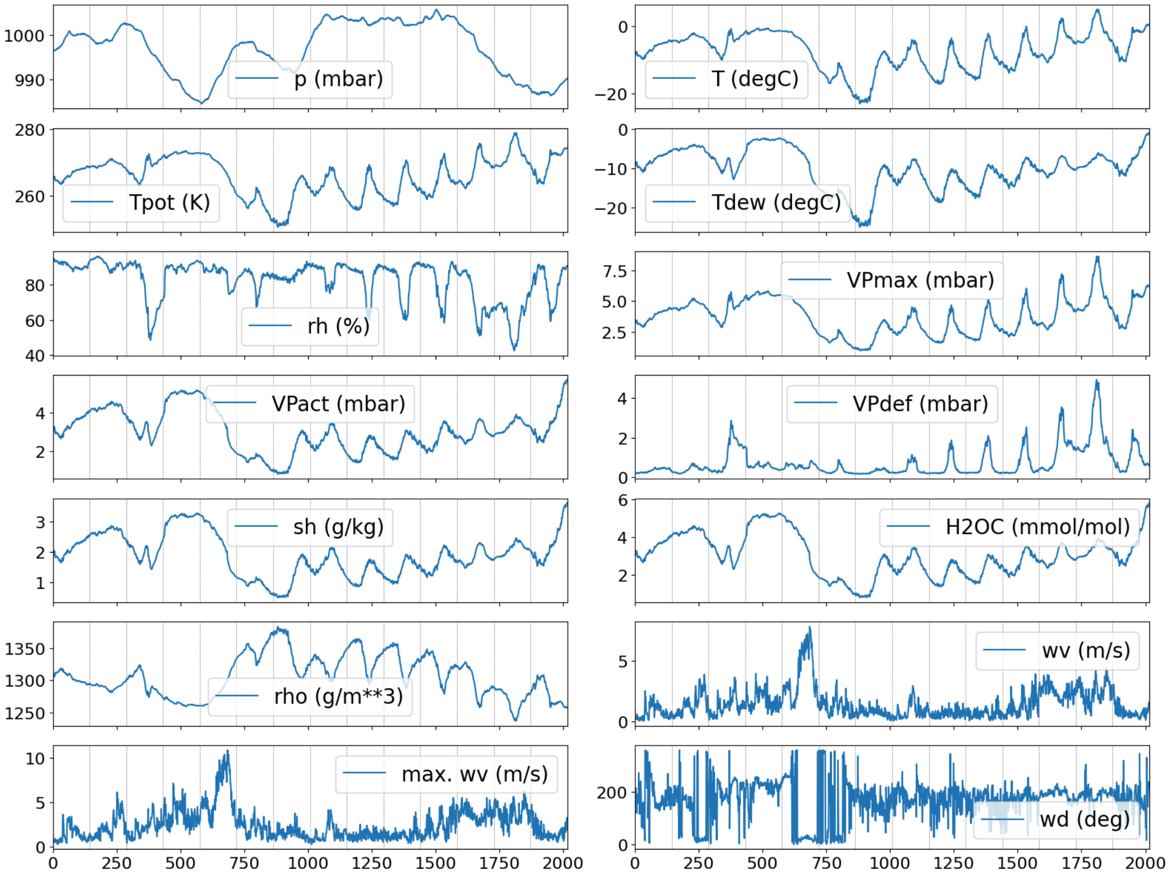
In [ ]:

```python
#%% import libraries and load data

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import deeptrack as dt
import torch
from torch.utils.data import DataLoader
import torch.nn as nn
import deeplay as dl

dataframe = pd.read_csv("jena_climate_2009_2016.csv", index_col=0)
data = dataframe.values
header = dataframe.columns.tolist()

start, days, daily_samples = 0, 14, 144
end = start + daily_samples * days

fig, axs = plt.subplots(7, 2, figsize=(16, 12), sharex=True)
for i, ax in enumerate(axs.flatten()):
    ax.plot(np.arange(start, end), data[start:end, i], label=header[i])
    ax.set_xlim(start, end)
    ax.tick_params(axis="both", which="major", labelsize=16)
    ax.legend(fontsize=20)

    for day in range(1, days):
        ax.axvline(x=start + daily_samples * day,
                   color="gray", linestyle="--", linewidth=0.5)
plt.tight_layout()
plt.show()
```

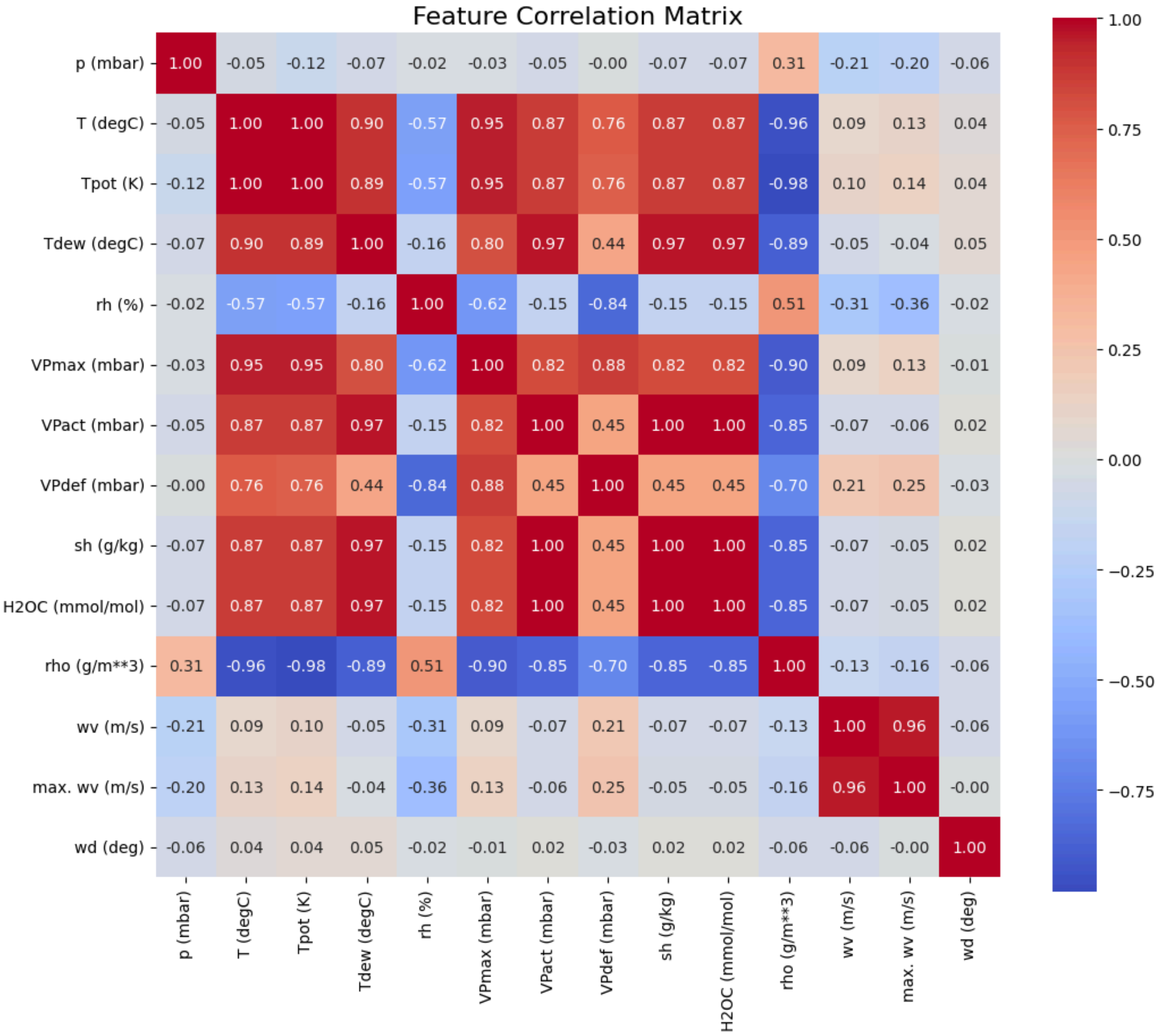
```
c:\Users\Green\AppData\Local\Programs\Python\Python311\Lib\site-packages\deeptrac
k\__init__.py:14: UserWarning: TensorFlow is detected in your environment. DeepTr
ack2 version 2.0++ no longer supports TensorFlow. If you need TensorFlow support,
please install the legacy version 1.7 of DeepTrack2:

    pip install deeptrack==1.7

For more details, refer to the DeepTrack documentation.
  warnings.warn(
WARNING:pint.util:Redefining '[magnetic_flux]' (<class 'pint.delegates.txt_defpar
ser.plain.DerivedDimensionDefinition'>)
```



