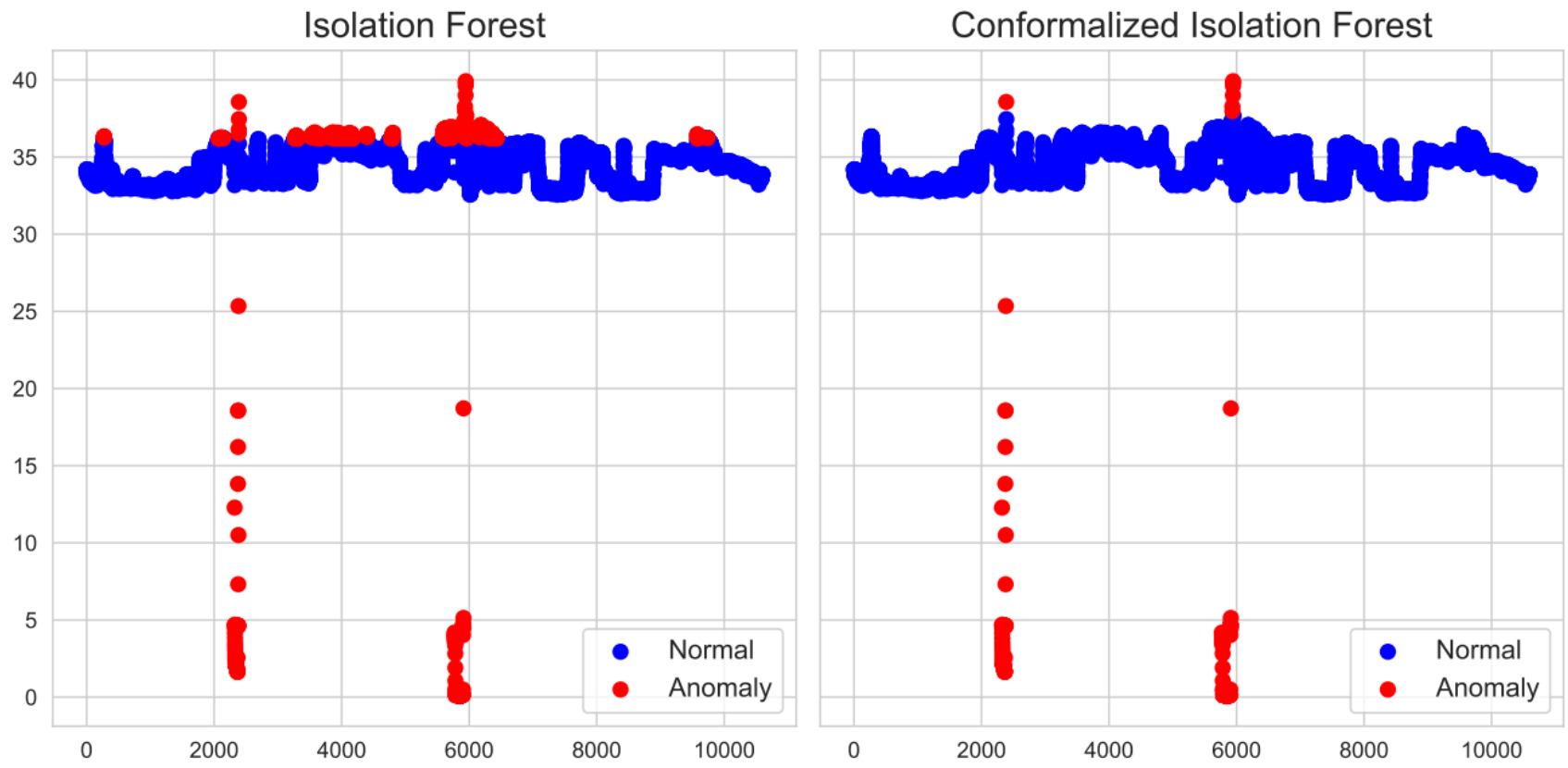
Prediction set with probabilistic guarantees

