

# NeSy4PPM: Multi-attribute (activity and resource) prediction Tutorial

This notebook demonstrates how to use the NeSy4PPM framework for multi-attribute suffix prediction, specifically focused on activity and resource prediction using neural architectures like LSTM and Transformer models. NeSy4PPM combines multi-attribute neural predictions with MP-declare BK compliance to produce accurate and compliant predictions under concept drift.

This notebook walks through the entire NeSy4PPM pipeline, including:

1. Data preparation
2. Learning pipeline
  1. Prefixes preprocessing
  2. Neural Network training
3. Prediction pipeline

## 1. Data preparation

The first step in the NeSy4PPM pipeline is to load and transform the event log (in a .xes, .csv or .xes.gz files) into a symbolic representation using the `LogData` class, where activity and resource labels are mapped to unique ASCII characters. Depending on the input configuration, the log can be:

- A **single event log**, which will be automatically split into training and evaluation subsets based on the case start timestamps.
- A pair of **separate training and test logs**, which will be loaded and concatenated internally.

### A. Single event log:

```
In [7]: from pathlib import Path
from NeSy4PPM.common import log_utils

log_path = Path.cwd().parent/'data'/'input'/'logs'
log_name = "helpdesk.xes"
train_ratio = 0.8
case_name_key = 'case:concept:name'
act_name_key = 'concept:name'
res_name_key = 'org:resource'
timestamp_key = 'time:timestamp'

log_data = log_utils.LogData(log_path=log_path, log_name=log_name, train_ratio=train_ratio)
```

```

        case_name_key=case_name_key,act_name_key=act_name_key,
        res_name_key=res_name_key,timestamp_key=timestamp_key,
print(f"Loaded log: {log_data.log_name}")
print(f"Trace max size: {log_data.max_len}")

```

```

parsing log, completed traces :: 100%|██████████| 4580/4580 [00:00<00:00, 6747.20it/s]

```

```

Loaded log: helpdesk

```

```

Trace max size: 15

```

## B. Separate training and test logs:

```

In [2]: from pathlib import Path
        from NeSy4PPM.common import log_utils

log_path = Path.cwd().parent/'data'/'input'/'logs'
train_log = "helpdesk_train.xes"
test_log = "helpdesk_test.xes"

log_data = log_utils.LogData(log_path=log_path,train_log=train_log,test_log=test_log)
print(f"Loaded log: {log_data.log_name}")
print(f"Trace max size: {log_data.max_len}")

```

```

parsing log, completed traces :: 100%|██████████| 3664/3664 [00:00<00:00, 5120.82it/s]

```

```

parsing log, completed traces :: 100%|██████████| 820/820 [00:00<00:00, 6406.36it/s]

```

```

Loaded log: helpdesk_train

```

```

Trace max size: 15

```

## 2. Learning pipeline

The **Learning Pipeline** is responsible for transforming symbolic traces into neural-compatible inputs and training an LSTM or Transformer model to perform next activity and resource prediction. This phase involves both **Prefixes preprocessing** by extracting and encoding prefixes from training set, and **Neural network training** that learn to generate the most likely continuations of incomplete process traces.

### 2.1 Prefixes preprocessing

The `Prefixes preprocessing` step extracts prefixes (i.e., partial traces executions) from the training log and encodes them into numerical representations suitable for neural models. This can be done by calling `extract_trace_prefixes` and `encode_prefixes` for extracting and encoding prefixes, respectively or only by calling `extract_encode_prefixes` function.

#### Step 1: Prefixes extraction

The `extract_trace_prefixes` function extracts all possible prefixes from each trace in the training log, up to a predefined maximum length. These prefixes represent partial executions of cases and are used as inputs to the neural model.

```
In [3]: from NeSy4PPM.learning.prefixes_preprocessing import extract_trace_prefixes
extracted_prefixes = extract_trace_prefixes(log_data=log_data, resource=True)
```

## Step 2: Prefixes encodings

Before training a neural model, extracted prefixes must be converted into vectorized formats. NeSy4PPM supports four encoding techniques for multi-attribute:

- One-hot encoding ,
- Index-based encoding ,
- Shrunked index-based encoding ,
- Multi-encoders encoding .

