QVM_verification_two_qubits

January 3, 2025

[48]:
```
!pip install graphviz
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting graphviz
  Downloading graphviz-0.20.3-py3-none-any.whl (47 kB)

47.1/47.1 KB 159.8 kB/s eta 0:00:00 MB/s eta
0:00:01
Installing collected packages: graphviz
Successfully installed graphviz-0.20.3
```

[1]:
```
from graphviz import Digraph

# Crear el grafo dirigido
g = Digraph('Workflow', format='png', node_attr={'shape': 'box', 'style':␣
 ↪'rounded,filled', 'fillcolor': 'lightblue', 'fontsize': '17'},␣
 ↪edge_attr={'fontsize': '10'})

# Inicio
g.node('A', 'Start', fillcolor='lightgreen')
g.edge('A', 'A', label='<<b>GitHub repository for reproducibility</b>>',␣
 ↪style='', color='black',fontsize='16', penwidth='2')
# Paso A: Feature Generation
g.node('B', '''Step A: Feature Generation
- /IMAGES/ with subfolders
- A1: Image Visualization
- A2: Extraction of 13 Features
- CSV generation in /FEATURE_RESULTS/
- Binary concatenation of CSV files''', fillcolor='#FF99FF')  # Nodo fusionado
g.edge('A', 'B')
g.edge('B', 'B', label='<<b>Binary_features_generation.ipynb</b>>', style='',␣
 ↪color='black',fontsize='16',penwidth='2')

# Paso B: Optimization of QVM Model Parameters
g.node('H', '''Step B: Optimization of QVM Model Parameters
- Import libraries (qiskit, etc.)
- Define functions: ZZFeatureMap
```

```
- Load and normalize data
- Define circuit and cost function
- Optimization with COBYLA''', fillcolor='#FF99FF')  # Nodo fusionado
g.edge('B', 'H')
g.edge('H', 'H', label='<<b>FIT_DP_NODP_CIRCUIT.ipynb</b>>', style='',␣
 ↪color='black',penwidth='2',fontsize='16')

# Paso C: Verification on Real Quantum Hardware or Simulation
g.node('R', '''Step C: Verification on Quantum Hardware/Simulation
- Environment setup
- Backend selection
- Load data and optimized parameters
- Transpile and execute circuits
- Results analysis and accuracy''', fillcolor='#FF99FF')  # Nodo fusionado
g.edge('H', 'R')
g.edge('R', 'R', label='<<b>QVM_verification_two_qubits.ipynb</b>>',␣
 ↪color='black',fontsize='16',penwidth='2')

# Paso D: Execution of QKM Model
g.node('ZA', '''Step D: Execution of QKM Model
- Real quantum computer execution
- Estimation Quantum Kernel''', fillcolor='#FF99FF')  # Nodo fusionado
g.edge('R', 'ZA')
g.edge('ZA', 'ZA', label='<<b>Git_Hub.</b>>', style='',␣
 ↪color='black',fontsize='16',penwidth='2')
# Final
g.node('END', 'End', fillcolor='lightgreen')
g.edge('ZA', 'END')

# Renderizar y mostrar el gráfico
g.render('workflow_diagram_compact', view=True)
```

[1]:
```
'workflow_diagram_compact.png'
```

[21]:
```
!pip install kroki
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting kroki
  Downloading kroki-0.1.2.tar.gz (8.8 kB)
  Preparing metadata (setup.py) … done
Requirement already satisfied: IPython in
/home/josemiguel/.local/lib/python3.10/site-packages (from kroki) (8.26.0)
Collecting i2
  Downloading i2-0.1.45-py3-none-any.whl (202 kB)

202.8/202.8 KB 310.7 kB/s eta 0:00:001m400.3 kB/s
eta 0:00:01
```

```
Requirement already satisfied: requests in
/home/josemiguel/.local/lib/python3.10/site-packages (from kroki) (2.32.3)
Requirement already satisfied: pygments>=2.4.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from IPython->kroki)
(2.18.0)
Requirement already satisfied: stack-data in
/home/josemiguel/.local/lib/python3.10/site-packages (from IPython->kroki)
(0.6.3)
Requirement already satisfied: exceptiongroup in
/home/josemiguel/.local/lib/python3.10/site-packages (from IPython->kroki)
(1.2.2)
Requirement already satisfied: matplotlib-inline in
/home/josemiguel/.local/lib/python3.10/site-packages (from IPython->kroki)
(0.1.7)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in
/home/josemiguel/.local/lib/python3.10/site-packages (from IPython->kroki)
(3.0.47)
Requirement already satisfied: typing-extensions>=4.6 in
/home/josemiguel/.local/lib/python3.10/site-packages (from IPython->kroki)
(4.12.2)
Requirement already satisfied: traitlets>=5.13.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from IPython->kroki)
(5.14.3)
Requirement already satisfied: pexpect>4.3 in /usr/lib/python3/dist-packages
(from IPython->kroki) (4.8.0)
Requirement already satisfied: jedi>=0.16 in
/home/josemiguel/.local/lib/python3.10/site-packages (from IPython->kroki)
(0.19.1)
Requirement already satisfied: decorator in
/home/josemiguel/.local/lib/python3.10/site-packages (from IPython->kroki)
(5.1.1)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/josemiguel/.local/lib/python3.10/site-packages (from requests->kroki)
(2.2.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/josemiguel/.local/lib/python3.10/site-packages (from requests->kroki)
(3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/lib/python3/dist-packages
(from requests->kroki) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python3/dist-
packages (from requests->kroki) (2020.6.20)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
jedi>=0.16->IPython->kroki) (0.8.4)
Requirement already satisfied: wcwidth in
/home/josemiguel/.local/lib/python3.10/site-packages (from prompt-
toolkit<3.1.0,>=3.0.41->IPython->kroki) (0.2.13)
Requirement already satisfied: asttokens>=2.1.0 in
```

```
/home/josemiguel/.local/lib/python3.10/site-packages (from stack-
data->IPython->kroki) (2.4.1)
Requirement already satisfied: executing>=1.2.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from stack-
data->IPython->kroki) (2.0.1)
Requirement already satisfied: pure-eval in
/home/josemiguel/.local/lib/python3.10/site-packages (from stack-
data->IPython->kroki) (0.2.3)
Requirement already satisfied: six>=1.12.0 in /usr/lib/python3/dist-packages
(from asttokens>=2.1.0->stack-data->IPython->kroki) (1.16.0)
Building wheels for collected packages: kroki
  Building wheel for kroki (setup.py) … done
  Created wheel for kroki: filename=kroki-0.1.2-py3-none-any.whl size=9066
sha256=e702d541090f136e332e8d8222e5d2de9e16c7a00f0de3031d89b8285dcb7e40
  Stored in directory: /home/josemiguel/.cache/pip/wheels/5e/31/e5/3be72bc7a0710
c313efdfdf6b5616598741bd49ef9735b435b
Successfully built kroki
Installing collected packages: i2, kroki
Successfully installed i2-0.1.45 kroki-0.1.2
```

