In [1]:

```python
from tqdm import tqdm
import os
import pandas as pd
import polars as pl
import numpy as np
from sklearn.svm import OneClassSVM
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import plotly.express as px
from sklearn.preprocessing import RobustScaler
from collections import Counter
from matplotlib import pyplot as plt
from sklearn.metrics import mean_absolute_percentage_error as MAPE
plt.rcParams["figure.figsize"] = (10,10)


from sklearn.decomposition import PCA
import glob

from he_svm import preprocess_a_sample, he_svm, preprocess_a_sample_encrypted
```

In [2]:

```python
errors_dfs = {}
files_with_mismatches = []

for file in glob.glob('results/Errors*.csv'):
    df = pl.read_csv(file)
    errors_dfs[file] = df
    mismatches = len(df.filter(pl.col("Correct?") == False))
    print(f'Case {file}, #mismatches: {mismatches} over {len(df)}')
    if mismatches > 0:
        files_with_mismatches.append(file)
```

```
Case results/Errors_1.1outer-100watt.csv, #mismatches: 0 over 271
Case results/Errors_0.7inner-200watt-jolm8U.csv, #mismatches: 0 over 250
Case results/Errors_1.5inner-200watt.csv, #mismatches: 0 over 291
Case results/Errors_1.3outer-300watt.csv, #mismatches: 0 over 255
Case results/Errors_1.5outer-200watt.csv, #mismatches: 0 over 251
Case results/Errors_0.7inner-100watt-67V2Iv.csv, #mismatches: 0 over 286
Case results/Errors_0.7outer-200watt-0Pp0qm.csv, #mismatches: 0 over 260
Case results/Errors_1.3outer-100watt.csv, #mismatches: 0 over 265
Case results/Errors_1.1inner-300watt.csv, #mismatches: 0 over 294
Case results/Errors_0.9inner-100watt.csv, #mismatches: 0 over 304
Case results/Errors_1.5inner-100watt.csv, #mismatches: 0 over 262
Case results/Errors_0.9inner-200watt.csv, #mismatches: 0 over 274
Case results/Errors_1.7inner-100watt.csv, #mismatches: 0 over 276
Case results/Errors_0.9inner-300watt.csv, #mismatches: 0 over 264
Case results/Errors_0.9outer-300watt.csv, #mismatches: 0 over 245
Case results/Errors_0.7outer-100watt-lB5LIS.csv, #mismatches: 0 over 260
Case results/Errors_1.3outer-200watt.csv, #mismatches: 0 over 278
Case results/Errors_0.9outer-100watt.csv, #mismatches: 0 over 281
Case results/Errors_1.7outer-300watt.csv, #mismatches: 0 over 274
Case results/Errors_1.5outer-300watt.csv, #mismatches: 0 over 270
Case results/Errors_1.3inner-300watt.csv, #mismatches: 0 over 255
Case results/Errors_0.9outer-200watt.csv, #mismatches: 0 over 281
Case results/Errors_0.7outer-300watt-PzsIeS.csv, #mismatches: 0 over 132
Case results/Errors_0.7inner-300watt-Zo8w7U.csv, #mismatches: 0 over 227
Case results/Errors_1.7inner-200watt.csv, #mismatches: 0 over 275
Case results/Errors_1.5inner-300watt.csv, #mismatches: 0 over 267
Case results/Errors_1.1outer-300watt.csv, #mismatches: 0 over 278
Case results/Errors_1.3inner-100watt.csv, #mismatches: 0 over 286
Case results/Errors_1.1inner-200watt.csv, #mismatches: 0 over 274
Case results/Errors_1.5outer-100watt.csv, #mismatches: 0 over 263
Case results/Errors_1.7outer-200watt.csv, #mismatches: 0 over 261
Case results/Errors_1.1inner-100watt.csv, #mismatches: 0 over 272
Case results/Errors_1.7outer-100watt.csv, #mismatches: 0 over 272
Case results/Errors_1.7inner-300watt.csv, #mismatches: 0 over 257
Case results/Errors_1.3inner-200watt.csv, #mismatches: 0 over 266
Case results/Errors_1.1outer-200watt.csv, #mismatches: 0 over 303
```

In [3]:
```
if len(files_with_mismatches) == 0:
    print("No mismatches! The processing is equal between encrypted and plain.")
```

```
No mismatches! The processing is equal between encrypted and plain.
```

In [4]:
```
sum([len(pd.read_csv(f)) for f in glob.glob('results/Errors*.csv')])
```