Each encoding is implemented via the function `encode_prefixes` and prepares both input features ( `x` ) and two targets labels: `y_a` for activity prediction and `y_g` for resource prediction.

### One-hot encoding

In the **One-hot encoding**, sequences of events are converted into high-dimensional binary feature vectors. Each feature corresponds to a concatenation of one-hot encoded activity and resource values derived from the log. To apply index-based encoding, set the `encoder` parameter to `Encodings.One_hot` when calling the `encode_prefixes` function:

```
In [4]: from NeSy4PPM.learning.prefixes_preprocessing import encode_prefixes
from NeSy4PPM.common.utils import Encodings

x, y_a, y_g = encode_prefixes(log_data, prefixes=extracted_prefixes, encoder=Encodings
```

Total resources: 22 - Target resources: 23

['Value 2', 'Value 5', 'Value 16', 'Value 15', 'Value 21', 'Value 10', 'Value 11', 'Value 12', 'Value 6', 'Value 7', 'Value 9', 'Value 14', 'Value 19', 'Value 17', 'Value 8', 'Value 13', 'Value 22', 'Value 1', 'Value 4', 'Value 3', 'Value 18', 'Value 20']

Total activities: 14 - Target activities: 15

['Assign seriousness', 'Take in charge ticket', 'Resolve ticket', 'Closed', 'Wait', 'Create SW anomaly', 'Insert ticket', 'Schedule intervention', 'INVALID', 'RESOLVED', 'VERIFIED', 'Resolve SW anomaly', 'Require upgrade', 'DUPLICATE']

Num. of learning sequences: 16937

Encoding...

Num. of features: 36

### Index-based encoding

In the **Index-based encoding**, sequences of events are transformed into numerical feature vectors, where each event is represented by a pair of indices: one for the activity and one for the resource. These indices correspond to the positions of the activity and resource in their respective predefined sets. To apply index-based encoding, set the `encoder` parameter to `Encodings.Index_based` when calling the `encode_prefixes` function:

```
In [5]: from NeSy4PPM.learning.prefixes_preprocessing import encode_prefixes
        from NeSy4PPM.common.utils import Encodings

x, y_a, y_g = encode_prefixes(log_data, prefixes=extracted_prefixes, encoder=Encodin

Total resources: 22 - Target resources: 23
['Value 2', 'Value 5', 'Value 16', 'Value 15', 'Value 21', 'Value 10', 'Value 11', 'Value 12', 'Value 6', 'Value 7', 'Value 9', 'Value 14', 'Value 19', 'Value 17', 'Value 8', 'Value 13', 'Value 22', 'Value 1', 'Value 4', 'Value 3', 'Value 18', 'Value 20']
Total activities: 14 - Target activities: 15
['Assign seriousness', 'Take in charge ticket', 'Resolve ticket', 'Closed', 'Wait', 'Create SW anomaly', 'Insert ticket', 'Schedule intervention', 'INVALID', 'RESOLVED', 'VERIFIED', 'Resolve SW anomaly', 'Require upgrade', 'DUPLICATE']
Num. of learning sequences: 16937
Encoding...
Num. of features: 30
```