[ ]:
```
#https://www.fisicacuantica.es/el-entrelazamiento-cuantico/
```

[5]:
```
!pip install qiskit-ibm-runtime
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: qiskit-ibm-runtime in
/home/josemiguel/.local/lib/python3.10/site-packages (0.27.0)
Requirement already satisfied: python-dateutil>=2.8.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit-ibm-runtime)
(2.9.0.post0)
Requirement already satisfied: websocket-client>=1.5.1 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit-ibm-runtime)
(1.8.0)
Requirement already satisfied: urllib3>=1.21.1 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit-ibm-runtime)
(2.2.2)
Requirement already satisfied: pydantic>=2.5.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit-ibm-runtime)
(2.8.2)
Requirement already satisfied: ibm-platform-services>=0.22.6 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit-ibm-runtime)
(0.55.3)
Requirement already satisfied: requests>=2.19 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit-ibm-runtime)
(2.32.3)
Requirement already satisfied: numpy>=1.13 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit-ibm-runtime)
(1.26.4)
```

```
Requirement already satisfied: qiskit>=1.1.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit-ibm-runtime)
(1.1.2)
Requirement already satisfied: requests-ntlm>=1.1.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit-ibm-runtime)
(1.3.0)
Requirement already satisfied: ibm-cloud-sdk-core<4.0.0,>=3.20.6 in
/home/josemiguel/.local/lib/python3.10/site-packages (from ibm-platform-
services>=0.22.6->qiskit-ibm-runtime) (3.20.6)
Requirement already satisfied: typing-extensions>=4.6.1 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
pydantic>=2.5.0->qiskit-ibm-runtime) (4.12.2)
Requirement already satisfied: annotated-types>=0.4.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
pydantic>=2.5.0->qiskit-ibm-runtime) (0.7.0)
Requirement already satisfied: pydantic-core==2.20.1 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
pydantic>=2.5.0->qiskit-ibm-runtime) (2.20.1)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from
python-dateutil>=2.8.0->qiskit-ibm-runtime) (1.16.0)
Requirement already satisfied: scipy>=1.5 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=1.1.0->qiskit-ibm-runtime) (1.14.0)
Requirement already satisfied: dill>=0.3 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=1.1.0->qiskit-ibm-runtime) (0.3.8)
Requirement already satisfied: sympy>=1.3 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=1.1.0->qiskit-ibm-runtime) (1.13.1)
Requirement already satisfied: rustworkx>=0.14.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=1.1.0->qiskit-ibm-runtime) (0.15.1)
Requirement already satisfied: stevedore>=3.0.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=1.1.0->qiskit-ibm-runtime) (5.2.0)
Requirement already satisfied: symengine>=0.11 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=1.1.0->qiskit-ibm-runtime) (0.11.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
requests>=2.19->qiskit-ibm-runtime) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/lib/python3/dist-packages
(from requests>=2.19->qiskit-ibm-runtime) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python3/dist-
packages (from requests>=2.19->qiskit-ibm-runtime) (2020.6.20)
Requirement already satisfied: pyspnego>=0.4.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from requests-
ntlm>=1.1.0->qiskit-ibm-runtime) (0.11.1)
```

```
Requirement already satisfied: cryptography>=1.3 in /usr/lib/python3/dist-
packages (from requests-ntlm>=1.1.0->qiskit-ibm-runtime) (3.4.8)
Requirement already satisfied: PyJWT<3.0.0,>=2.8.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from ibm-cloud-sdk-
core<4.0.0,>=3.20.6->ibm-platform-services>=0.22.6->qiskit-ibm-runtime) (2.9.0)
Requirement already satisfied: pbr!=2.1.0,>=2.0.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
stevedore>=3.0.0->qiskit>=1.1.0->qiskit-ibm-runtime) (6.0.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
sympy>=1.3->qiskit>=1.1.0->qiskit-ibm-runtime) (1.3.0)
```

[ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[6]:
```
!pip install qiskit
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: qiskit in
/home/josemiguel/.local/lib/python3.10/site-packages (1.1.2)
Requirement already satisfied: dill>=0.3 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit) (0.3.8)
Requirement already satisfied: sympy>=1.3 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit) (1.13.1)
Requirement already satisfied: numpy<3,>=1.17 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit) (1.26.4)
Requirement already satisfied: scipy>=1.5 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit) (1.14.0)
Requirement already satisfied: typing-extensions in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit) (4.12.2)
Requirement already satisfied: symengine>=0.11 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit) (0.11.0)
Requirement already satisfied: python-dateutil>=2.8.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit) (2.9.0.post0)
Requirement already satisfied: rustworkx>=0.14.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit) (0.15.1)
Requirement already satisfied: stevedore>=3.0.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit) (5.2.0)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from
python-dateutil>=2.8.0->qiskit) (1.16.0)
Requirement already satisfied: pbr!=2.1.0,>=2.0.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
stevedore>=3.0.0->qiskit) (6.0.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from sympy>=1.3->qiskit)
(1.3.0)
```

[8]:
```
!pip install qiskit_algorithms
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: qiskit_algorithms in
/home/josemiguel/.local/lib/python3.10/site-packages (0.3.0)
Requirement already satisfied: qiskit>=0.44 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit_algorithms)
(1.1.2)
Requirement already satisfied: scipy>=1.4 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit_algorithms)
(1.14.0)
Requirement already satisfied: numpy>=1.17 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit_algorithms)
(1.26.4)
Requirement already satisfied: stevedore>=3.0.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=0.44->qiskit_algorithms) (5.2.0)
Requirement already satisfied: symengine>=0.11 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=0.44->qiskit_algorithms) (0.11.0)
Requirement already satisfied: python-dateutil>=2.8.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=0.44->qiskit_algorithms) (2.9.0.post0)
Requirement already satisfied: dill>=0.3 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=0.44->qiskit_algorithms) (0.3.8)
Requirement already satisfied: rustworkx>=0.14.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=0.44->qiskit_algorithms) (0.15.1)
Requirement already satisfied: sympy>=1.3 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=0.44->qiskit_algorithms) (1.13.1)
Requirement already satisfied: typing-extensions in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=0.44->qiskit_algorithms) (4.12.2)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from
python-dateutil>=2.8.0->qiskit>=0.44->qiskit_algorithms) (1.16.0)
Requirement already satisfied: pbr!=2.1.0,>=2.0.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
stevedore>=3.0.0->qiskit>=0.44->qiskit_algorithms) (6.0.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
sympy>=1.3->qiskit>=0.44->qiskit_algorithms) (1.3.0)
```

[4]:
```
!pip install qiskit-aer
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: qiskit-aer in
```

```
/home/josemiguel/.local/lib/python3.10/site-packages (0.14.2)
Requirement already satisfied: psutil>=5 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit-aer) (6.0.0)
Requirement already satisfied: scipy>=1.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit-aer) (1.14.0)
Requirement already satisfied: qiskit>=0.45.2 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit-aer) (1.1.2)
Requirement already satisfied: numpy>=1.16.3 in
/home/josemiguel/.local/lib/python3.10/site-packages (from qiskit-aer) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=0.45.2->qiskit-aer) (2.9.0.post0)
Requirement already satisfied: typing-extensions in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=0.45.2->qiskit-aer) (4.12.2)
Requirement already satisfied: rustworkx>=0.14.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=0.45.2->qiskit-aer) (0.15.1)
Requirement already satisfied: stevedore>=3.0.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=0.45.2->qiskit-aer) (5.2.0)
Requirement already satisfied: symengine>=0.11 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=0.45.2->qiskit-aer) (0.11.0)
Requirement already satisfied: sympy>=1.3 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=0.45.2->qiskit-aer) (1.13.1)
Requirement already satisfied: dill>=0.3 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
qiskit>=0.45.2->qiskit-aer) (0.3.8)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from
python-dateutil>=2.8.0->qiskit>=0.45.2->qiskit-aer) (1.16.0)
Requirement already satisfied: pbr!=2.1.0,>=2.0.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
stevedore>=3.0.0->qiskit>=0.45.2->qiskit-aer) (6.0.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from
sympy>=1.3->qiskit>=0.45.2->qiskit-aer) (1.3.0)
```

[9]:
```
!pip install pylatexenc
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pylatexenc in
/home/josemiguel/.local/lib/python3.10/site-packages (2.10)
```

[7]:
```
from qiskit_aer import AerSimulator,Aer# ojo son distintos AerSimulator y Aer
↪AerSimulator es para entrar en estadisticas de IBM
```

```
import pennylane as qml
from pennylane import numpy as np
import matplotlib.pyplot as plt
from time import time
from sklearn import svm
import scipy
```