In [ ]:

```
# %% Show corr matrix and redundant features

correlation_matrix = dataframe.corr()

plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm', square=T
plt.title("Feature Correlation Matrix", fontsize=16)
plt.show()

threshold = 0.8

high_corr_pairs = []
for i in range(len(correlation_matrix.columns)):
    for j in range(i):
        if abs(correlation_matrix.iloc[i, j]) > threshold:
            pair = (correlation_matrix.columns[i], correlation_matrix.columns[j]
            high_corr_pairs.append(pair)

# Print correlated pairs
for col1, col2, corr in high_corr_pairs:
    print(f"{col1} and {col2} have a correlation of {corr:.2f}")

reduced_df = dataframe.drop(columns=[
    "Tpot (K)",
    "Tdew (degC)",
    "VPmax (mbar)",
    "VPdef (mbar)",
    "sh (g/kg)",
    "H2OC (mmol/mol)",
    "max. wv (m/s)"
])
""" Decided to keep T as temperature units, VPact as Pressure,
rho as air density and wv as wind. Drop 7 (half)."""
```

```
data = reduced_df.values
header = reduced_df.columns.tolist()
```

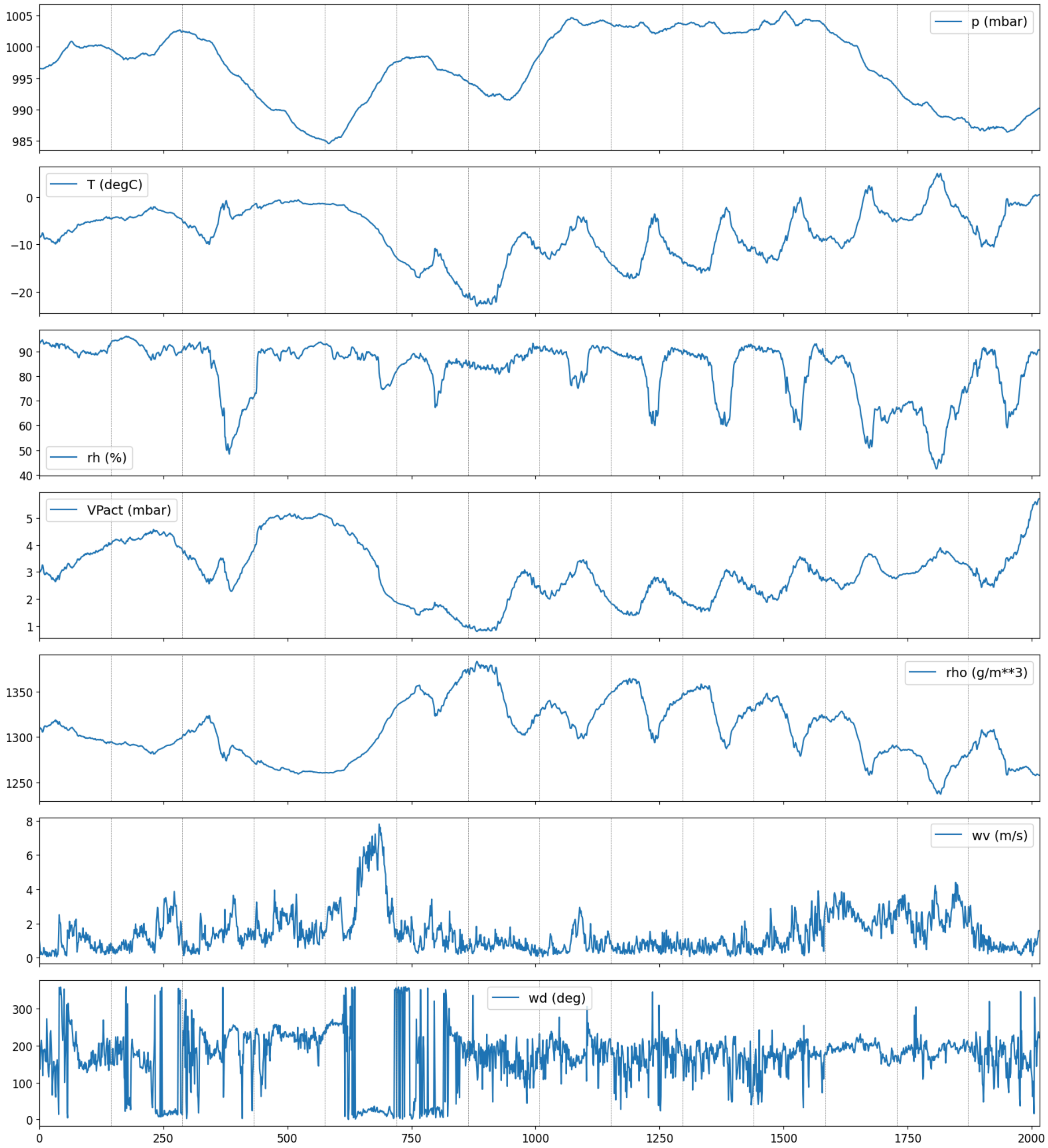


	p (mbar)	T (degC)	Tpot (K)	Tdew (degC)	rh (%)	VPmax (mbar)	VPact (mbar)	VPdef (mbar)	sh (g/kg)	H2OC (mmol/mol)	rho (g/m**3)	wv (m/s)	max. wv (m/s)	wd (deg)
p (mbar)	1.00	-0.05	-0.12	-0.07	-0.02	-0.03	-0.05	-0.00	-0.07	-0.07	0.31	-0.21	-0.20	-0.06
T (degC)	-0.05	1.00	1.00	0.90	-0.57	0.95	0.87	0.76	0.87	0.87	-0.96	0.09	0.13	0.04
Tpot (K)	-0.12	1.00	1.00	0.89	-0.57	0.95	0.87	0.76	0.87	0.87	-0.98	0.10	0.14	0.04
Tdew (degC)	-0.07	0.90	0.89	1.00	-0.16	0.80	0.97	0.44	0.97	0.97	-0.89	-0.05	-0.04	0.05
rh (%)	-0.02	-0.57	-0.57	-0.16	1.00	-0.62	-0.15	-0.84	-0.15	-0.15	0.51	-0.31	-0.36	-0.02
VPmax (mbar)	-0.03	0.95	0.95	0.80	-0.62	1.00	0.82	0.88	0.82	0.82	-0.90	0.09	0.13	-0.01
VPact (mbar)	-0.05	0.87	0.87	0.97	-0.15	0.82	1.00	0.45	1.00	1.00	-0.85	-0.07	-0.06	0.02
VPdef (mbar)	-0.00	0.76	0.76	0.44	-0.84	0.88	0.45	1.00	0.45	0.45	-0.70	0.21	0.25	-0.03
sh (g/kg)	-0.07	0.87	0.87	0.97	-0.15	0.82	1.00	0.45	1.00	1.00	-0.85	-0.07	-0.05	0.02
H2OC (mmol/mol)	-0.07	0.87	0.87	0.97	-0.15	0.82	1.00	0.45	1.00	1.00	-0.85	-0.07	-0.05	0.02
rho (g/m**3)	0.31	-0.96	-0.98	-0.89	0.51	-0.90	-0.85	-0.70	-0.85	-0.85	1.00	-0.13	-0.16	-0.06
wv (m/s)	-0.21	0.09	0.10	-0.05	-0.31	0.09	-0.07	0.21	-0.07	-0.07	-0.13	1.00	0.96	-0.06
max. wv (m/s)	-0.20	0.13	0.14	-0.04	-0.36	0.13	-0.06	0.25	-0.05	-0.05	-0.16	0.96	1.00	-0.00
wd (deg)	-0.06	0.04	0.04	0.05	-0.02	-0.01	0.02	-0.03	0.02	0.02	-0.06	-0.06	-0.00	1.00

```
Tpot (K) and T (degC) have a correlation of 1.00
Tdew (degC) and T (degC) have a correlation of 0.90
Tdew (degC) and Tpot (K) have a correlation of 0.89
VPmax (mbar) and T (degC) have a correlation of 0.95
VPmax (mbar) and Tpot (K) have a correlation of 0.95
VPact (mbar) and T (degC) have a correlation of 0.87
VPact (mbar) and Tpot (K) have a correlation of 0.87
VPact (mbar) and Tdew (degC) have a correlation of 0.97
VPact (mbar) and VPmax (mbar) have a correlation of 0.82
VPdef (mbar) and rh (%) have a correlation of -0.84
VPdef (mbar) and VPmax (mbar) have a correlation of 0.88
sh (g/kg) and T (degC) have a correlation of 0.87
sh (g/kg) and Tpot (K) have a correlation of 0.87
sh (g/kg) and Tdew (degC) have a correlation of 0.97
sh (g/kg) and VPmax (mbar) have a correlation of 0.82
sh (g/kg) and VPact (mbar) have a correlation of 1.00
H2OC (mmol/mol) and T (degC) have a correlation of 0.87
H2OC (mmol/mol) and Tpot (K) have a correlation of 0.87
H2OC (mmol/mol) and Tdew (degC) have a correlation of 0.97