## Shrunked index-based encoding

In the **Shrunked index-based encoding**, sequences of events are transformed into numerical feature vectors by assigning a unique integer index to each activity–resource pair. To apply shrunked index-based encoding, set the encoder parameter to `Encodings.Shrunked_based` when calling the `encode_prefixes` function:

```
In [6]: from NeSy4PPM.learning.prefixes_preprocessing import encode_prefixes
        from NeSy4PPM.common.utils import Encodings

x, y_a, y_g = encode_prefixes(log_data, prefixes=extracted_prefixes, encoder=Encodin

Total resources: 22 - Target resources: 23
['Value 2', 'Value 5', 'Value 16', 'Value 15', 'Value 21', 'Value 10', 'Value 11', 'Value 12', 'Value 6', 'Value 7', 'Value 9', 'Value 14', 'Value 19', 'Value 17', 'Value 8', 'Value 13', 'Value 22', 'Value 1', 'Value 4', 'Value 3', 'Value 18', 'Value 20']
Total activities: 14 - Target activities: 15
['Assign seriousness', 'Take in charge ticket', 'Resolve ticket', 'Closed', 'Wait', 'Create SW anomaly', 'Insert ticket', 'Schedule intervention', 'INVALID', 'RESOLVED', 'VERIFIED', 'Resolve SW anomaly', 'Require upgrade', 'DUPLICATE']
Num. of learning sequences: 16937
Encoding...
Num. of features: 15
```

## Multi-encoders encoding

In the **Multi-encoders encoding**, sequences of events are represented using separate embedding spaces for activities and resources. Each activity and resource is first embedded

independently, and then enriched with cross-information using a modulation mechanism that captures their interactions. The final representation combines the modulated embeddings using learned alignment weights. To apply multi-encoders encoding, set the encoder parameter to `Encodings.Multi_encoders` when calling the `encode_prefixes` function:

```
In [7]: from NeSy4PPM.learning.prefixes_preprocessing import encode_prefixes
        from NeSy4PPM.common.utils import Encodings

x, y_a, y_g = encode_prefixes(log_data, prefixes=extracted_prefixes, encoder=Encodin

Total resources: 22 - Target resources: 23
['Value 2', 'Value 5', 'Value 16', 'Value 15', 'Value 21', 'Value 10', 'Value 11', 'Value 12', 'Value 6', 'Value 7', 'Value 9', 'Value 14', 'Value 19', 'Value 17', 'Value 8', 'Value 13', 'Value 22', 'Value 1', 'Value 4', 'Value 3', 'Value 18', 'Value 20']
Total activities: 14 - Target activities: 15
['Assign seriousness', 'Take in charge ticket', 'Resolve ticket', 'Closed', 'Wait', 'Create SW anomaly', 'Insert ticket', 'Schedule intervention', 'INVALID', 'RESOLVED', 'VERIFIED', 'Resolve SW anomaly', 'Require upgrade', 'DUPLICATE']
Num. of learning sequences: 16937
Encoding...
Num. of features: 15
```

## Steps 1&2: End-to-end prefixes preprocessing

```
In [3]: from NeSy4PPM.learning.prefixes_preprocessing import extract_encode_prefixes
        from NeSy4PPM.common.utils import Encodings

encoder = Encodings.Index_based
x, y_a, y_g = extract_encode_prefixes(log_data, encoder=encoder, resource=True)

Total resources: 22 - Target resources: 23
['Value 2', 'Value 5', 'Value 16', 'Value 15', 'Value 21', 'Value 10', 'Value 11', 'Value 12', 'Value 6', 'Value 7', 'Value 9', 'Value 14', 'Value 19', 'Value 17', 'Value 8', 'Value 13', 'Value 22', 'Value 1', 'Value 4', 'Value 3', 'Value 18', 'Value 20']
Total activities: 14 - Target activities: 15
['Assign seriousness', 'Take in charge ticket', 'Resolve ticket', 'Closed', 'Wait', 'Create SW anomaly', 'Insert ticket', 'Schedule intervention', 'INVALID', 'RESOLVED', 'VERIFIED', 'Resolve SW anomaly', 'Require upgrade', 'DUPLICATE']
Num. of learning sequences: 16937
Encoding...
Num. of features: 30
```