{ 5 , 6 } 99.7% 0.03%

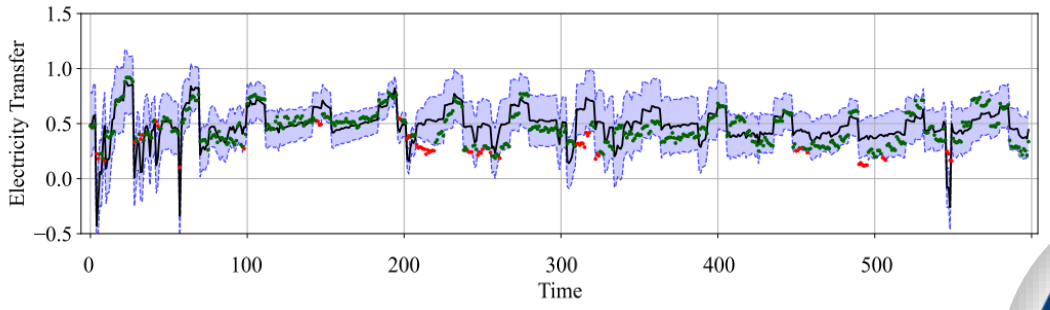
# PUNCC: Object Detection

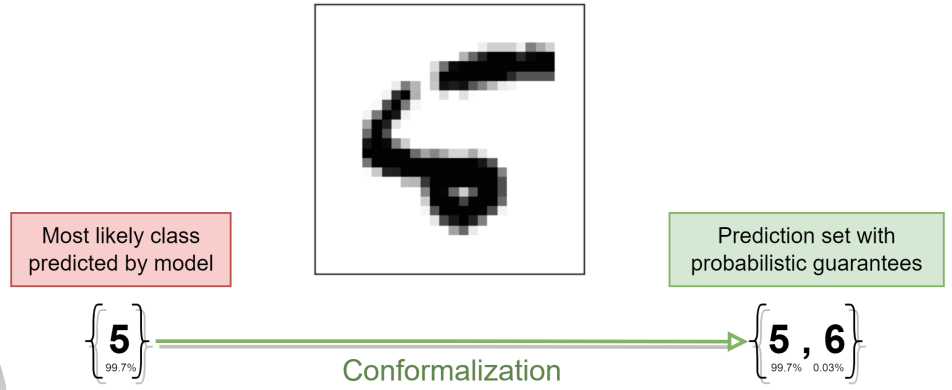


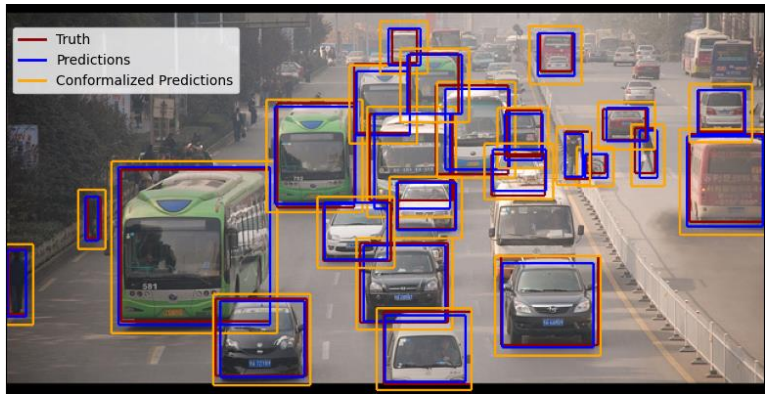
# PUNCC: Outlier Detection

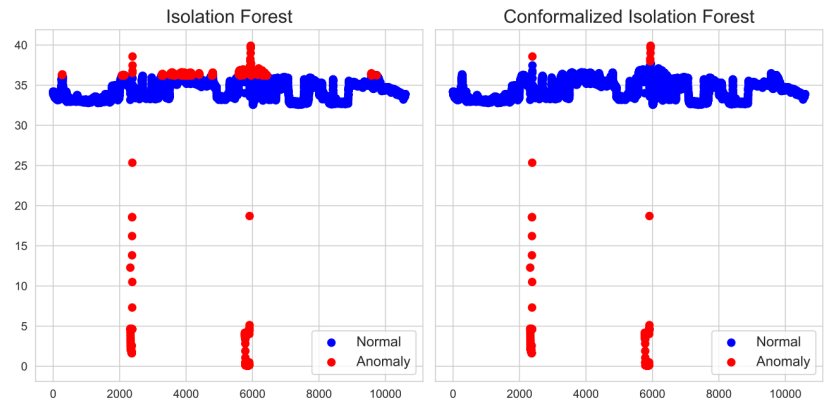


# PUNCC for different ML Tasks



# Conformal Prediction in few lines of code

DEEL
DEpendable & Explainable Learning

## Conformal Regression

```python
from deel.puncc.regression import SplitCP

# Instantiate conformal predictor
cp_alg = SplitCP(Predictor)

# Compute calibration scores
cp_alg.fit(X_calib, y_calib)

# Generate prediction sets
y_pred, y_low, y_high = cp_alg.predict(X_new, alpha=0.1)
```

# Conformal Prediction in few lines of code

DEEL
DEpendable & Explainable Learning

## Conformal Classification

```python
from deel.puncc.classification import APS

# Instantiate conformal predictor
cp_alg = APS(Predictor)

# Compute calibration scores
cp_alg.fit(X_calib, y_calib)

# Generate prediction sets
y_pred, set_pred = cp_alg.predict(X_new, alpha=0.1)
```

# Conformal Prediction in few lines of code

DEEL
DEpendable & Explainable Learning

## Conformal Object Detection

```python
from deel.puncc.object_detection import SplitBoxWise

# Instantiate conformal predictor
cp_alg = SplitBoxWise(Predictor)

# Compute calibration scores
cp_alg.fit(X_calib, y_calib)

# Generate prediction sets
y_pred, y_inner, y_outer = cp_alg.predict(X_new, alpha=0.1)
```