H2OC (mmol/mol) and VPmax (mbar) have a correlation of 0.82
H2OC (mmol/mol) and VPact (mbar) have a correlation of 1.00
H2OC (mmol/mol) and sh (g/kg) have a correlation of 1.00
rho (g/m**3) and T (degC) have a correlation of -0.96
rho (g/m**3) and Tpot (K) have a correlation of -0.98
rho (g/m**3) and Tdew (degC) have a correlation of -0.89
rho (g/m**3) and VPmax (mbar) have a correlation of -0.90
rho (g/m**3) and VPact (mbar) have a correlation of -0.85
rho (g/m**3) and sh (g/kg) have a correlation of -0.85
rho (g/m**3) and H2OC (mmol/mol) have a correlation of -0.85
max. wv (m/s) and wv (m/s) have a correlation of 0.96
```

In [ ]:

```python
#%% Plot reduced data frame

fig, axs = plt.subplots(len(header), 1, figsize=(16, 2.5 * len(header)), sharex=
if len(header) == 1:
    axs = [axs]  # Make iterable if only one subplot

for i, ax in enumerate(axs):
    ax.plot(np.arange(start, end), data[start:end, i], label=header[i])
    ax.set_xlim(start, end)
    ax.tick_params(axis="both", which="major", labelsize=12)
    ax.legend(fontsize=14)

    for day in range(1, days):
        ax.axvline(x=start + daily_samples * day,
                   color="gray", linestyle="--", linewidth=0.5)

plt.tight_layout()
plt.show()
```



In [ ]:

```
# %% Preprocessing the data

n_samples, n_features = data.shape[0], data.shape[1]
past_seq = 2 * daily_samples
lag = 72
temp_idx = 1  # Temperature (Celsius) index.

in_sequences, targets = [], []
for i in range(past_seq, n_samples - lag, daily_samples):
    in_sequences.append(data[i - past_seq:i, :])
    targets.append(data[i + lag:i + lag + 1, temp_idx])
in_sequences, targets = np.asarray(in_sequences), np.asarray(targets)

print(in_sequences.shape)
print(targets.shape)

sources = dt.sources.Source(inputs=in_sequences, targets=targets)
train_sources, val_sources = dt.sources.random_split(sources, [0.8, 0.2])

train_mean = np.mean([src["inputs"] for src in train_sources], axis=(0, 1))
train_std = np.std([src["inputs"] for src in train_sources], axis=(0, 1))
```

```python
inputs_pipeline = (dt.Value(sources.inputs - train_mean) / train_std
                   >> dt.pytorch.ToTensor(dtype=torch.float))
targets_pipeline = (dt.Value(sources.targets - train_mean[temp_idx])
                    / train_std[temp_idx])

train_dataset = dt.pytorch.Dataset(inputs_pipeline & targets_pipeline,
                                   inputs=train_sources)
val_dataset = dt.pytorch.Dataset(inputs_pipeline & targets_pipeline,
                                 inputs=val_sources)

train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=32, shuffle=False)
```

```
(2918, 288, 7)
(2918, 1)
```

In [ ]:

```python
# %% Common-Sense Benchmark and decide device

temperature = data[:, temp_idx]
benchmark_celsius = np.mean(
    np.abs(
        temperature[daily_samples + lag::daily_samples]
        - temperature[lag:-(daily_samples - lag):daily_samples]
    )
)
benchmark = benchmark_celsius / train_std[temp_idx]

print(f"Benchmark Celsius: {benchmark_celsius}")

print(f"Normalized Benchmark: {benchmark}")

def get_device():
    """Select device where to perform the computations."""
    if torch.cuda.is_available():
        return torch.device("cuda:0")
    elif torch.backends.mps.is_available():
        return torch.device("mps")
    else:
        return torch.device("cpu")

device = get_device()
print(device)
```

```
Benchmark Celsius: 2.664549503254539
Normalized Benchmark: 0.3171728842761766
cuda:0
```