## 2.2 Neural Network training

Once the prefixes are encoded, NeSy4PPM proceeds to train a neural network that learns to predict the next activity and resource given a partial trace. The training is handled via the `train` function, which takes the encoded prefix data ( `x` , `y_a` , `y_g` ) and builds a model according to the chosen architecture. NeSy4PPM supports two neural architectures:

- **LSTM (Long Short-Term Memory)** networks, which are recurrent neural networks designed to handle sequential data with long-range dependencies. To use LSTM, set the `model_arch` parameter to `NN_model.LSTM`.
- **Transformer** architectures, which use attention mechanisms to model relationships across all positions in the prefix sequence simultaneously. To use a Transformer, set the `model_arch` parameter to `NN_model.Transformer`.

```
In [ ]: from NeSy4PPM.learning.train_model import train
        from NeSy4PPM.common.utils import NN_model

        model = NN_model.Transformer
        model_folder = Path.cwd().parent/'data'/'output'
        train(log_data, encoder, model_arch=model, output_folder=model_folder, x=x, y_a=y_a)
```

## 3. Prediction pipeline

The **Prediction Pipeline** in NeSy4PPM is responsible for generating multi-attribute (activity and resource) suffix predictions from a prefix (i.e., an incomplete trace) using a trained neural model. To enhance both accuracy and compliance under concept drift, it supports two main prediction modes:

- **BK-contextualized Beam Search:** the BK is used *during* beam search to guide which branches are explored based on compliance.
- **BK-based Filtering:** the BK is used *after* the beam search to filter out non-compliant predicted suffixes.

### 3.1 Set prediction parameters

The prediction process begins by specifying the following parameters that control how the prediction algorithm operates:

- `log_data.evaluation_prefix_start`: the minimum prefix length (in events) for prediction.
  - `log_data.evaluation_prefix_end`: the maximum prefix length for prediction.
  - `model_arch`: the trained model architecture ( `NN_model.LSTM` or `NN_model.Transformer` ).
  - `encoder`: the encoding method used during training ( `Encodings.One_hot` , `Encodings.Index_based` , `Encodings.Shrinked_index_based` or `Encodings.Multi_Encoders` ).
  - `output_folder`: the path where the trained model and prediction results are saved.
  - `bk_file_path`: the path to the **BK** (background knowledge) file.
  - `beam_size`: the number of alternative suffixes explored in parallel by the beam search.
- A **simple autoregressive prediction** can be performed by setting `beam_size`

to `0` (greedy search).

- `weight` : a float value in `[0, 1]` that balances the importance of neural predictions and BK compliance. A value of 0 uses only the neural model, while higher values increase the importance of BK during the search.
- `BK_end` : a boolean parameter indicating whether BK is applied at the end (i.e., filtering) instead of during the search.

```
In [5]: from NeSy4PPM.common.utils import NN_model
        from NeSy4PPM.common.utils import Encodings

        (log_data.evaluation_prefix_start, log_data.evaluation_prefix_end) = (1,4)
        model_arch = NN_model.Transformer
        encoder = Encodings.Index_based
        output_folder= Path.cwd().parent/'data'/'output'
        bk_file_path = Path.cwd().parent/'data'/'input'/'declare_models'/'BK_helpdesk.decl'
        beam_size = 3
        weight = [0.9]
        BK_end = False
```

## 3.2 Load the Background Knowledge (BK)

After setting the parameters, a background knowledge (BK) model must be loaded using the `load_bk` function. For multi-attribute prediction, only MP-declare models ( `.decl` ) are supported.

```
In [6]: from NeSy4PPM.common.utils import load_bk

        bk_model = load_bk(bk_file_path)
```

```
0 Existence1[Closed] |A.org:resource is Value 3 |
1 Chain Precedence[Resolve ticket, Closed] |A.org:resource is Value 3 | |
```

## 3.3 Perform Prediction

NeSy4PPM implements the `predict_evaluate` function, which generates activity-resource suffixes using the proposed neuro-symbolic beam search algorithm and computes two evaluation metrics:

- **Damerau-Levenshtein Similarity**, measuring the similarity between the predicted and actual suffixes based on edit distance,
- **Jaccard Similarity**, measuring the overlap between the sets of predicted and actual activities. suffix prediction using a trained neural model and loaded `BK` model.

