NeSy4PPM: Single-attribute (activity) prediction Tutorial

This notebook demonstrates how to use the NeSy4PPM framework for single-attribute suffix prediction, specifically focused on activity prediction using neural architectures like LSTM and Transformer models. NeSy4PPM combines neural learning with symbolic background knowledge (BK) to produce accurate and compliant predictions under various conditions, including concept drift.

This notebook guides you through the full 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` format) into a symbolic representation using the `LogData` class, where activity 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 test subsets based on the case start timestamps.
- A pair of **separate training and test logs**.

A. Single event log:

In [11]:
```
from pathlib import Path
from NeSy4PPM.commons 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'
timestamp_key = 'time:timestamp'

log_data = log_utils.LogData(log_path=log_path,log_name=log_name,train_ratio=train_
                             case_name_key=case_name_key,act_name_key=act_name_key,
                             timestamp_key=timestamp_key,resource=False)
```

```
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, 7500.51it/
s]
Loaded log: helpdesk
Trace max size: 15
```

B. Separate training and test logs:

In [2]:
```
from pathlib import Path
from NeSy4PPM.commons import log_utils
log_path = Path.cwd().parent/'data'/'input'/'logs'
train_log = "helpdesk_train.xes"
test_log = "helpdesk_filtred.xes"

log_data = log_utils.LogData(log_path=log_path,train_log=train_log,test_log=test_lo
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, 9148.38it/
s]
parsing log, completed traces :: 100%|██████████| 320/320 [00:00<00:00, 10232.19it/
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 prediction. This phase involves both **Prefixes preprocessing** by extracting and encoding prefixes form 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)
```

Step 2: Prefixes encodings

To enable deep learning, extracted prefixes must be converted into vectorized formats. NeSy4PPM supports two encoding techniques for single attribute: (i) `One-hot` and (ii) `Index-based` encodings. Each encoding is implemented via the function `encode_prefixes` and prepares both input features (`x`) and target labels `y_a` for activity prediction.

One-hot encoding

In **One-hot encoding**, sequences of events are converted into high-dimensional binary feature vectors. Each feature corresponds to a one-hot encoded activity values derived from the log. This encoding can be applied by setting the `encoder` parameter to `Encodings.One_hot` in the `encode_prefixes` function:

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

x, y_a, target_res= encode_prefixes(log_data,prefixes=extracted_prefixes,encoder=En
```

```
Total activities: 13 - Target activities: 14
        ['Assign seriousness', 'Take in charge ticket', 'Resolve ticket', 'Closed',
'Wait', 'Create SW anomaly', 'Insert ticket', 'Schedule intervention', 'INVALID', 'R
ESOLVED', 'VERIFIED', 'Resolve SW anomaly', 'Require upgrade']
Num. of learning sequences: 16937
Num. of features: 14
```

Index-based encoding

In **Index-based encoding**, sequences of events are transformed into numerical feature vectors. Each event is represented by an integer index that corresponds to the position of the activity in the predefined activity set. 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.commons.utils import Encodings

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

```
Total activities: 13 - Target activities: 14
        ['Assign seriousness', 'Take in charge ticket', 'Resolve ticket', 'Closed',
'Wait', 'Create SW anomaly', 'Insert ticket', 'Schedule intervention', 'INVALID', 'R
ESOLVED', 'VERIFIED', 'Resolve SW anomaly', 'Require upgrade']
Num. of learning sequences: 16937
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.commons.utils import Encodings

encoder = Encodings.Index_based
x, y_a, _ = extract_encode_prefixes(log_data, encoder=encoder)
```

```
Total activities: 13 - Target activities: 14
        ['Assign seriousness', 'Take in charge ticket', 'Resolve ticket', 'Closed',
'Wait', 'Create SW anomaly', 'Insert ticket', 'Schedule intervention', 'INVALID', 'R
ESOLVED', 'VERIFIED', 'Resolve SW anomaly', 'Require upgrade']
Num. of learning sequences: 16937
Num. of features: 15
```

2.2 Neural Network training

Once the prefixes are encoded, NeSy4PPM proceeds to train a neural network that learns to predict the next activity given a partial trace. The training is handled via the `train` function, which takes the encoded prefix data (`x` , `y_a`) 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.commons.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 activity 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` or `Encodings.Index_based`).
- `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,
- `fitness_method` : the method used to compute compliance scores—i.e., the alignment or replay fitness between the predicted trace and the procedural model. This parameter is only applicable when the BK model is procedural, and must be set to one of the following: : `conformance_diagnostics_alignments` , `fitness_alignments` , `conformance_diagnostics_token_based_replay` or `fitness_token_based_replay` .

In [7]:
```
from NeSy4PPM.commons.utils import NN_model
from NeSy4PPM.commons.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_filtr
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. `BK` can represent domain constraints or business rules, and can be encoded in various formats:

- **Procedural models**: `.bpmn` (Business Process Model and Notation), `.pnml` (Petri Nets)
- **Declare models**: `.decl` (Declare constraints)
- **Probabilistic Declare models**: `.txt` (Declare constraints annotated with probabilities)