[6]:
```
get_ipython().system('pip install pennylane')
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pennylane in
/home/josemiguel/.local/lib/python3.10/site-packages (0.37.0)
Requirement already satisfied: cachetools in
/home/josemiguel/.local/lib/python3.10/site-packages (from pennylane) (5.5.0)
Requirement already satisfied: semantic-version>=2.7 in
/home/josemiguel/.local/lib/python3.10/site-packages (from pennylane) (2.10.0)
Requirement already satisfied: appdirs in
/home/josemiguel/.local/lib/python3.10/site-packages (from pennylane) (1.4.4)
Requirement already satisfied: networkx in
/home/josemiguel/.local/lib/python3.10/site-packages (from pennylane) (3.3)
Requirement already satisfied: numpy<2.0 in
/home/josemiguel/.local/lib/python3.10/site-packages (from pennylane) (1.26.4)
Requirement already satisfied: requests in
/home/josemiguel/.local/lib/python3.10/site-packages (from pennylane) (2.32.3)
Requirement already satisfied: packaging in
/home/josemiguel/.local/lib/python3.10/site-packages (from pennylane) (24.1)
Requirement already satisfied: scipy in
/home/josemiguel/.local/lib/python3.10/site-packages (from pennylane) (1.14.0)
Requirement already satisfied: toml in
/home/josemiguel/.local/lib/python3.10/site-packages (from pennylane) (0.10.2)
Requirement already satisfied: autograd in
/home/josemiguel/.local/lib/python3.10/site-packages (from pennylane) (1.7.0)
Requirement already satisfied: pennylane-lightning>=0.37 in
/home/josemiguel/.local/lib/python3.10/site-packages (from pennylane) (0.37.0)
Requirement already satisfied: typing-extensions in
/home/josemiguel/.local/lib/python3.10/site-packages (from pennylane) (4.12.2)
Requirement already satisfied: rustworkx in
/home/josemiguel/.local/lib/python3.10/site-packages (from pennylane) (0.15.1)
Requirement already satisfied: autoray>=0.6.11 in
/home/josemiguel/.local/lib/python3.10/site-packages (from pennylane) (0.6.12)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/josemiguel/.local/lib/python3.10/site-packages (from requests->pennylane)
(2.2.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/josemiguel/.local/lib/python3.10/site-packages (from requests->pennylane)
(3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/lib/python3/dist-packages
```

```
(from requests->pennylane) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python3/dist-
packages (from requests->pennylane) (2020.6.20)
```

[44]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle

import warnings
warnings.filterwarnings("ignore")
# constants
n = 2
RANDOM_STATE = 42
LR = 1e-3
class_labels = ['0', '1']

def normalizeData(DATA_PATH = "./FEATURE_RESULTS/FEATURE_resultante_DP_NODP.
 ↪csv"):
    """
    Normalizes the data
    """
    # Reads the data
    data = pd.read_csv(DATA_PATH)
    data = shuffle(data, random_state=RANDOM_STATE)
    X, Y = data[['area_pixels', ' mean_coords_x']].values, data[' class'].values

    # normalize the data
    scaler = MinMaxScaler(feature_range=(-0 * np.pi, 2 * np.pi))
    X = scaler.fit_transform(X)
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,␣
 ↪random_state=RANDOM_STATE)
    return X_train, X_test, Y_train, Y_test


# In[5]:


import qiskit_algorithms
from qiskit_algorithms.optimizers import SPSA

from qiskit import QuantumCircuit

from qiskit.circuit.library import ZZFeatureMap, RealAmplitudes
```

```
from qiskit.quantum_info import Statevector

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split

from sklearn.utils import shuffle

import warnings
warnings.filterwarnings("ignore")

TRAIN_DATA, TEST_DATA, TRAIN_LABELS, TEST_LABELS = normalizeData()
# Replace all occurrences of 2 with 0
TRAIN_LABELS = np.where(TRAIN_LABELS == 2, 0, TRAIN_LABELS)
TEST_LABELS = np.where(TEST_LABELS == 2, 0, TEST_LABELS)
#print(TRAIN_DATA)
```

[9]:
```
# OJO OJO OJO



from qiskit_aer import AerSimulator,Aer # ojo son distintos AerSimulator y Aer␣
 ↪AerSimulator es para entrar en estadisticas de IBM
#import pennylane as qml
#from pennylane import numpy as np
import matplotlib.pyplot as plt
from time import time
from sklearn import svm
import scipy
```

[10]:
```
TRAIN_LABELS
```

[10]:
```
array([1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0,
       1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0,
       1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1,
       0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1,
       1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0,
       0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0,
       1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1,
       0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,
       0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1,
```

```
       0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1,
       0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1,
       1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0,
       1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0,
       1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0,
       0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0,
       1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0,
       1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1,
       0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0])
```

[11]:
```
TEST_LABELS
```

[11]:
```
array([1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1,
       0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1,
       1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1,
       0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1,
       0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1,
       1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0,
       1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1])
```

[45]:
```
print(TEST_LABELS.shape)
```

```
(174,)
```

[13]:
```
#Or, optionally use the save_account() method to save your credentials for easy␣
 ↪access later on, before initializing the service.
from qiskit_ibm_runtime import QiskitRuntimeService

# Save an IBM Quantum account and set it as your default account.



# Load saved credentials
```

```
QiskitRuntimeService.save_account(channel="ibm_quantum", token="......
 ↪f8a50f104a0ea070104db339580e2611293f672831a5b......", overwrite=True,␣
 ↪set_as_default=True)



#to run on a real quantum computer

service = QiskitRuntimeService()
backend = service.least_busy(operational=True, simulator=False)
backend.name
```

[13]:
```
'ibm_brisbane'
```

[14]:
```
#Or, optionally use the save_account() method to save your credentials for easy␣
 ↪access later on, before initializing the service.
from qiskit_ibm_runtime import QiskitRuntimeService