By default, this function operates on the **entire test log**, predicting suffixes for all traces defined in the test set.

```
In [ ]: ## Entire test log prediction
        from NeSy4PPM.prediction import evaluation
```

```
evaluation.predict_evaluate(log_data, model_arch=model_arch, encoder=encoder,  
                             output_folder=output_folder, bk_model=bk_model, beam_si
```

However, `predict_evaluate` function can also be used to predict suffixes for a specific **subset of traces** by providing a list of case IDs from the test log.

```
In [7]: ## A subset of test log prediction  
from NeSy4PPM.prediction import evaluation  
traces_ids = ['Case 1327']  
evaluation.predict_evaluate(log_data, model_arch=model_arch, encoder=encoder, evalua  
                             output_folder=output_folder, bk_model=bk_model, beam_si
```

DEBUG:h5py.\_conv:Creating converter from 3 to 5

fold 0 - Activity & Resource Prediction

Model filepath: C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM\docs\source\data\output\keras\_trans\_index-based\0\models\CFR\helpdesk\_train

Latest checkpoint file: C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM\docs\source\data\output\keras\_trans\_index-based\0\models\CFR\helpdesk\_train\model\_024-1.193.keras

C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: DeprecationWarning: non-integer arguments to randrange() have been deprecated since Python 3.10 and will be removed in a subsequent version

return self.randrange(a, b+1)

C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: DeprecationWarning: non-integer arguments to randrange() have been deprecated since Python 3.10 and will be removed in a subsequent version

return self.randrange(a, b+1)

C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: DeprecationWarning: non-integer arguments to randrange() have been deprecated since Python 3.10 and will be removed in a subsequent version

return self.randrange(a, b+1)

C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: DeprecationWarning: non-integer arguments to randrange() have been deprecated since Python 3.10 and will be removed in a subsequent version

return self.randrange(a, b+1)

C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: DeprecationWarning: non-integer arguments to randrange() have been deprecated since Python 3.10 and will be removed in a subsequent version

return self.randrange(a, b+1)

C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: DeprecationWarning: non-integer arguments to randrange() have been deprecated since Python 3.10 and will be removed in a subsequent version

return self.randrange(a, b+1)

C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: DeprecationWarning: non-integer arguments to randrange() have been deprecated since Python 3.10 and will be removed in a subsequent version

return self.randrange(a, b+1)

C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: DeprecationWarning: non-integer arguments to randrange() have been deprecated since Python 3.10 and will be removed in a subsequent version

return self.randrange(a, b+1)

C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: DeprecationWarning: non-integer arguments to randrange() have been deprecated since Python 3.10 and will be removed in a subsequent version