In [8]:
```
from NeSy4PPM.commons.utils import load_bk

bk_model = load_bk(bk_file_path)
```

```
0 Alternate Precedence[Wait, Closed] | | |
1 Alternate Response[Assign seriousness, Wait] | | |
2 Alternate Precedence[Wait, Resolve ticket] | | |
3 Exactly1[Wait] | |
4 Chain Response[Take in charge ticket, Wait] | | |
```

3.3 Perform Prediction

NeSy4PPM implements the `predict_evaluate` function, which generates activity 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 [ ]:
```python
### 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 [9]:
```python
### 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 Prediction
Model filepath: C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM\docs\source\data\outp
ut\keras_trans_index-based\0\models\CF\helpdesk_train
Latest checkpoint file: C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM\docs\source\d
ata\output\keras_trans_index-based\0\models\CF\helpdesk_train\model_020-0.424.keras
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)
```

```
['Case ID', 'Prefix length', 'Trace Prefix Act', 'Ground truth', 'Predicted', 'Damer
au-Levenshtein', 'Jaccard', '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 [00:00<00:00,  1.08it/s]
['Case 1327', 1, 'Assign seriousness', 'Wait, Resolve ticket, Closed', 'Wait, Resolv
e ticket, Closed', 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<00:00,  2.21it/s]
['Case 1327', 2, 'Assign seriousness, Wait', 'Resolve ticket, Closed', 'Resolve tick
et, Closed', 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<00:00,  4.72it/s]
['Case 1327', 3, 'Assign seriousness, Wait, Resolve ticket', 'Closed', 'Closed', 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)
TIME TO FINISH --- 2.2968838214874268 seconds ---
fold 1 - Activity Prediction
Model filepath: C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM\docs\source\data\outp
ut\keras_trans_index-based\1\models\CF\helpdesk_train
Latest checkpoint file: C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM\docs\source\d
ata\output\keras_trans_index-based\1\models\CF\helpdesk_train\model_022-0.424.keras
['Case ID', 'Prefix length', 'Trace Prefix Act', 'Ground truth', 'Predicted', 'Damer
au-Levenshtein', 'Jaccard', '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 [00:00<00:00,  1.31it/s]
['Case 1327', 1, 'Assign seriousness', 'Wait, Resolve ticket, Closed', 'Wait, Resolv
e ticket, Closed', 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<00:00,  2.63it/s]
['Case 1327', 2, 'Assign seriousness, Wait', 'Resolve ticket, Closed', 'Resolve tick
et, Closed', 1.0, 1.0, 0.9]
  0%|          | 0/1 [00:00<?, ?it/s]
['Case 1327', 3, 'Assign seriousness, Wait, Resolve ticket', 'Closed', 'Closed', 1.
0, 1.0, 0.9]
```

```
C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM\venv\lib\site-packages\tqdm\std.py:91
7: FutureWarning: DataFrameGroupBy.apply operated on the grouping columns. This beha
vior is deprecated, and in a future version of pandas the grouping columns will be e
xcluded from the operation. Either pass `include_groups=False` to exclude the groupi
ngs or explicitly select the grouping columns after groupby to silence this warning.
  return getattr(df, df_function)(wrapper, **kwargs)
100%|██████████| 1/1 [00:00<00:00,  5.30it/s]
  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)
```

```
TIME TO FINISH --- 3.8046345710754395 seconds ---
fold 2 - Activity Prediction
Model filepath: C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM\docs\source\data\outp
ut\keras_trans_index-based\2\models\CF\helpdesk_train
Latest checkpoint file: C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM\docs\source\d
ata\output\keras_trans_index-based\2\models\CF\helpdesk_train\model_021-0.423.keras
['Case ID', 'Prefix length', 'Trace Prefix Act', 'Ground truth', 'Predicted', 'Damer
au-Levenshtein', 'Jaccard', '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 [00:00<00:00,  1.40it/s]
['Case 1327', 1, 'Assign seriousness', 'Wait, Resolve ticket, Closed', 'Wait, Resolv
e ticket, Closed', 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<00:00,  2.61it/s]
['Case 1327', 2, 'Assign seriousness, Wait', 'Resolve ticket, Closed', 'Resolve tick
et, Closed', 1.0, 1.0, 0.9]
  0%|          | 0/1 [00:00<?, ?it/s]
['Case 1327', 3, 'Assign seriousness, Wait, Resolve ticket', 'Closed', 'Closed', 1.
0, 1.0, 0.9]
C:\Users\JOukharijane\Desktop\PostDoc\NeSy4PPM\venv\lib\site-packages\tqdm\std.py:91
7: FutureWarning: DataFrameGroupBy.apply operated on the grouping columns. This beha
vior is deprecated, and in a future version of pandas the grouping columns will be e
xcluded from the operation. Either pass `include_groups=False` to exclude the groupi
ngs or explicitly select the grouping columns after groupby to silence this warning.
  return getattr(df, df_function)(wrapper, **kwargs)
100%|██████████| 1/1 [00:00<00:00,  5.25it/s]
  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 --- 5.233134746551514 seconds ---
```