

Package ‘criticality’

July 22, 2025

Title Modeling Fissile Material Operations in Nuclear Facilities
Version 0.9.3
Depends R (>= 3.6.0), caret
Description A collection of functions for modeling fissile material operations in nuclear facilities, based on Zywiec et al (2021) <[doi:10.1016/j.j.ress.2020.107322](https://doi.org/10.1016/j.j.ress.2020.107322)>.
License MIT + file LICENSE
Encoding UTF-8
RoxygenNote 7.2.3
Imports bnlearn, dplyr, evd, fitdistrplus, ggplot2, keras, magrittr, parallel, reticulate, scales
NeedsCompilation no
Author William Zywiec [aut, cre] (ORCID: <<https://orcid.org/0000-0002-1842-9599>>)
Maintainer William Zywiec <willzywiec@gmail.com>
Repository CRAN
Date/Publication 2023-05-29 22:20:14 UTC

Contents

BN	2
Model	2
NN	3
Plot	5
Risk	5
Sample	7
Scale	8
Tabulate	9
Test	9
Index	11

BN	<i>BN Function</i>
----	--------------------

Description

This function creates a Bayesian network from pre-formatted nuclear facility data.

Usage

```
BN(dist = "gamma", facility.data, ext.dir)
```

Arguments

- dist Truncated probability distribution (e.g., "gamma", "normal")
- facility.data .csv file name
- ext.dir External directory (full path)

Value

A Bayesian network that models fissile material operations (op), controls (ctrl), and parameters that affect nuclear criticality safety

Examples

```
ext.dir <- paste0(tempdir(), "/criticality/extdata")
dir.create(ext.dir, recursive = TRUE, showWarnings = FALSE)

extdata <- paste0(.libPaths()[1], "/criticality/extdata")
file.copy(paste0(extdata, "/facility.csv"), ext.dir, recursive = TRUE)
file.copy(paste0(extdata, "/mcnp-dataset.RData"), ext.dir, recursive = TRUE)

BN(
  facility.data = "facility.csv",
  ext.dir = ext.dir
)
```

Model	<i>Model Function</i>
-------	-----------------------

Description

This function builds the deep neural network metamodel architecture.

Usage

```
Model(
  dataset,
  layers = "8192-256-256-256-256-16",
  loss = "sse",
  opt.alg = "adamax",
  learning.rate = 0.00075,
  ext.dir
)
```

Arguments

dataset	Training and test data
layers	String that defines the deep neural network architecture (e.g., "64-64")
loss	Loss function
opt.alg	Optimization algorithm
learning.rate	Learning rate
ext.dir	External directory (full path)

Value

A deep neural network metamodel of Monte Carlo radiation transport code simulation data

 NN

NN Function

Description

This function trains an ensemble of deep neural networks to predict keff values (imports Tabulate, Scale, Model, Fit, Plot, and Test functions).

Usage

```
NN(
  batch.size = 8192,
  code = "mcnp",
  dataset,
  ensemble.size = 5,
  epochs = 1500,
  layers = "8192-256-256-256-256-16",
  loss = "sse",
  opt.alg = "adamax",
  learning.rate = 0.00075,
  val.split = 0.2,
  overwrite = FALSE,
```

```

    remodel = FALSE,
    replot = TRUE,
    verbose = FALSE,
    ext.dir,
    training.dir = NULL
  )

```

Arguments

<code>batch.size</code>	Batch size
<code>code</code>	Monte Carlo radiation transport code (e.g., "cog", "mcnp")
<code>dataset</code>	Training and test data
<code>ensemble.size</code>	Number of deep neural networks in the ensemble
<code>epochs</code>	Number of training epochs
<code>layers</code>	String that defines the deep neural network architecture (e.g., "64-64")
<code>loss</code>	Loss function
<code>opt.alg</code>	Optimization algorithm
<code>learning.rate</code>	Learning rate
<code>val.split</code>	Validation split
<code>overwrite</code>	Boolean (TRUE/FALSE) that determines if files should be overwritten
<code>remodel</code>	Boolean (TRUE/FALSE) that determines if an existing metamodel should be reused
<code>replot</code>	Boolean (TRUE/FALSE) that determines if .png files should be replotted
<code>verbose</code>	Boolean (TRUE/FALSE) that determines if TensorFlow and Fit function output should be displayed
<code>ext.dir</code>	External directory (full path)
<code>training.dir</code>	Training directory (full path)

Value

A list of lists containing an ensemble of deep neural networks and weights

Examples

```

ext.dir <- paste0(tempdir(), "/criticality/extdata")
dir.create(ext.dir, recursive = TRUE, showWarnings = FALSE)

extdata <- paste0(.libPaths()[1], "/criticality/extdata")
file.copy(paste0(extdata, "/facility.csv"), ext.dir, recursive = TRUE)
file.copy(paste0(extdata, "/mcnp-dataset.RData"), ext.dir, recursive = TRUE)

config <- FALSE
try(config <- reticulate::py_config())$available)
try(if (config == TRUE) {
  NN(

```

```
    batch.size = 128,  
    ensemble.size = 1,  
    epochs = 10,  
    layers = "256-256-16",  
    loss = "sse",  
    replot = FALSE,  
    ext.dir = ext.dir  
  )  
})
```

Plot*Plot Function*

Description

This function generates and saves plots and data.

Usage

```
Plot(i, history = NULL, plot.dir)
```

Arguments

i	Model number
history	Training history
plot.dir	Plot directory (full path)

Value

No output (generates and saves ggplot2 files and training histories)

Risk*Risk Function*

Description

This function estimates process criticality accident risk (imports Sample function).