return self.randrange(a, b+1)



```
['Case ID', 'Prefix length', 'Trace Prefix Act', 'Ground truth', 'Predicted Acts',  
'Damerau-Levenshtein Acts', 'Jaccard Acts', 'Trace Prefix Res', 'Ground Truth Resources',  
'Predicted Resources', 'Damerau-Levenshtein Resources', 'Jaccard Resources',  
'Damerau-Levenshtein Combined', 'Weight']
```

```
0%|          | 0/1 [00:00<?, ?it/s]C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM  
\venv\lib\site-packages\tqdm\std.py:917: FutureWarning: DataFrameGroupBy.apply opera  
ted on the grouping columns. This behavior is deprecated, and in a future version of  
pandas the grouping columns will be excluded from the operation. Either pass `includ  
e_groups=False` to exclude the groupings or explicitly select the grouping columns a  
fter groupby to silence this warning.
```

```
return getattr(df, df_function)(wrapper, **kwargs)
```

```
100%|██████████| 1/1 [00:05<00:00, 5.77s/it]
```

```
['Case 1327', 1, 'Assign seriousness', 'Wait, Resolve ticket, Closed', 'Take in char  
ge ticket, Resolve ticket, Closed', 0.6666666666666667, 0.5, 'Value 13', 'Value 1, V  
alue 13, Value 3', 'Value 13, Value 13, Value 3', 0.6666666666666667, 0.666666666666  
6666, 0.6666666666666667, 0.9]
```

```
0%|          | 0/1 [00:00<?, ?it/s]C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM  
\venv\lib\site-packages\tqdm\std.py:917: FutureWarning: DataFrameGroupBy.apply opera  
ted on the grouping columns. This behavior is deprecated, and in a future version of  
pandas the grouping columns will be excluded from the operation. Either pass `includ  
e_groups=False` to exclude the groupings or explicitly select the grouping columns a  
fter groupby to silence this warning.
```

```
return getattr(df, df_function)(wrapper, **kwargs)
```

```
100%|██████████| 1/1 [00:03<00:00, 3.93s/it]
```

```
['Case 1327', 2, 'Assign seriousness, Wait', 'Resolve ticket, Closed', 'Resolve tick  
et, Closed', 1.0, 1.0, 'Value 13, Value 1', 'Value 13, Value 3', 'Value 1, Value 3',  
0.5, 0.3333333333333333, 0.75, 0.9]
```

```
0%|          | 0/1 [00:00<?, ?it/s]C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM  
\venv\lib\site-packages\tqdm\std.py:917: FutureWarning: DataFrameGroupBy.apply opera  
ted on the grouping columns. This behavior is deprecated, and in a future version of  
pandas the grouping columns will be excluded from the operation. Either pass `includ  
e_groups=False` to exclude the groupings or explicitly select the grouping columns a  
fter groupby to silence this warning.
```

```
return getattr(df, df_function)(wrapper, **kwargs)
```

```
100%|██████████| 1/1 [00:02<00:00, 2.21s/it]
```

```
['Case 1327', 3, 'Assign seriousness, Wait, Resolve ticket', 'Closed', 'Closed', 1.  
0, 1.0, 'Value 13, Value 1, Value 13', 'Value 3', 'Value 3', 1.0, 1.0, 1.0, 0.9]
```

```

0%|          | 0/1 [00:00<?, ?it/s]C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM
\venv\lib\site-packages\tqdm\std.py:917: FutureWarning: DataFrameGroupBy.apply opera
ted on the grouping columns. This behavior is deprecated, and in a future version of
pandas the grouping columns will be excluded from the operation. Either pass `includ
e_groups=False` to exclude the groupings or explicitly select the grouping columns a
fter groupby to silence this warning.
    return getattr(df, df_function)(wrapper, **kwargs)
100%|██████████| 1/1 [00:00<?, ?it/s]
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
TIME TO FINISH --- 12.620225429534912 seconds ---
fold 1 - Activity & Resource Prediction
Model filepath: C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM\docs\source\data\outp
ut\keras_trans_index-based\1\models\CFR\helpdesk_train
Latest checkpoint file: C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM\docs\source\d
ata\output\keras_trans_index-based\1\models\CFR\helpdesk_train\model_014-1.198.keras
['Case ID', 'Prefix length', 'Trace Prefix Act', 'Ground truth', 'Predicted Acts',
'Damerau-Levenshtein Acts', 'Jaccard Acts', 'Trace Prefix Res', 'Ground Truth Resour
ces', 'Predicted Resources', 'Damerau-Levenshtein Resources', 'Jaccard Resources',
'Damerau-Levenshtein Combined', 'Weight']