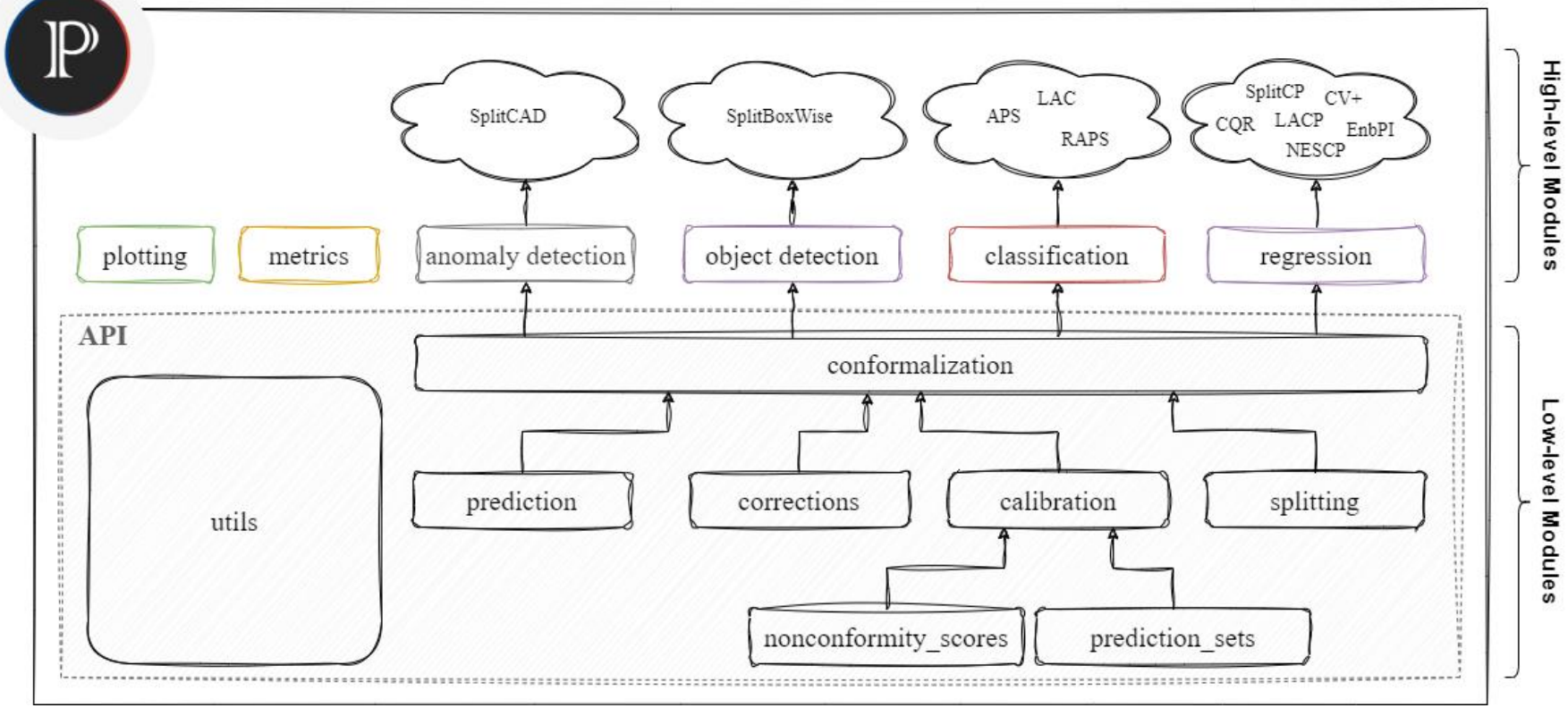
# Interoperability

PUNCC supports popular data types and ML libraries and more …


```python
from deel.puncc.api.prediction import BasePredictor
from deel.puncc.classification import APS

# My scikit-learn classifier
sklearn_classifier_model = ...

# Redefine the predict method of your classifier
def MyPredictor(BasePredictor):
  def predict(X):
    return self.model.predict_proba(X)

# Wrap scikit-learn classifier to interoperate with puncc
predictor = MyPredictor(sklearn_classifier_model)

# Instantiate the model
cp_alg = APS(predictor)
```