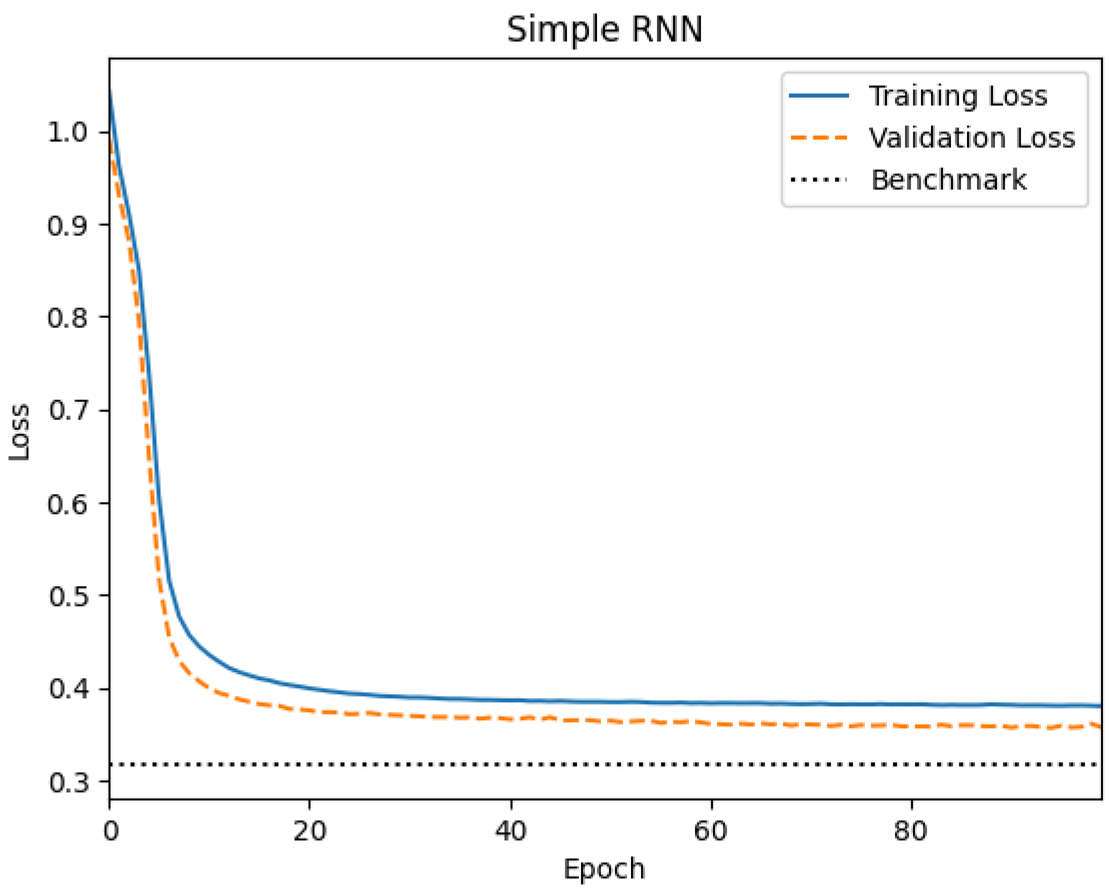
In [ ]:

```python
# %% Simple recurrent Neural Network and training

rnn = nn.RNN(input_size=in_sequences.shape[2], hidden_size=2, batch_first=True)
fc = nn.Linear(in_features=2, out_features=1)
rnn.to(device); fc.to(device);

criterion = nn.L1Loss()  # MAE Loss.
parameter_list = list(rnn.parameters()) + list(fc.parameters())
optimizer = torch.optim.Adam(parameter_list, lr=0.001)

epochs = 100
train_losses, val_losses = [], []
for epoch in range(epochs):
    train_loss = 0.0
    for in_sequences, targets in train_loader:
        optimizer.zero_grad()

        in_sequences, targets = in_sequences.to(device), targets.to(device)
        hidden_sequences, _ = rnn(in_sequences)  # RNN layer.
        last_hidden_states = hidden_sequences[:, -1, :]  # Last hidden states.
        predictions = fc(last_hidden_states)

        loss = criterion(predictions, targets)
        loss.backward()
        optimizer.step()
        train_loss += loss.item()
    train_losses.append(train_loss / len(train_loader))
    print(f"Epoch {epoch} Training Loss: {train_losses[-1]:.4f}")

    val_loss = 0.0
    with torch.no_grad():
        for in_sequences, targets in val_loader:
            in_sequences, targets = in_sequences.to(device), targets.to(device)
            hidden_sequences, _ = rnn(in_sequences)
            last_hidden_states = hidden_sequences[:, -1, :]
            predictions = fc(last_hidden_states)

            loss = criterion(predictions, targets)
            val_loss += loss.item()
        val_losses.append(val_loss / len(val_loader))
        print(f"Epoch {epoch} Validation Loss: {val_losses[-1]:.4f}")
```

```
Epoch 0 Training Loss: 1.0423
Epoch 0 Validation Loss: 0.9912
Epoch 1 Training Loss: 0.9616
Epoch 1 Validation Loss: 0.9304
Epoch 2 Training Loss: 0.9096
Epoch 2 Validation Loss: 0.8802
Epoch 3 Training Loss: 0.8490
Epoch 3 Validation Loss: 0.7937
Epoch 4 Training Loss: 0.7364
Epoch 4 Validation Loss: 0.6470
Epoch 5 Training Loss: 0.6022
Epoch 5 Validation Loss: 0.5160
Epoch 6 Training Loss: 0.5151
Epoch 6 Validation Loss: 0.4540
Epoch 7 Training Loss: 0.4765
Epoch 7 Validation Loss: 0.4286
Epoch 8 Training Loss: 0.4566
Epoch 8 Validation Loss: 0.4153
Epoch 9 Training Loss: 0.4442
Epoch 9 Validation Loss: 0.4067
Epoch 10 Training Loss: 0.4351
Epoch 10 Validation Loss: 0.3990
Epoch 11 Training Loss: 0.4276
Epoch 11 Validation Loss: 0.3940
Epoch 12 Training Loss: 0.4209
Epoch 12 Validation Loss: 0.3910
Epoch 13 Training Loss: 0.4165
Epoch 13 Validation Loss: 0.3872
Epoch 14 Training Loss: 0.4130
Epoch 14 Validation Loss: 0.3845
Epoch 15 Training Loss: 0.4097
Epoch 15 Validation Loss: 0.3821
Epoch 16 Training Loss: 0.4077
Epoch 16 Validation Loss: 0.3808
Epoch 17 Training Loss: 0.4046
Epoch 17 Validation Loss: 0.3801
Epoch 18 Training Loss: 0.4027
Epoch 18 Validation Loss: 0.3769
Epoch 19 Training Loss: 0.4010
Epoch 19 Validation Loss: 0.3763
Epoch 20 Training Loss: 0.3990
Epoch 20 Validation Loss: 0.3752
Epoch 21 Training Loss: 0.3975
Epoch 21 Validation Loss: 0.3735
Epoch 22 Training Loss: 0.3960
Epoch 22 Validation Loss: 0.3734
Epoch 23 Training Loss: 0.3947
Epoch 23 Validation Loss: 0.3731
Epoch 24 Training Loss: 0.3934
Epoch 24 Validation Loss: 0.3712
Epoch 25 Training Loss: 0.3929
Epoch 25 Validation Loss: 0.3718
Epoch 26 Training Loss: 0.3920
Epoch 26 Validation Loss: 0.3725
Epoch 27 Training Loss: 0.3909
Epoch 27 Validation Loss: 0.3709
Epoch 28 Training Loss: 0.3904
Epoch 28 Validation Loss: 0.3704
Epoch 29 Training Loss: 0.3900
Epoch 29 Validation Loss: 0.3700
Epoch 30 Training Loss: 0.3892
Epoch 30 Validation Loss: 0.3695
Epoch 31 Training Loss: 0.3893
Epoch 31 Validation Loss: 0.3689
Epoch 32 Training Loss: 0.3888
Epoch 32 Validation Loss: 0.3682
Epoch 33 Training Loss: 0.3881
Epoch 33 Validation Loss: 0.3680
Epoch 34 Training Loss: 0.3876
Epoch 34 Validation Loss: 0.3682
Epoch 35 Training Loss: 0.3876
Epoch 35 Validation Loss: 0.3673
Epoch 36 Training Loss: 0.3872
Epoch 36 Validation Loss: 0.3678