```

```
0%|          | 0/1 [00:00<?, ?it/s]C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM
\venv\lib\site-packages\tqdm\std.py:917: FutureWarning: DataFrameGroupBy.apply opera
ted on the grouping columns. This behavior is deprecated, and in a future version of
pandas the grouping columns will be excluded from the operation. Either pass `includ
e_groups=False` to exclude the groupings or explicitly select the grouping columns a
fter groupby to silence this warning.
```

```
    return getattr(df, df_function)(wrapper, **kwargs)
```

```
100%|██████████| 1/1 [00:05<00:00, 5.45s/it]
```

```
['Case 1327', 1, 'Assign seriousness', 'Wait, Resolve ticket, Closed', 'Take in char
ge ticket, Resolve ticket, Closed', 0.6666666666666667, 0.5, 'Value 13', 'Value 1, V
alue 13, Value 3', 'Value 13, Value 13, Value 3', 0.6666666666666667, 0.66666666666
6666, 0.6666666666666667, 0.9]
```

```
0%|          | 0/1 [00:00<?, ?it/s]C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM
\venv\lib\site-packages\tqdm\std.py:917: FutureWarning: DataFrameGroupBy.apply opera
ted on the grouping columns. This behavior is deprecated, and in a future version of
pandas the grouping columns will be excluded from the operation. Either pass `includ
e_groups=False` to exclude the groupings or explicitly select the grouping columns a
fter groupby to silence this warning.
```

```
    return getattr(df, df_function)(wrapper, **kwargs)
```

```
100%|██████████| 1/1 [00:03<00:00, 3.84s/it]
```

```
['Case 1327', 2, 'Assign seriousness, Wait', 'Resolve ticket, Closed', 'Resolve tick
et, Closed', 1.0, 1.0, 'Value 13, Value 1', 'Value 13, Value 3', 'Value 2, Value 3',
0.5, 0.3333333333333333, 0.75, 0.9]
```

```
0%|          | 0/1 [00:00<?, ?it/s]C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM
\venv\lib\site-packages\tqdm\std.py:917: FutureWarning: DataFrameGroupBy.apply opera
ted on the grouping columns. This behavior is deprecated, and in a future version of
pandas the grouping columns will be excluded from the operation. Either pass `includ
e_groups=False` to exclude the groupings or explicitly select the grouping columns a
fter groupby to silence this warning.
```

```
    return getattr(df, df_function)(wrapper, **kwargs)
```

```
100%|██████████| 1/1 [00:02<00:00, 2.16s/it]
```

```
['Case 1327', 3, 'Assign seriousness, Wait, Resolve ticket', 'Closed', 'Closed', 1.
0, 1.0, 'Value 13, Value 1, Value 13', 'Value 3', 'Value 3', 1.0, 1.0, 1.0, 0.9]
```

```

0%|          | 0/1 [00:00<?, ?it/s]C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM
\venv\lib\site-packages\tqdm\std.py:917: FutureWarning: DataFrameGroupBy.apply opera
ted on the grouping columns. This behavior is deprecated, and in a future version of
pandas the grouping columns will be excluded from the operation. Either pass `includ
e_groups=False` to exclude the groupings or explicitly select the grouping columns a
fter groupby to silence this warning.
    return getattr(df, df_function)(wrapper, **kwargs)
100%|██████████| 1/1 [00:00<?, ?it/s]
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
C:\Users\JOukharijane\AppData\Local\Programs\Python\Python310\lib\random.py:370: Dep
recationWarning: non-integer arguments to randrange() have been deprecated since Pyt
hon 3.10 and will be removed in a subsequent version
    return self.randrange(a, b+1)
TIME TO FINISH --- 24.26862668991089 seconds ---
fold 2 - Activity & Resource Prediction
Model filepath: C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM\docs\source\data\outp
ut\keras_trans_index-based\2\models\CFR\helpdesk_train
Latest checkpoint file: C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM\docs\source\d
ata\output\keras_trans_index-based\2\models\CFR\helpdesk_train\model_022-1.191.keras
['Case ID', 'Prefix length', 'Trace Prefix Act', 'Ground truth', 'Predicted Acts',
'Damerau-Levenshtein Acts', 'Jaccard Acts', 'Trace Prefix Res', 'Ground Truth Resour
ces', 'Predicted Resources', 'Damerau-Levenshtein Resources', 'Jaccard Resources',
'Damerau-Levenshtein Combined', 'Weight']