Can work on top of UQ models and libraries

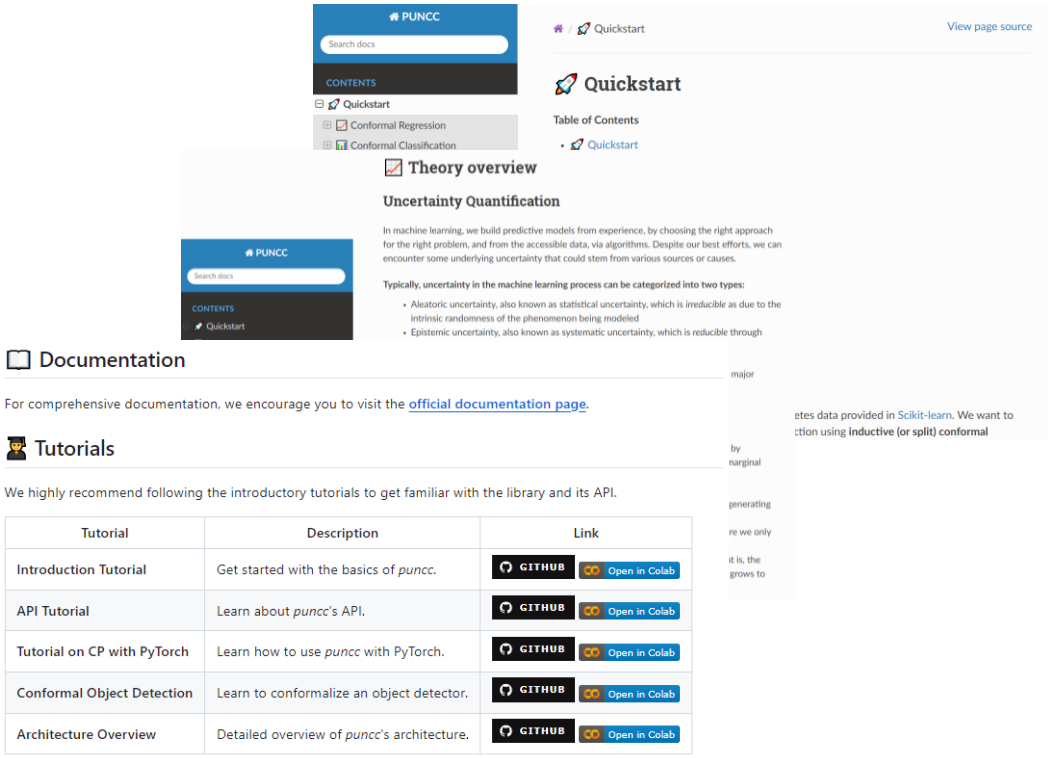
# More Flexibility with the API




**Low-level API**: full customization of CP algorithm (nonconformity scores, cross-validation, ….)

# User Experience

- ✓ Ease of use
- ✓ Documentation and tutoriels
- ✓ Scientific rigor
- ✓ Introduction to the theory to help with method selection
- ✓ Tests, CI/CD, ...
- ✓ New features, updates, patches, …
- ✓ Open to contributions: issues, PRs, …