# Save an IBM Quantum account and set it as your default account.

# Save an IBM Quantum account and set it as your default account.
QiskitRuntimeService.save_account(channel="ibm_quantum", token="..........
 ↪57f8a50f104a0ea070104db339..........", overwrite=True, set_as_default=True)

service = QiskitRuntimeService()
#backend = service.least_busy(operational=True, simulator=False)
#backend.name


from qiskit_ibm_runtime.fake_provider import␣
 ↪FakeManilaV2,FakeBrisbane,FakeKyoto,FakeOsaka

# If you want simulation with updated errors from real computer

backend =FakeOsaka()


from qiskit.circuit import QuantumCircuit
from qiskit.transpiler.preset_passmanagers import generate_preset_pass_manager
from qiskit_ibm_runtime import SamplerV2 as Sampler
from qiskit_ibm_runtime.fake_provider import FakeManilaV2
```

[ ]:
```
from qiskit_ibm_runtime.fake_provider import␣
 ↪FakeManilaV2,FakeBrisbane,FakeKyoto,FakeOsaka
backend =FakeOsaka()
```

```
from qiskit.circuit import QuantumCircuit
from qiskit.transpiler.preset_passmanagers import generate_preset_pass_manager
from qiskit_ibm_runtime import SamplerV2 as Sampler
from qiskit_ibm_runtime.fake_provider import FakeManilaV2
```

[15]:
```
backend.name
```

[15]:
```
'fake_osaka'
```

[17]:
```
#variable to store all transpiled circuits
counts= []
circuits_a = []
circuits_aa = []

#function to obtain the all transpiled circuit
def VARIA_circuitos_DP_NODP(data, opt_var):
    """
    """
    #backend = Aer.get_backend('statevector_simulator')
    #feature_map = ZZFeatureMap_11_parametros_MODI(feature_dimension=2, reps=1,␣
 ↪x=data,teta=opt_var)
    feature_map=ZZFeatureMap_10_parametros(feature_dimension=2, reps=1,␣
 ↪theta_param=2, x=data, teta=opt_var)
    #backend = FakeOsaka()
    passmanager = generate_preset_pass_manager(optimization_level=3,␣
 ↪backend=backend)

    transpiled_circuit = passmanager.run(feature_map)
    circuits_aa.append(transpiled_circuit)



    return 1
```

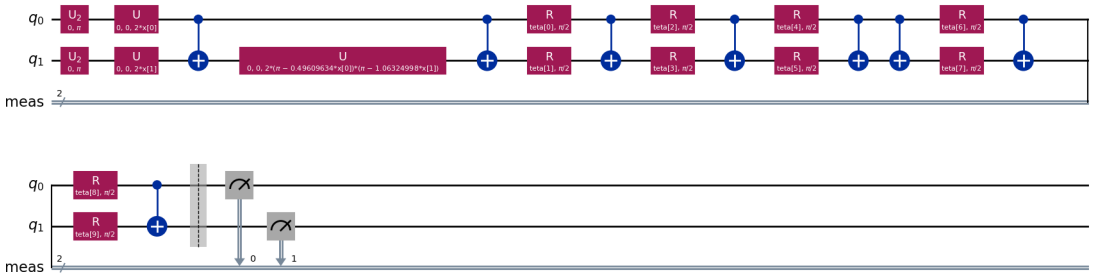
[58]:
```
import numpy as np
from qiskit.circuit import ParameterVector
from qiskit import QuantumCircuit
import numpy as np

from qiskit import QuantumCircuit

"""

"""

x = ParameterVector('x', 2)
teta=ParameterVector('teta',10)
```

```
# define your parameters
def ZZFeatureMap_10_parametros(feature_dimension=2, reps=1, theta_param=np.pi/
 ↪2, x=x, teta=teta):
    circuit = QuantumCircuit(feature_dimension)
#1.06324998,  0.49609634
    for i in range(reps):
        if i == 0:
            circuit.h(range(feature_dimension))

        for j in range(feature_dimension):
            circuit.p(theta_param * x[j] * 1, j)

        circuit.cx(0, 1)

        circuit.p(theta_param * (np.pi - x[0] *0.49609634) * (np.pi - x[1] *1.
 ↪06324998), 1)

        circuit.cx(0, 1)

        # Ensure that the indices are within the range of feature_dimension
        if feature_dimension > 1:
            circuit.ry(teta[0], 0)
            circuit.ry(teta[1], 1)

        if feature_dimension > 1:
            circuit.cx(0, 1)
            circuit.ry(teta[2], 0)
            circuit.ry(teta[3], 1)
        if feature_dimension > 1:
            circuit.cx(0, 1)
            circuit.ry(teta[4], 0)
            circuit.ry(teta[5], 1)
            circuit.cx(0, 1)
        if feature_dimension > 1:
            circuit.cx(0, 1)
            circuit.ry(teta[6], 0)
            circuit.ry(teta[7], 1)
            circuit.cx(0, 1)
            circuit.ry(teta[8], 0)
            circuit.ry(teta[9], 1)
            circuit.cx(0, 1)
            circuit.measure_all()

    return circuit

feature_map = ZZFeatureMap_10_parametros(feature_dimension=2, reps=1,␣
 ↪theta_param=2, x=x,teta=teta)
```

```
#   OPTIMUM VALUES FOR:

# DP_BREAK

opt_var=np.array([-0.45369495, 3.01307608, 0.34955593, -0.42861045, -3.038998 ,␣
 ↪-5.66827399, 1.69208433,
                 6.0084911 , 5.83775323, 2.25658061, 1.55460149])


# DP_NODP

opt_var=np.array([-1.36465941,  0.72901008,  0.46274449, -0.22550087,  0.
 ↪71628267,
       -5.0369175 ,  0.25267942,  3.20192607,  2.22427876,  2.87675972])


counts= []
circuits_a = []
circuits_aa = []

# 136 circuits random circuits for this combination PD_NOPD

n_iterations =136

for i in range(n_iterations):
    # Get the ith element of training data
    #train_data_element = TRAIN_DATA[i]
    train_data_element =TRAIN_DATA[i]
    # Call the function with the current training data element and opt_var
    VARIA_circuitos_DP_NODP(train_data_element, opt_var)
```

[56]:
```
len(TEST_DATA)

feature_map = ZZFeatureMap_10_parametros(feature_dimension=2, reps=1,␣
 ↪theta_param=2, x=x,teta=teta)
#print(feature_map)
feature_map.decompose().draw("mpl")
```

[56]:



[19]:
```
backend
```

[19]:
```
<qiskit_ibm_runtime.fake_provider.backends.osaka.fake_osaka.FakeOsaka at
0x70dfff497a60>
```

[47]:
```
from qiskit_ibm_runtime import QiskitRuntimeService, SamplerV2 as Sampler

# Assuming you have already set up your Qiskit service and backend

# Initialize the sampler and run the circuits

sampler = Sampler(backend)
job = sampler.run(circuits_aa)
result = job.result()

# Iterate over the results to print and save the counts
with open('./FEATURE_RESULTS/
 ↪variacional_train_136_DP_NODP_IBM_Osaka_results_counts.txt', 'w') as f:
    for idx, pub_result in enumerate(result):
        counts = pub_result.data.meas.get_counts()
        f.write(f"Counts for pub {idx}: {counts}\n")
        print(f" > Counts for pub {idx}: {counts}")
```

```
> Counts for pub 0: {'11': 416, '01': 160, '00': 347, '10': 101}
> Counts for pub 1: {'10': 387, '01': 279, '00': 278, '11': 80}
> Counts for pub 2: {'11': 560, '00': 389, '01': 60, '10': 15}
> Counts for pub 3: {'11': 536, '01': 83, '00': 391, '10': 14}
> Counts for pub 4: {'00': 409, '01': 67, '11': 537, '10': 11}
> Counts for pub 5: {'10': 569, '00': 128, '01': 293, '11': 34}
> Counts for pub 6: {'00': 239, '01': 130, '11': 194, '10': 461}
> Counts for pub 7: {'01': 110, '00': 364, '11': 497, '10': 53}
> Counts for pub 8: {'11': 520, '00': 416, '01': 66, '10': 22}
> Counts for pub 9: {'01': 517, '10': 320, '00': 168, '11': 19}
> Counts for pub 10: {'00': 297, '10': 566, '01': 99, '11': 62}
> Counts for pub 11: {'00': 339, '11': 482, '01': 138, '10': 65}
```

```
> Counts for pub 12: {'00': 387, '11': 559, '01': 55, '10': 23}
> Counts for pub 13: {'10': 657, '00': 157, '01': 166, '11': 44}
> Counts for pub 14: {'01': 321, '10': 566, '00': 109, '11': 28}
> Counts for pub 15: {'01': 720, '00': 159, '10': 49, '11': 96}
> Counts for pub 16: {'00': 321, '11': 535, '10': 101, '01': 67}
> Counts for pub 17: {'10': 610, '00': 183, '11': 176, '01': 55}
> Counts for pub 18: {'10': 518, '01': 174, '00': 229, '11': 103}
> Counts for pub 19: {'10': 446, '01': 192, '11': 111, '00': 275}
> Counts for pub 20: {'00': 193, '11': 59, '01': 186, '10': 586}
> Counts for pub 21: {'10': 20, '01': 640, '11': 103, '00': 261}
> Counts for pub 22: {'01': 73, '11': 331, '10': 407, '00': 213}
> Counts for pub 23: {'11': 548, '00': 390, '01': 66, '10': 20}
> Counts for pub 24: {'01': 159, '11': 447, '00': 359, '10': 59}
> Counts for pub 25: {'11': 402, '10': 120, '00': 347, '01': 155}
> Counts for pub 26: {'10': 682, '00': 261, '01': 51, '11': 30}
> Counts for pub 27: {'01': 510, '10': 358, '00': 125, '11': 31}
> Counts for pub 28: {'01': 306, '10': 411, '00': 244, '11': 63}
> Counts for pub 29: {'01': 167, '10': 230, '11': 348, '00': 279}
> Counts for pub 30: {'00': 340, '01': 424, '11': 58, '10': 202}
> Counts for pub 31: {'11': 103, '10': 76, '01': 518, '00': 327}
> Counts for pub 32: {'10': 381, '01': 530, '11': 8, '00': 105}
> Counts for pub 33: {'11': 529, '00': 394, '10': 26, '01': 75}
> Counts for pub 34: {'11': 499, '00': 333, '01': 133, '10': 59}
> Counts for pub 35: {'01': 407, '00': 343, '11': 200, '10': 74}
> Counts for pub 36: {'00': 282, '01': 98, '10': 625, '11': 19}
> Counts for pub 37: {'00': 357, '11': 402, '10': 119, '01': 146}
> Counts for pub 38: {'11': 232, '01': 358, '00': 373, '10': 61}
> Counts for pub 39: {'00': 370, '01': 161, '11': 430, '10': 63}
> Counts for pub 40: {'11': 425, '10': 110, '00': 337, '01': 152}
> Counts for pub 41: {'11': 541, '01': 60, '00': 408, '10': 15}
> Counts for pub 42: {'00': 313, '01': 157, '11': 443, '10': 111}
> Counts for pub 43: {'01': 586, '10': 261, '00': 138, '11': 39}
> Counts for pub 44: {'10': 140, '01': 187, '11': 387, '00': 310}
> Counts for pub 45: {'01': 772, '00': 179, '11': 57, '10': 16}
> Counts for pub 46: {'11': 511, '01': 93, '00': 308, '10': 112}
> Counts for pub 47: {'00': 394, '10': 15, '11': 548, '01': 67}
> Counts for pub 48: {'00': 280, '10': 448, '01': 156, '11': 140}
> Counts for pub 49: {'01': 16, '10': 651, '00': 212, '11': 145}
> Counts for pub 50: {'00': 418, '11': 506, '01': 80, '10': 20}
> Counts for pub 51: {'00': 332, '11': 450, '10': 104, '01': 138}
> Counts for pub 52: {'00': 405, '11': 528, '01': 66, '10': 25}
> Counts for pub 53: {'00': 308, '11': 403, '01': 188, '10': 125}
> Counts for pub 54: {'01': 80, '11': 550, '00': 376, '10': 18}
> Counts for pub 55: {'11': 498, '00': 348, '01': 130, '10': 48}
> Counts for pub 56: {'10': 309, '11': 144, '00': 282, '01': 289}
> Counts for pub 57: {'11': 528, '00': 400, '01': 75, '10': 21}
> Counts for pub 58: {'11': 298, '00': 447, '01': 227, '10': 52}
> Counts for pub 59: {'11': 566, '00': 376, '01': 62, '10': 20}
```