```

```
0%|          | 0/1 [00:00<?, ?it/s]C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM
\venv\lib\site-packages\tqdm\std.py:917: FutureWarning: DataFrameGroupBy.apply opera
ted on the grouping columns. This behavior is deprecated, and in a future version of
pandas the grouping columns will be excluded from the operation. Either pass `includ
e_groups=False` to exclude the groupings or explicitly select the grouping columns a
fter groupby to silence this warning.
```

```
    return getattr(df, df_function)(wrapper, **kwargs)
```

```
100%|██████████| 1/1 [00:05<00:00,  5.44s/it]
```

```
['Case 1327', 1, 'Assign seriousness', 'Wait, Resolve ticket, Closed', 'Take in char
ge ticket, Resolve ticket, Closed', 0.6666666666666667, 0.5, 'Value 13', 'Value 1, V
alue 13, Value 3', 'Value 13, Value 13, Value 3', 0.6666666666666667, 0.66666666666
6666, 0.6666666666666667, 0.9]
```

```
0%|          | 0/1 [00:00<?, ?it/s]C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM
\venv\lib\site-packages\tqdm\std.py:917: FutureWarning: DataFrameGroupBy.apply opera
ted on the grouping columns. This behavior is deprecated, and in a future version of
pandas the grouping columns will be excluded from the operation. Either pass `includ
e_groups=False` to exclude the groupings or explicitly select the grouping columns a
fter groupby to silence this warning.
```

```
    return getattr(df, df_function)(wrapper, **kwargs)
```

```
100%|██████████| 1/1 [00:03<00:00,  3.80s/it]
```

```
['Case 1327', 2, 'Assign seriousness, Wait', 'Resolve ticket, Closed', 'Resolve tick
et, Closed', 1.0, 1.0, 'Value 13, Value 1', 'Value 13, Value 3', 'Value 1, Value 3',
0.5, 0.33333333333333326, 0.75, 0.9]
```

```
0%|          | 0/1 [00:00<?, ?it/s]C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM
\venv\lib\site-packages\tqdm\std.py:917: FutureWarning: DataFrameGroupBy.apply opera
ted on the grouping columns. This behavior is deprecated, and in a future version of
pandas the grouping columns will be excluded from the operation. Either pass `includ
e_groups=False` to exclude the groupings or explicitly select the grouping columns a
fter groupby to silence this warning.
```

```
    return getattr(df, df_function)(wrapper, **kwargs)
```

```
100%|██████████| 1/1 [00:02<00:00,  2.15s/it]
```

```
['Case 1327', 3, 'Assign seriousness, Wait, Resolve ticket', 'Closed', 'Closed', 1.
0, 1.0, 'Value 13, Value 1, Value 13', 'Value 3', 'Value 3', 1.0, 1.0, 1.0, 0.9]
```

```
0%|          | 0/1 [00:00<?, ?it/s]C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM
\venv\lib\site-packages\tqdm\std.py:917: FutureWarning: DataFrameGroupBy.apply opera
ted on the grouping columns. This behavior is deprecated, and in a future version of
pandas the grouping columns will be excluded from the operation. Either pass `includ
e_groups=False` to exclude the groupings or explicitly select the grouping columns a
fter groupby to silence this warning.
```

```
    return getattr(df, df_function)(wrapper, **kwargs)
```

```
100%|██████████| 1/1 [00:00<?, ?it/s]
```

```
TIME TO FINISH --- 35.831690311431885 seconds ---
```