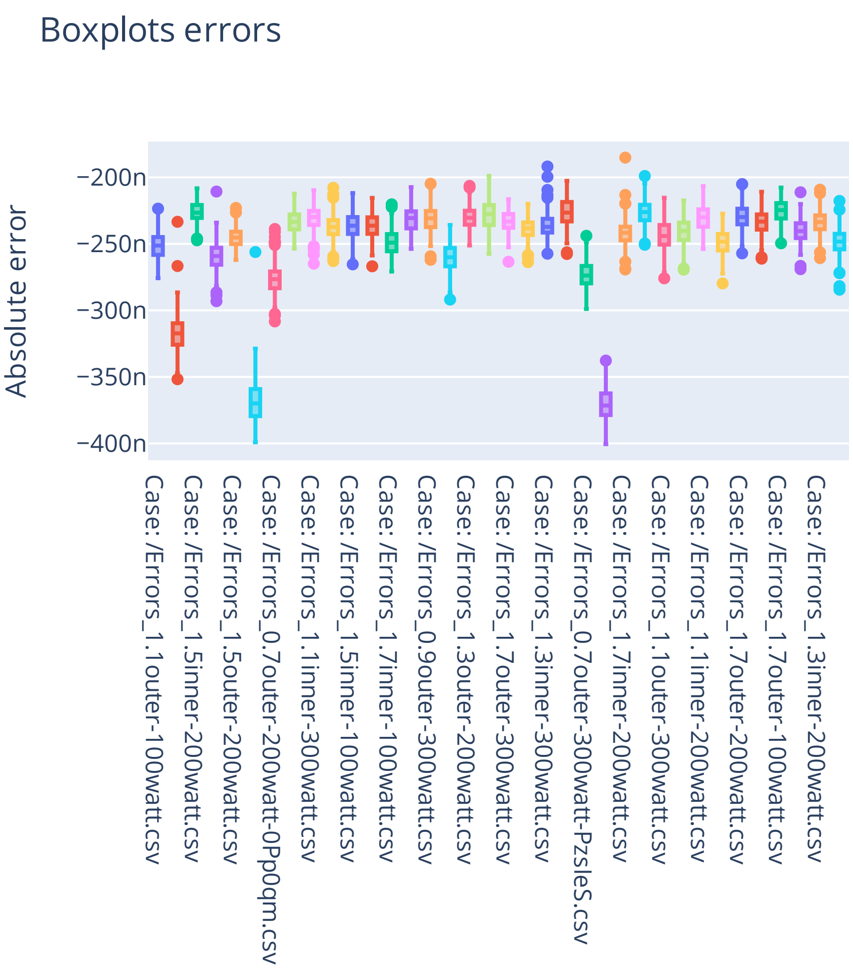
Out[4]:
```
9580
```

In [5]:
```
for file in files_with_mismatches:
    print(errors_dfs[file].filter(pl.col("Correct?") == False).write_csv())
```

In [6]:
```
fig = go.Figure()

for file in glob.glob('results/Errors*.csv'):
    df = errors_dfs[file]
    fig.add_trace(go.Box(y=df.select(pl.col('Expected') - pl.col('Predicted (enc)')
                         name=f'Case: {file[7:]}'))


fig.update_layout(title_text=f"Boxplots errors", showlegend=False)
fig.update_yaxes(title_text='Absolute error')
fig.show()
```



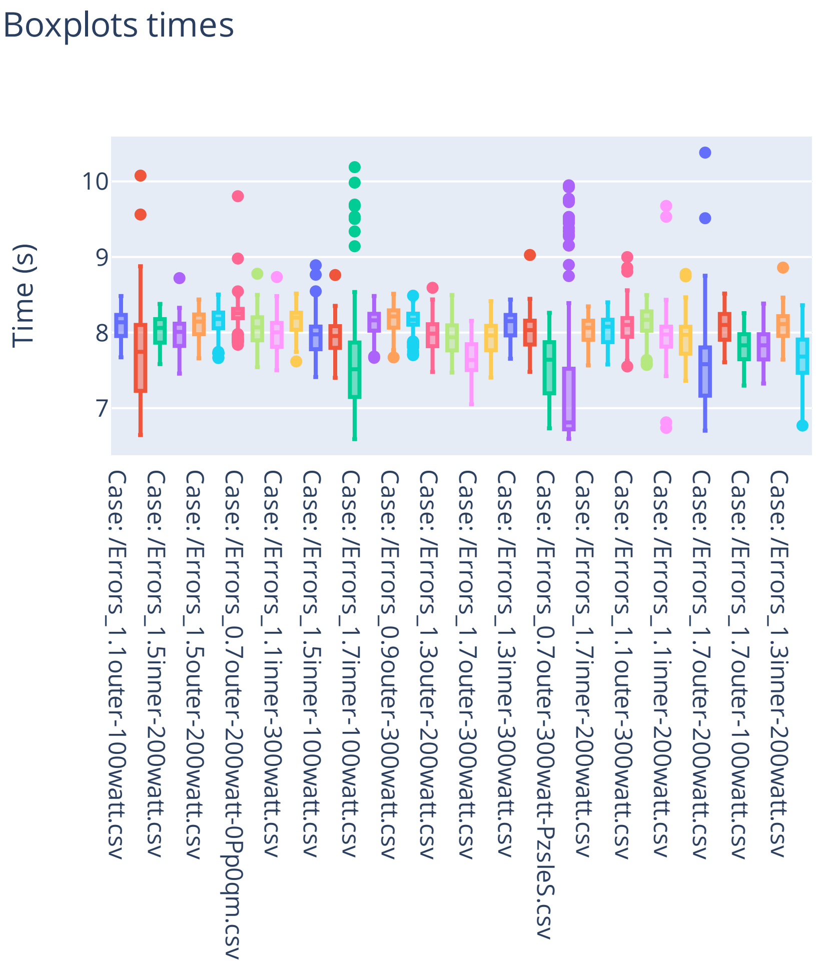
In [ ]:

In [7]:

```python
fig = go.Figure()

for file in glob.glob('results/Errors*.csv'):
    df = errors_dfs[file]
    fig.add_trace(go.Box(y=df.select(pl.col('Time enc (s)')).to_numpy().flatten(),
                         name=f'Case: {file[7:]}'))


fig.update_layout(title_text=f"Boxplots times", showlegend=False)
fig.update_yaxes(title_text='Time (s)')
fig.show()
```



In [8]:

```python
times = np.array([])
for file in glob.glob('results/Errors*.csv'):
    df = errors_dfs[file]
    times = np.append(times, df.select(pl.col('Time enc (s)'))[:, 0].to_numpy())
```

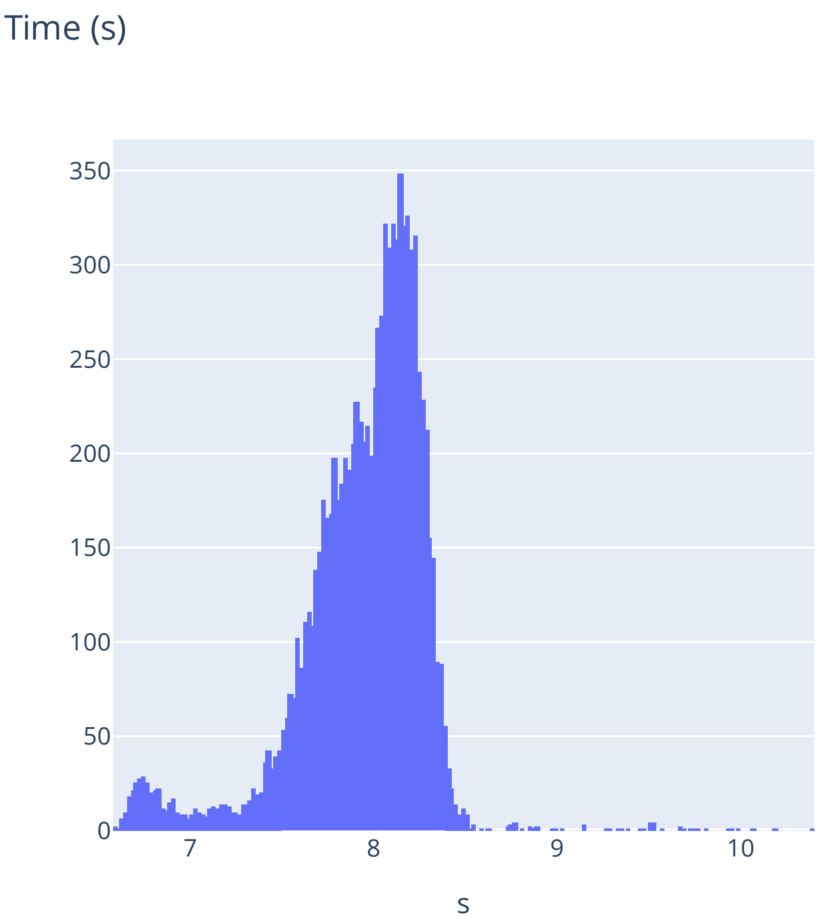
In [9]:

```python
fig = go.Figure()

fig.add_trace(go.Histogram(x=times))

fig.update_layout(
        title_text='Time (s)',
        xaxis_title_text='s', # xaxis label
        yaxis_title_text='', # yaxis label
)

fig.show()
```



In [10]:

```
for value in times:
    print(value)
    break
```

```
8.060501570347697
```

In [11]:

```
try:
    os.remove('times.csv')
except FileNotFoundError:
    pass

with open('times.csv', 'a') as f: # Open a file in append mode
    f.write('dist\n')
    for value in times:
        f.write(f"{value}\n")
```

`times.csv` is used to make the histogram in the paper (if the plot looks a bit different from the one in the paper, it may be because we re-run the same experiments more times. However, the times are very similar).

In [ ]:

```
```