Epoch 37 Training Loss: 0.3866
Epoch 37 Validation Loss: 0.3663
Epoch 38 Training Loss: 0.3866
Epoch 38 Validation Loss: 0.3674
Epoch 39 Training Loss: 0.3863
Epoch 39 Validation Loss: 0.3670
Epoch 40 Training Loss: 0.3861
Epoch 40 Validation Loss: 0.3657
Epoch 41 Training Loss: 0.3862
Epoch 41 Validation Loss: 0.3662
Epoch 42 Training Loss: 0.3854
Epoch 42 Validation Loss: 0.3678
Epoch 43 Training Loss: 0.3856
Epoch 43 Validation Loss: 0.3652
Epoch 44 Training Loss: 0.3851
Epoch 44 Validation Loss: 0.3679
Epoch 45 Training Loss: 0.3854
Epoch 45 Validation Loss: 0.3643
Epoch 46 Training Loss: 0.3849
Epoch 46 Validation Loss: 0.3645
Epoch 47 Training Loss: 0.3846
Epoch 47 Validation Loss: 0.3648
Epoch 48 Training Loss: 0.3846
Epoch 48 Validation Loss: 0.3647
Epoch 49 Training Loss: 0.3845
Epoch 49 Validation Loss: 0.3634
Epoch 50 Training Loss: 0.3843
Epoch 50 Validation Loss: 0.3642
Epoch 51 Training Loss: 0.3843
Epoch 51 Validation Loss: 0.3627
Epoch 52 Training Loss: 0.3846
Epoch 52 Validation Loss: 0.3633
Epoch 53 Training Loss: 0.3843
Epoch 53 Validation Loss: 0.3640
Epoch 54 Training Loss: 0.3836
Epoch 54 Validation Loss: 0.3643
Epoch 55 Training Loss: 0.3834
Epoch 55 Validation Loss: 0.3617
Epoch 56 Training Loss: 0.3834
Epoch 56 Validation Loss: 0.3630
Epoch 57 Training Loss: 0.3837
Epoch 57 Validation Loss: 0.3626
Epoch 58 Training Loss: 0.3831
Epoch 58 Validation Loss: 0.3636
Epoch 59 Training Loss: 0.3833
Epoch 59 Validation Loss: 0.3624
Epoch 60 Training Loss: 0.3829
Epoch 60 Validation Loss: 0.3606
Epoch 61 Training Loss: 0.3832
Epoch 61 Validation Loss: 0.3604
Epoch 62 Training Loss: 0.3832
Epoch 62 Validation Loss: 0.3598
Epoch 63 Training Loss: 0.3830
Epoch 63 Validation Loss: 0.3609
Epoch 64 Training Loss: 0.3831
Epoch 64 Validation Loss: 0.3601
Epoch 65 Training Loss: 0.3831
Epoch 65 Validation Loss: 0.3613
Epoch 66 Training Loss: 0.3825
Epoch 66 Validation Loss: 0.3602
Epoch 67 Training Loss: 0.3827
Epoch 67 Validation Loss: 0.3610
Epoch 68 Training Loss: 0.3823
Epoch 68 Validation Loss: 0.3590
Epoch 69 Training Loss: 0.3820
Epoch 69 Validation Loss: 0.3604
Epoch 70 Training Loss: 0.3822
Epoch 70 Validation Loss: 0.3600
Epoch 71 Training Loss: 0.3826
Epoch 71 Validation Loss: 0.3597
Epoch 72 Training Loss: 0.3819
Epoch 72 Validation Loss: 0.3585
Epoch 73 Training Loss: 0.3817
Epoch 73 Validation Loss: 0.3598
Epoch 74 Training Loss: 0.3819
Epoch 74 Validation Loss: 0.3581
Epoch 75 Training Loss: 0.3818
Epoch 75 Validation Loss: 0.3595
Epoch 76 Training Loss: 0.3818
Epoch 76 Validation Loss: 0.3589
Epoch 77 Training Loss: 0.3822
Epoch 77 Validation Loss: 0.3591
Epoch 78 Training Loss: 0.3816
Epoch 78 Validation Loss: 0.3597
Epoch 79 Training Loss: 0.3818
Epoch 79 Validation Loss: 0.3581
Epoch 80 Training Loss: 0.3817
Epoch 80 Validation Loss: 0.3581
Epoch 81 Training Loss: 0.3819
Epoch 81 Validation Loss: 0.3580
Epoch 82 Training Loss: 0.3813
Epoch 82 Validation Loss: 0.3578
Epoch 83 Training Loss: 0.3810
Epoch 83 Validation Loss: 0.3599
Epoch 84 Training Loss: 0.3812
Epoch 84 Validation Loss: 0.3582
Epoch 85 Training Loss: 0.3810
Epoch 85 Validation Loss: 0.3594
Epoch 86 Training Loss: 0.3810
Epoch 86 Validation Loss: 0.3592
Epoch 87 Training Loss: 0.3811
Epoch 87 Validation Loss: 0.3583
Epoch 88 Training Loss: 0.3819
Epoch 88 Validation Loss: 0.3580
Epoch 89 Training Loss: 0.3814
Epoch 89 Validation Loss: 0.3587
Epoch 90 Training Loss: 0.3812
Epoch 90 Validation Loss: 0.3567
Epoch 91 Training Loss: 0.3807
Epoch 91 Validation Loss: 0.3584
Epoch 92 Training Loss: 0.3808
Epoch 92 Validation Loss: 0.3585
Epoch 93 Training Loss: 0.3808
Epoch 93 Validation Loss: 0.3573
Epoch 94 Training Loss: 0.3805
Epoch 94 Validation Loss: 0.3565
Epoch 95 Training Loss: 0.3803
Epoch 95 Validation Loss: 0.3588
Epoch 96 Training Loss: 0.3806
Epoch 96 Validation Loss: 0.3569
Epoch 97 Training Loss: 0.3807
Epoch 97 Validation Loss: 0.3576
Epoch 98 Training Loss: 0.3804
Epoch 98 Validation Loss: 0.3609
Epoch 99 Training Loss: 0.3799
Epoch 99 Validation Loss: 0.3570
```