DEEL

DEpendable & Explainable Learning

# Hands-on Session

https://github.com/deel-ai/uq-masterclass

IVADO

IRT SAINT EXUPÉRY

CRIAQ

Consortium de recherche et d'innovation en aérospatiale au Québec

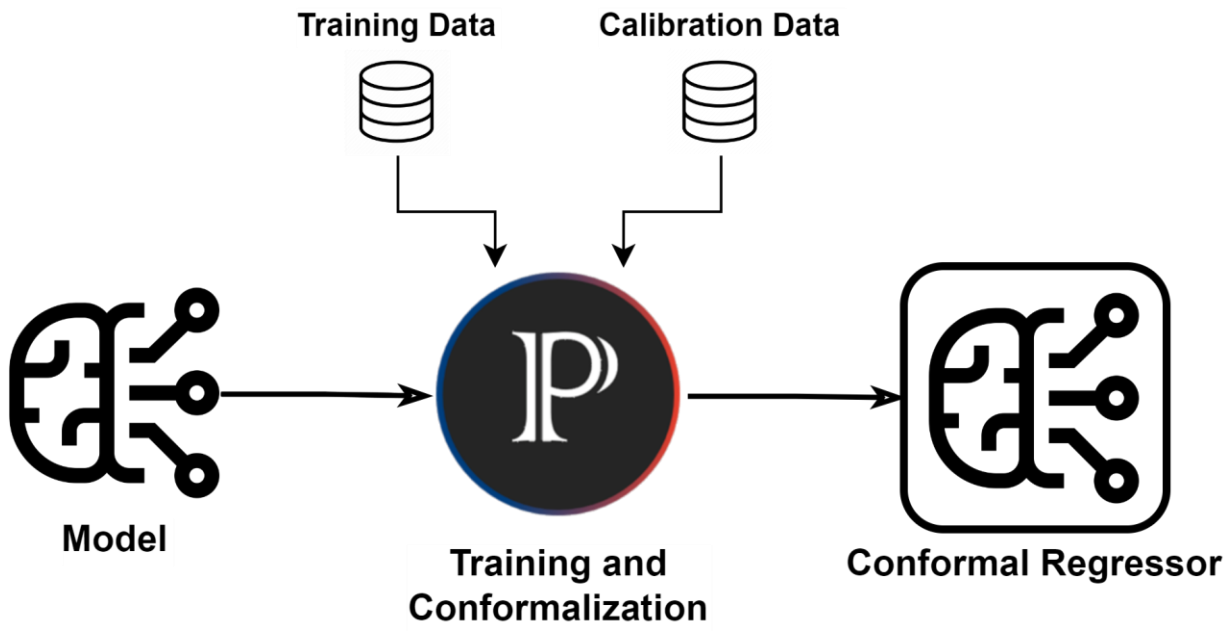