Usage

```
Risk(
  bn,
  code = "mcnp",
  cores = parallel::detectCores()/2,
  dist = "gamma",
  facility.data,
  keff.cutoff = 0.9,
  metamodel,
  risk.pool = 100,
  sample.size = 1e+09,
  usl = 0.95,
  ext.dir,
  training.dir = NULL
)
```

Arguments

<code>bn</code>	Bayesian network
<code>code</code>	Monte Carlo radiation transport code (e.g., "cog", "mcnp")
<code>cores</code>	Number of CPU cores to use for generating Bayesian network samples
<code>dist</code>	Truncated probability distribution (e.g., "gamma", "normal")
<code>facility.data</code>	.csv file name
<code>keff.cutoff</code>	keff cutoff value (e.g., $\text{keff} \geq 0.9$)
<code>metamodel</code>	List of deep neural network metamodels and weights
<code>risk.pool</code>	Number of times risk is calculated
<code>sample.size</code>	Number of samples used to calculate risk
<code>usl</code>	Upper subcritical limit (e.g., $\text{keff} \geq 0.95$)
<code>ext.dir</code>	External directory (full path)
<code>training.dir</code>	Training directory (full path)

Value

A list of lists containing process criticality accident risk estimates and Bayesian network samples

Examples

```
ext.dir <- paste0(tempdir(), "/criticality/extdata")
dir.create(ext.dir, recursive = TRUE, showWarnings = FALSE)

extdata <- paste0(.libPaths()[1], "/criticality/extdata")
file.copy(paste0(extdata, "/facility.csv"), ext.dir, recursive = TRUE)
file.copy(paste0(extdata, "/mcnp-dataset.RData"), ext.dir, recursive = TRUE)

config <- FALSE
try(config <- reticulate::py_config()$available)
```

```

try(if (config == TRUE) {
  Risk(
    bn = BN(
      facility.data = "facility.csv",
      ext.dir = ext.dir),
    code = "mcnp",
    cores = 1,
    facility.data = "facility.csv",
    keff.cutoff = 0.5,
    metamodel = NN(
      batch.size = 128,
      ensemble.size = 1,
      epochs = 10,
      layers = "256-256-16",
      replot = FALSE,
      ext.dir = ext.dir),
    risk.pool = 10,
    sample.size = 1e+04,
    ext.dir = ext.dir,
    training.dir = NULL
  )
})

```

Sample

Sample Function

Description

This function samples the Bayesian network and generates keff predictions using a deep neural network metamodel.

Usage

```

Sample(
  bn,
  code = "mcnp",
  cores = parallel::detectCores()/2,
  keff.cutoff = 0.9,
  metamodel,
  sample.size = 1e+09,
  ext.dir,
  risk.dir = NULL
)

```

Arguments

bn	Bayesian network object
code	Monte Carlo radiation transport code (e.g., "cog", "mcnp")

cores	Number of CPU cores to use for generating Bayesian network samples
keff.cutoff	keff cutoff value (e.g., 0.9)
metamodel	List of deep neural network metamodels and weights
sample.size	Number of samples used to calculate risk
ext.dir	External directory (full path)
risk.dir	Risk directory

Value

A list of Bayesian network samples with predicted keff values

Scale	<i>Scale Function</i>
-------	-----------------------

Description

This function centers, scales, and one-hot encodes variables.

Usage

```
Scale(code = "mcnp", dataset = NULL, output, ext.dir)
```

Arguments

code	Monte Carlo radiation transport code (e.g., "cog", "mcnp")
dataset	Training and test data
output	Processed output from Monte Carlo radiation transport code simulations
ext.dir	External directory (full path)

Value

A list of centered, scaled, and one-hot-encoded training and test data

Tabulate	<i>Tabulate Function</i>
----------	--------------------------

Description

This function loads/saves training and test data (imports Scale function).

Usage

```
Tabulate(code = "mcnp", ext.dir)
```

Arguments

code	Monte Carlo radiation transport code (e.g., "cog", "mcnp")
ext.dir	External directory (full path)

Value

A list of centered, scaled, and one-hot-encoded training and test data

Examples

```
ext.dir <- paste0(tempdir(), "/criticality/extdata")
dir.create(ext.dir, recursive = TRUE, showWarnings = FALSE)

extdata <- paste0(.libPaths()[1], "/criticality/extdata")
file.copy(paste0(extdata, "/facility.csv"), ext.dir, recursive = TRUE)
file.copy(paste0(extdata, "/mcnp-dataset.RData"), ext.dir, recursive = TRUE)

Tabulate(
  ext.dir = ext.dir
)
```

Test	<i>Test Function</i>
------	----------------------

Description

This function calculates deep neural network metamodel weights and generates keff predictions for all training and test data.

Usage

```
Test(dataset, ensemble.size = 5, loss = "sse", ext.dir, training.dir)
```

Arguments

<code>dataset</code>	Training and test data
<code>ensemble.size</code>	Number of deep neural networks in the ensemble
<code>loss</code>	Loss function
<code>ext.dir</code>	External directory (full path)
<code>training.dir</code>	Training directory (full path)

Value

A list of deep neural network weights

Index

BN, [2](#)

Model, [2](#)

NN, [3](#)

Plot, [5](#)

Risk, [5](#)

Sample, [7](#)

Scale, [8](#)

Tabulate, [9](#)

Test, [9](#)