In [ ]:

```
#%% Plot training and val loss

def plot_training(epochs, train_losses, val_losses, benchmark, title = "RNN trai
    """Plot the training and validation losses."""
    plt.plot(range(epochs), train_losses, label="Training Loss")
    plt.plot(range(epochs), val_losses, "--", label="Validation Loss")
    plt.plot([0, epochs - 1], [benchmark, benchmark], ":k", label="Benchmark")
    plt.xlabel("Epoch")
    plt.xlim([0, epochs - 1])
    plt.ylabel("Loss")
    plt.legend()
    plt.title(title)
    plt.show()

plot_training(epochs, train_losses, val_losses, benchmark, title = "Simple RNN")
```



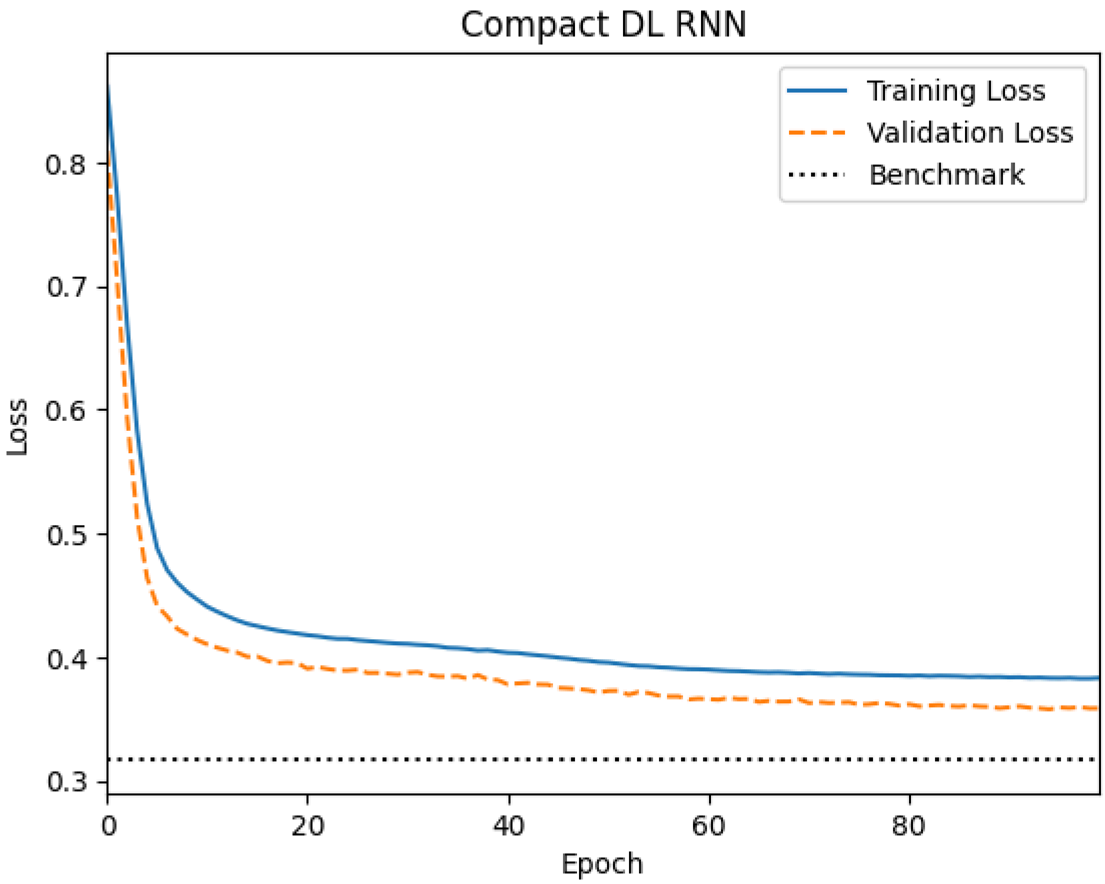
In [ ]:

```
#%% More Compact RNN

rnn_dl = dl.RecurrentModel(
    in_features=n_features,
    hidden_features=[2],
    out_features=1,
    rnn_type="RNN",
)
rnn_simple = dl.Regressor(rnn_dl, optimizer=dl.Adam(lr=0.001)).create()

print(rnn_simple)

trainer = dl.Trainer(max_epochs=epochs, accelerator="auto")
trainer.fit(rnn_simple, train_loader, val_loader)

train_losses = trainer.history.history["train_loss_epoch"]["value"]
val_losses = trainer.history.history["val_loss_epoch"]["value"][1:]
plot_training(epochs, train_losses, val_losses, benchmark, title = "Compact DL R
Regressor(
  (loss): L1Loss()
  (optimizer): Adam[Adam](lr=0.001)
  (train_metrics): MetricCollection,
    prefix=train
  )
  (val_metrics): MetricCollection,
    prefix=val
  )
  (test_metrics): MetricCollection,
    prefix=test
  )
  (model): RecurrentModel(
    (blocks): LayerList(
      (0): Sequence1dBlock(
        (layer): RNN(7, 2, batch_first=True)
      )
    )
    (head): MultiLayerPerceptron(
      (blocks): LayerList(
        (0): LinearBlock(
          (layer): Linear(in_features=2, out_features=1, bias=True)
          (activation): Identity()
        )
      )
    )
  )
)
```

```
INFO:
  | Name          | Type             | Params | Mode
-----------------------------------------------------------
0 | loss          | L1Loss           | 0      | train
1 | train_metrics | MetricCollection | 0      | train
2 | val_metrics   | MetricCollection | 0      | train
3 | test_metrics  | MetricCollection | 0      | train
4 | model         | RecurrentModel   | 25     | train
5 | optimizer     | Adam             | 0      | train
-----------------------------------------------------------
25        Trainable params
0         Non-trainable params
25        Total params
0.000     Total estimated model params size (MB)
14        Modules in train mode
0         Modules in eval mode
INFO:lightning.pytorch.callbacks.model_summary:
  | Name          | Type             | Params | Mode
-----------------------------------------------------------
0 | loss          | L1Loss           | 0      | train
1 | train_metrics | MetricCollection | 0      | train
2 | val_metrics   | MetricCollection | 0      | train
3 | test_metrics  | MetricCollection | 0      | train
4 | model         | RecurrentModel   | 25     | train
5 | optimizer     | Adam             | 0      | train
-----------------------------------------------------------
25        Trainable params
0         Non-trainable params
25        Total params
0.000     Total estimated model params size (MB)
14        Modules in train mode
0         Modules in eval mode
```

```
Sanity Checking: |          | 0/? [00:00<?, ?it/s]
c:\Users\Green\AppData\Local\Programs\Python\Python311\Lib\site-packages\lightnin
g\pytorch\trainer\connectors\data_connector.py:425: The 'val_dataloader' does not
have many workers which may be a bottleneck. Consider increasing the value of the
`num_workers` argument` to `num_workers=15` in the `DataLoader` to improve perfor
mance.
c:\Users\Green\AppData\Local\Programs\Python\Python311\Lib\site-packages\lightnin
g\pytorch\trainer\connectors\data_connector.py:425: The 'train_dataloader' does n
ot have many workers which may be a bottleneck. Consider increasing the value of
the `num_workers` argument` to `num_workers=15` in the `DataLoader` to improve pe
rformance.
Training: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
```

```
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
```