ANITI

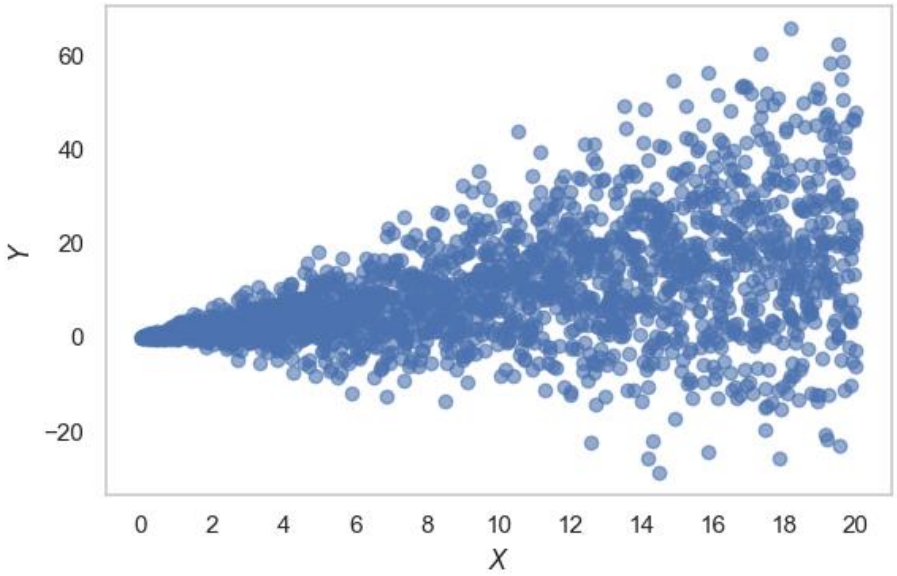
Porté par l'Université de Toulouse

Institut intelligence et données

# Tutorial on Conformal Regression

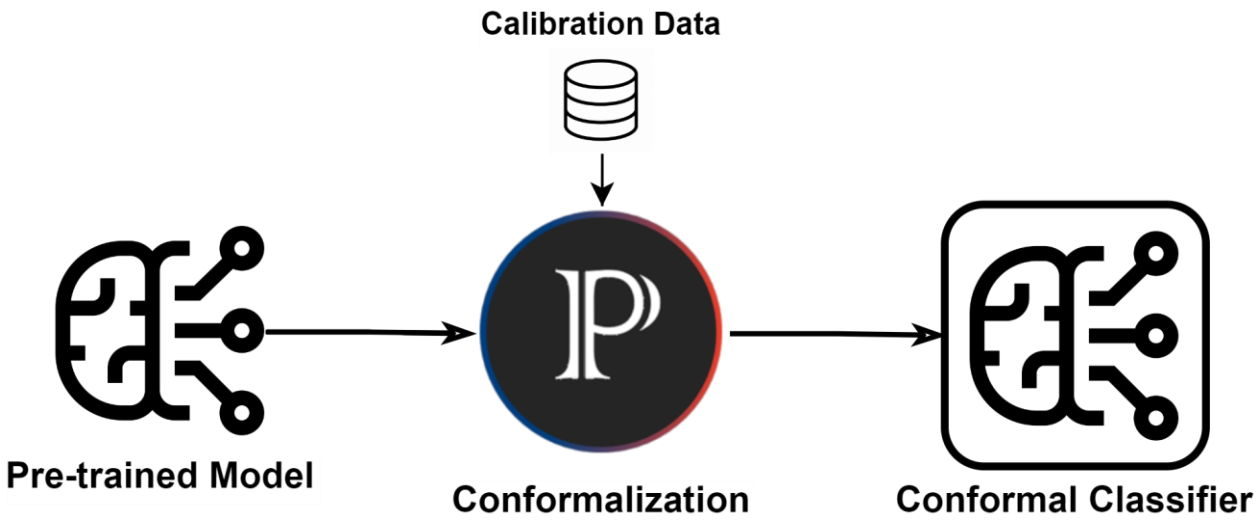
- Model to be selected and trained from scratch