```
> Counts for pub 60: {'10': 78, '00': 357, '01': 152, '11': 437}
> Counts for pub 61: {'11': 45, '10': 542, '01': 109, '00': 328}
> Counts for pub 62: {'11': 95, '10': 446, '00': 255, '01': 228}
> Counts for pub 63: {'11': 437, '00': 340, '10': 91, '01': 156}
> Counts for pub 64: {'00': 115, '01': 661, '10': 239, '11': 9}
> Counts for pub 65: {'11': 58, '10': 643, '01': 215, '00': 108}
> Counts for pub 66: {'11': 530, '00': 404, '01': 71, '10': 19}
> Counts for pub 67: {'11': 516, '00': 428, '01': 62, '10': 18}
> Counts for pub 68: {'01': 328, '00': 309, '11': 249, '10': 138}
> Counts for pub 69: {'11': 510, '10': 24, '00': 402, '01': 88}
> Counts for pub 70: {'01': 290, '10': 526, '00': 144, '11': 64}
> Counts for pub 71: {'11': 472, '00': 385, '10': 50, '01': 117}
> Counts for pub 72: {'11': 553, '00': 398, '01': 60, '10': 13}
> Counts for pub 73: {'10': 520, '00': 287, '01': 149, '11': 68}
> Counts for pub 74: {'11': 419, '10': 137, '00': 319, '01': 149}
> Counts for pub 75: {'00': 238, '01': 738, '11': 27, '10': 21}
> Counts for pub 76: {'01': 425, '00': 365, '10': 163, '11': 71}
> Counts for pub 77: {'11': 154, '01': 479, '00': 333, '10': 58}
> Counts for pub 78: {'01': 831, '00': 119, '10': 22, '11': 52}
> Counts for pub 79: {'00': 374, '11': 471, '01': 124, '10': 55}
> Counts for pub 80: {'10': 527, '01': 189, '00': 300, '11': 8}
> Counts for pub 81: {'11': 81, '10': 505, '01': 165, '00': 273}
> Counts for pub 82: {'00': 297, '11': 59, '10': 537, '01': 131}
> Counts for pub 83: {'10': 639, '00': 115, '01': 210, '11': 60}
> Counts for pub 84: {'10': 490, '01': 213, '00': 238, '11': 83}
> Counts for pub 85: {'11': 544, '00': 399, '01': 67, '10': 14}
> Counts for pub 86: {'00': 321, '10': 149, '11': 147, '01': 407}
> Counts for pub 87: {'01': 519, '00': 313, '10': 117, '11': 75}
> Counts for pub 88: {'00': 412, '11': 526, '10': 23, '01': 63}
> Counts for pub 89: {'00': 297, '10': 123, '01': 474, '11': 130}
> Counts for pub 90: {'10': 355, '11': 132, '01': 275, '00': 262}
> Counts for pub 91: {'00': 176, '10': 672, '11': 153, '01': 23}
> Counts for pub 92: {'00': 359, '01': 74, '10': 30, '11': 561}
> Counts for pub 93: {'00': 380, '11': 349, '01': 237, '10': 58}
> Counts for pub 94: {'00': 186, '10': 538, '11': 232, '01': 68}
> Counts for pub 95: {'01': 526, '11': 66, '00': 195, '10': 237}
> Counts for pub 96: {'00': 200, '10': 777, '01': 14, '11': 33}
> Counts for pub 97: {'00': 384, '11': 549, '01': 73, '10': 18}
> Counts for pub 98: {'00': 474, '11': 395, '01': 117, '10': 38}
> Counts for pub 99: {'11': 558, '01': 58, '00': 390, '10': 18}
> Counts for pub 100: {'00': 400, '11': 527, '01': 77, '10': 20}
> Counts for pub 101: {'11': 529, '00': 415, '01': 61, '10': 19}
> Counts for pub 102: {'10': 582, '01': 220, '11': 56, '00': 166}
> Counts for pub 103: {'01': 578, '00': 334, '11': 102, '10': 10}
> Counts for pub 104: {'11': 389, '01': 183, '10': 140, '00': 312}
> Counts for pub 105: {'10': 377, '00': 299, '01': 281, '11': 67}
> Counts for pub 106: {'00': 374, '11': 550, '01': 75, '10': 25}
> Counts for pub 107: {'00': 354, '01': 155, '11': 448, '10': 67}
```