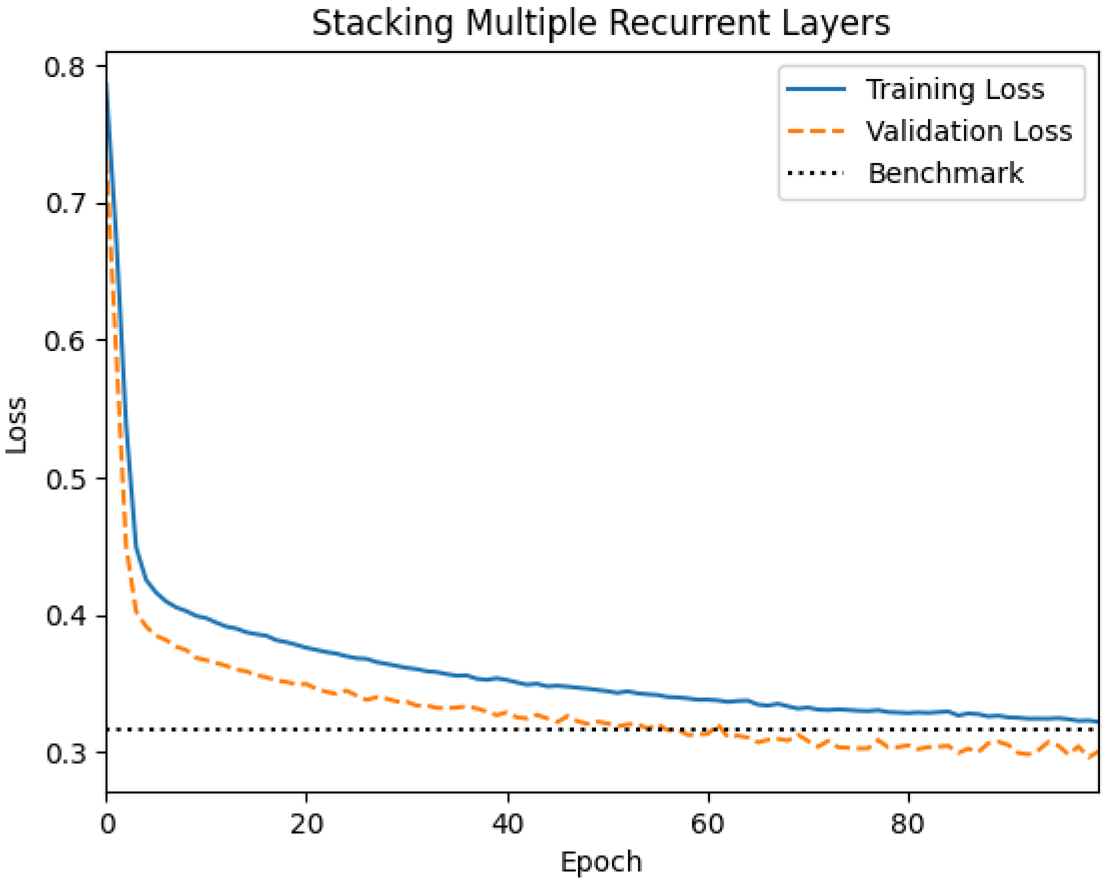


In [ ]:

```
#%% Stacking Multiple recurrent layers

rnn_dl = dl.RecurrentModel(
    in_features=n_features,
    hidden_features=[16, 16, 16],
    out_features=1,
    rnn_type="RNN",
)
rnn_stacked = dl.Regressor(rnn_dl, optimizer=dl.Adam(lr=0.0001)).create()

trainer = dl.Trainer(max_epochs=epochs, accelerator="auto")
trainer.fit(rnn_stacked, train_loader, val_loader)

train_losses = trainer.history.history["train_loss_epoch"]["value"]
val_losses = trainer.history.history["val_loss_epoch"]["value"][1:]
plot_training(epochs, train_losses, val_losses, benchmark, title = "Stacking Mul
INFO:
  | Name          | Type             | Params | Mode
-----------------------------------------------------------
0 | loss          | L1Loss           | 0      | train
1 | train_metrics | MetricCollection | 0      | train
2 | val_metrics   | MetricCollection | 0      | train
3 | test_metrics  | MetricCollection | 0      | train
4 | model         | RecurrentModel   | 1.5 K  | train
5 | optimizer     | Adam             | 0      | train
-----------------------------------------------------------
1.5 K     Trainable params
0         Non-trainable params
1.5 K     Total params
0.006     Total estimated model params size (MB)
18        Modules in train mode
0         Modules in eval mode
INFO:lightning.pytorch.callbacks.model_summary:
  | Name          | Type             | Params | Mode
-----------------------------------------------------------
0 | loss          | L1Loss           | 0      | train
1 | train_metrics | MetricCollection | 0      | train
2 | val_metrics   | MetricCollection | 0      | train
3 | test_metrics  | MetricCollection | 0      | train
4 | model         | RecurrentModel   | 1.5 K  | train
5 | optimizer     | Adam             | 0      | train
-----------------------------------------------------------
1.5 K     Trainable params
0         Non-trainable params
1.5 K     Total params
0.006     Total estimated model params size (MB)
18        Modules in train mode
0         Modules in eval mode
Sanity Checking: |          | 0/? [00:00<?, ?it/s]
Training: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
```

```
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
```

```
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
```