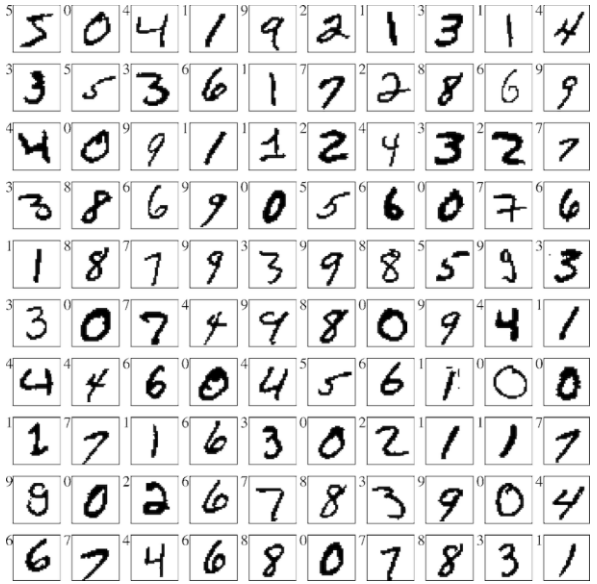


# Tutorial on Conformal Classification

- Pretrained classifier within existing ML pipeline

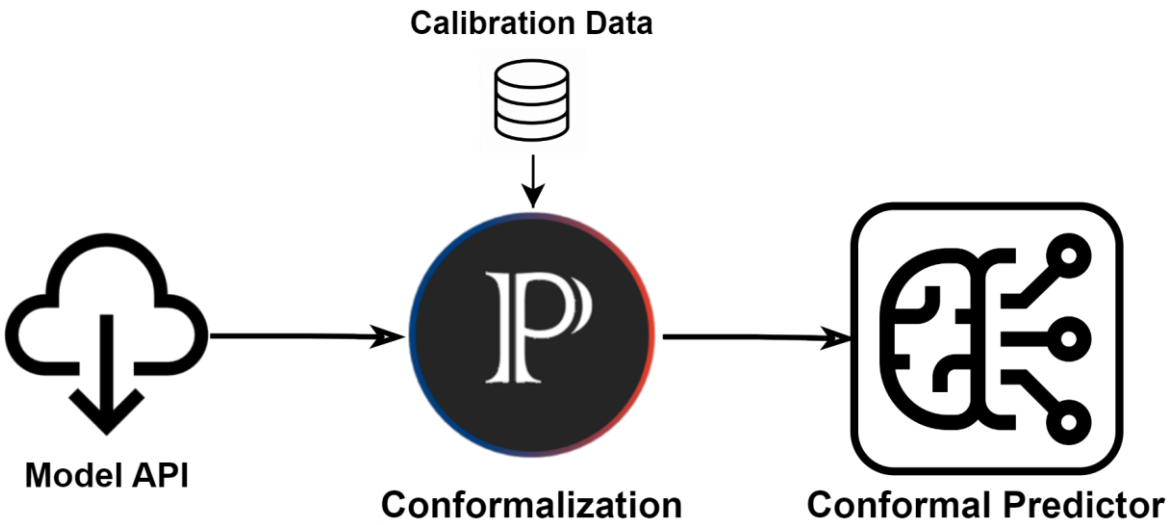


Source: D. Decoste

# Tutorial on Conformal Object Detection

- Predictions accessible via API





https://cocodataset.org/

DEEL
DEpendable & Explainable Learning

DEEL
# PUNCC
A Conformal Prediction Library

NumPy pandas scikit-learn PyTorch XGBoost

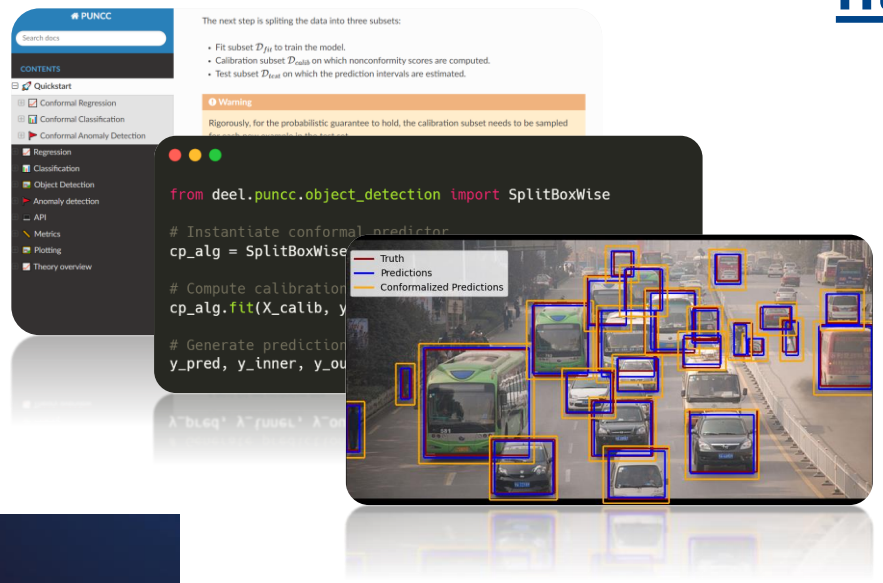
Python 3.8 + | License MIT | pylint passing | tox passing

***Puncc*** (short for Predictive **u**ncertainty **c**alibration and **c**onformalization) is an open-source Python library. It seamlessly integrates a collection of state-of-the-art conformal prediction algorithms and associated techniques for diverse machine learning tasks, including regression, classification and anomaly detection. ***Puncc*** can be used with any predictive model to provide rigorous uncertainty estimations. Under data exchangeability (or *i.i.d*), the generated prediction sets are guaranteed to cover the true outputs within a user-defined error $\alpha$.

Documentation is available online.

## Table of contents






https://github.com/deel-ai/puncc

Proceedings of Machine Learning Research 204:1–20, 2023 Conformal and Probabilistic Prediction with Applications

## PUNCC: a Python Library for Predictive Uncertainty Calibration and Conformalization


**Mouhcine Mendil** MOUHCINE.MENDIL@IRT-SAINTEXUPERY.COM
*IRT Saint Exupéry, Toulouse, France*
**Luca Mossina** LUCA.MOSSINA@IRT-SAINTEXUPERY.COM
*IRT Saint Exupéry, Toulouse, France*
**David Vigouroux** DAVID.VIGOUROUX@IRT-SAINTEXUPERY.COM
*IRT Saint Exupéry, Toulouse, France*

DEEL
France-Quebec

# Thanks for your attention !

IVADO

IRT SAINT EXUPÉRY

CRIAQ
Consortium de recherche et d'innovation en aérospatiale au Québec

ANITI
Porté par l'Université de Toulouse

Institut intelligence et données