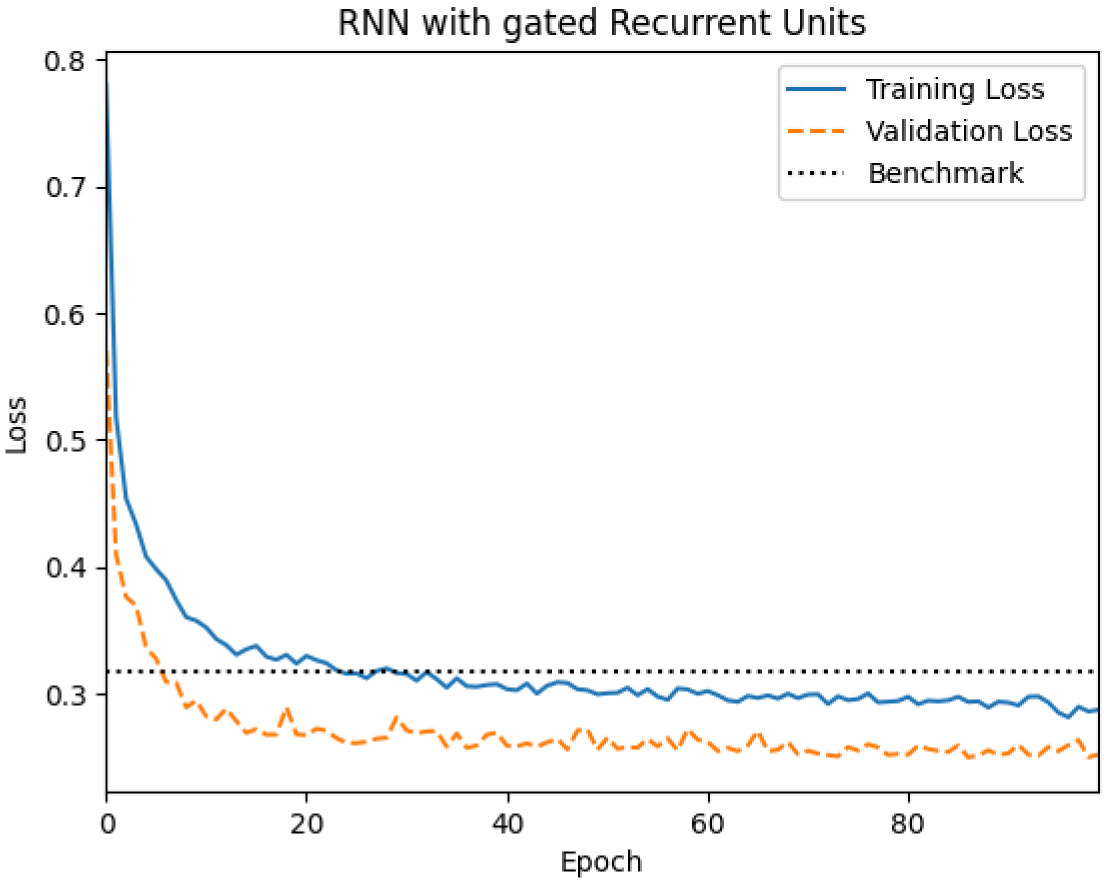


In [ ]:

```
#%% Using gated Recurrent Units

gru_dl = dl.RecurrentModel(
    in_features=n_features,
    hidden_features=[8, 8, 8],
    out_features=1,
    rnn_type="GRU",
    dropout=0.2,
)
gru_stacked = dl.Regressor(gru_dl, optimizer=dl.Adam(lr=0.001)).create()

trainer = dl.Trainer(max_epochs=epochs, accelerator="auto")
trainer.fit(gru_stacked, train_loader, val_loader)

train_losses = trainer.history.history["train_loss_epoch"]["value"]
val_losses = trainer.history.history["val_loss_epoch"]["value"][1:]
plot_training(epochs, train_losses, val_losses, benchmark, title = "RNN with gat
INFO:
  | Name          | Type              | Params | Mode
-----------------------------------------------------------
0 | loss          | L1Loss            | 0      | train
1 | train_metrics | MetricCollection  | 0      | train
2 | val_metrics   | MetricCollection  | 0      | train
3 | test_metrics  | MetricCollection  | 0      | train
4 | model         | RecurrentModel    | 1.3 K  | train
5 | optimizer     | Adam              | 0      | train
-----------------------------------------------------------
1.3 K     Trainable params
0         Non-trainable params
1.3 K     Total params
0.005     Total estimated model params size (MB)
24        Modules in train mode
0         Modules in eval mode
INFO:lightning.pytorch.callbacks.model_summary:
  | Name          | Type              | Params | Mode
-----------------------------------------------------------
0 | loss          | L1Loss            | 0      | train
1 | train_metrics | MetricCollection  | 0      | train
2 | val_metrics   | MetricCollection  | 0      | train
3 | test_metrics  | MetricCollection  | 0      | train
4 | model         | RecurrentModel    | 1.3 K  | train
5 | optimizer     | Adam              | 0      | train
-----------------------------------------------------------
1.3 K     Trainable params
0         Non-trainable params
1.3 K     Total params
0.005     Total estimated model params size (MB)
24        Modules in train mode
0         Modules in eval mode
Sanity Checking: |          | 0/? [00:00<?, ?it/s]
Training: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
```

```
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
```

```
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
```

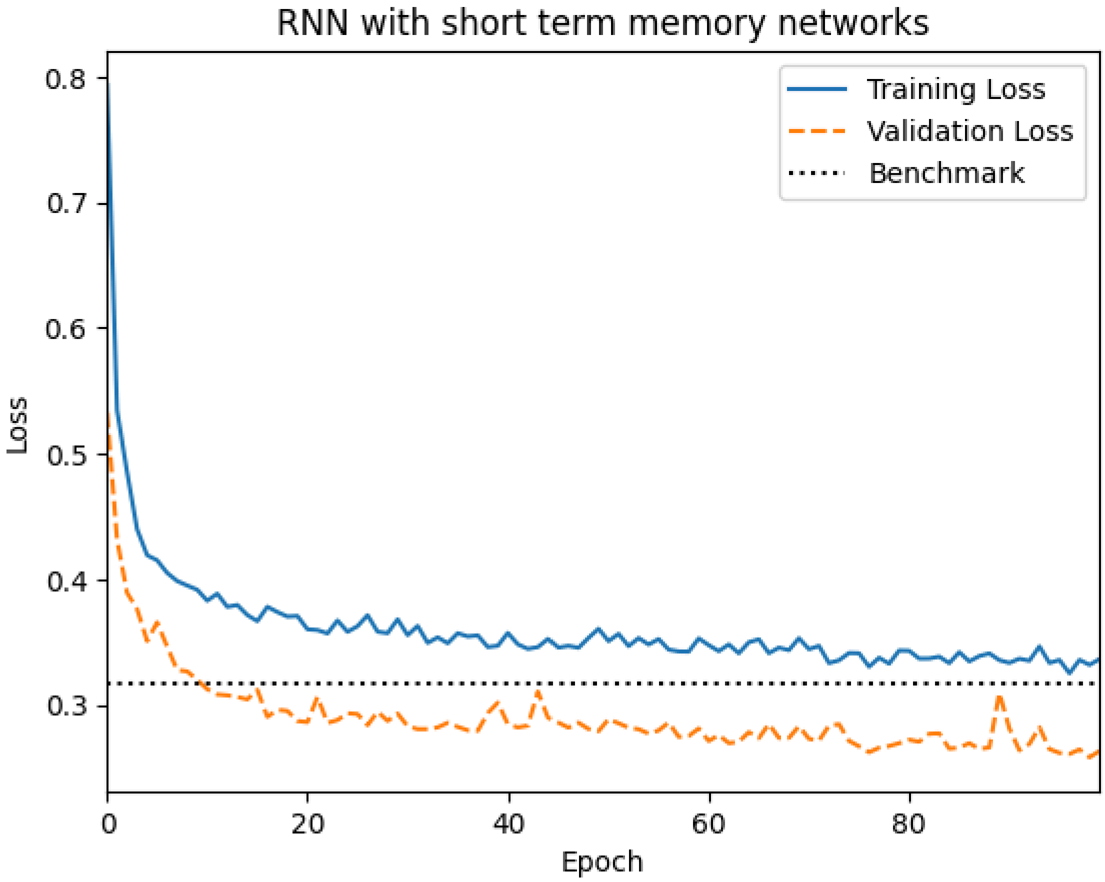


In [ ]:

```
#%% Using Short term memory networks

lstm_dl = dl.RecurrentModel(
    in_features=n_features,
    hidden_features=[8, 8, 8],
    out_features=1,
    rnn_type="LSTM",
    dropout=0.3,
)
lstm_stacked = dl.Regressor(lstm_dl, optimizer=dl.Adam(lr=0.001)).create()

trainer = dl.Trainer(max_epochs=epochs, accelerator="auto")
trainer.fit(lstm_stacked, train_loader, val_loader)

train_losses = trainer.history.history["train_loss_epoch"]["value"]
val_losses = trainer.history.history["val_loss_epoch"]["value"][1:]
plot_training(epochs, train_losses, val_losses, benchmark, title = "RNN with shc
INFO:
  | Name          | Type             | Params | Mode 
-----------------------------------------------------------
0 | loss          | L1Loss           | 0      | train
1 | train_metrics | MetricCollection | 0      | train
2 | val_metrics   | MetricCollection | 0      | train
3 | test_metrics  | MetricCollection | 0      | train
4 | model         | RecurrentModel   | 1.7 K  | train
5 | optimizer     | Adam             | 0      | train
-----------------------------------------------------------
1.7 K     Trainable params
0         Non-trainable params
1.7 K     Total params
0.007     Total estimated model params size (MB)
24        Modules in train mode
0         Modules in eval mode
INFO:lightning.pytorch.callbacks.model_summary:
  | Name          | Type             | Params | Mode 
-----------------------------------------------------------
0 | loss          | L1Loss           | 0      | train
1 | train_metrics | MetricCollection | 0      | train
2 | val_metrics   | MetricCollection | 0      | train
3 | test_metrics  | MetricCollection | 0      | train
4 | model         | RecurrentModel   | 1.7 K  | train
5 | optimizer     | Adam             | 0      | train
-----------------------------------------------------------
1.7 K     Trainable params
0         Non-trainable params
1.7 K     Total params
0.007     Total estimated model params size (MB)
24        Modules in train mode
0         Modules in eval mode
Sanity Checking: |          | 0/? [00:00<?, ?it/s]
Training: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
```

```
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
```

```
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
Validation: |          | 0/? [00:00<?, ?it/s]
```


RNN with short term memory networks
Training Loss
Validation Loss
Benchmark
Loss
Epoch