```
> Counts for pub 108: {'00': 408, '11': 515, '01': 86, '10': 15}
> Counts for pub 109: {'11': 424, '00': 381, '10': 55, '01': 164}
> Counts for pub 110: {'00': 301, '10': 608, '01': 109, '11': 6}
> Counts for pub 111: {'00': 76, '01': 786, '11': 67, '10': 95}
> Counts for pub 112: {'11': 538, '00': 403, '01': 67, '10': 16}
> Counts for pub 113: {'10': 621, '00': 150, '01': 162, '11': 91}
> Counts for pub 114: {'10': 721, '00': 204, '11': 45, '01': 54}
> Counts for pub 115: {'01': 65, '00': 411, '11': 525, '10': 23}
> Counts for pub 116: {'01': 618, '00': 139, '10': 230, '11': 37}
> Counts for pub 117: {'11': 547, '00': 395, '01': 64, '10': 18}
> Counts for pub 118: {'00': 396, '01': 72, '11': 545, '10': 11}
> Counts for pub 119: {'11': 427, '01': 336, '00': 217, '10': 44}
> Counts for pub 120: {'10': 132, '01': 734, '00': 135, '11': 23}
> Counts for pub 121: {'01': 185, '10': 655, '00': 154, '11': 30}
> Counts for pub 122: {'11': 441, '00': 321, '10': 107, '01': 155}
> Counts for pub 123: {'10': 661, '11': 119, '00': 193, '01': 51}
> Counts for pub 124: {'11': 106, '00': 355, '10': 406, '01': 157}
> Counts for pub 125: {'01': 601, '00': 181, '11': 189, '10': 53}
> Counts for pub 126: {'01': 402, '00': 303, '11': 150, '10': 169}
> Counts for pub 127: {'01': 562, '00': 129, '10': 309, '11': 24}
> Counts for pub 128: {'10': 180, '01': 511, '00': 233, '11': 100}
> Counts for pub 129: {'01': 454, '10': 208, '11': 50, '00': 312}
> Counts for pub 130: {'11': 531, '00': 415, '01': 57, '10': 21}
> Counts for pub 131: {'11': 206, '10': 262, '01': 264, '00': 292}
> Counts for pub 132: {'00': 358, '01': 434, '10': 160, '11': 72}
> Counts for pub 133: {'01': 156, '11': 423, '10': 130, '00': 315}
> Counts for pub 134: {'00': 362, '10': 60, '11': 450, '01': 152}
> Counts for pub 135: {'00': 406, '10': 16, '11': 537, '01': 65}
```

[48]:
```python
# Extract counts for '00' from each post and calculate the total sum of counts␣
 ↪per post
# parity calculation is given for each Pub in "paridad"
counts_00 = []
paridad = []


counts_data = []
for idx, pub_result in enumerate(result):
        counts = pub_result.data.meas.get_counts()
        counts_data.append(counts)

        print(f" > Counts for pub {idx}: {counts}")

for counts in counts_data:
    total_counts = sum(counts.values())
    #print(total_counts)
    count_00 = counts.get('00', 0)
```

```
    count_01 = counts.get('01', 0)
    count_10 = counts.get('10', 0)
    count_11 = counts.get('11', 0)
    count_par = count_00 + count_11
    count_impar = count_01 + count_10


    if count_par > count_impar:
        pari = 1  # Paridad impar
    elif count_par < count_impar:
        pari = 0  # Paridad par
    else:
        if count_par == 0 or count_impar == 0:
            pari = None  # Indicar paridad inconclusa
        else:
            pari = 0  # Asignar paridad par arbitrariamente

    #count_00_pu = count_00 / total_counts if total_counts != 0 else 0  #␣
↪Evitar división por cero
    paridad.append(pari)
```

```
> Counts for pub 0: {'11': 416, '01': 160, '00': 347, '10': 101}
> Counts for pub 1: {'10': 387, '01': 279, '00': 278, '11': 80}
> Counts for pub 2: {'11': 560, '00': 389, '01': 60, '10': 15}
> Counts for pub 3: {'11': 536, '01': 83, '00': 391, '10': 14}
> Counts for pub 4: {'00': 409, '01': 67, '11': 537, '10': 11}
> Counts for pub 5: {'10': 569, '00': 128, '01': 293, '11': 34}
> Counts for pub 6: {'00': 239, '01': 130, '11': 194, '10': 461}
> Counts for pub 7: {'01': 110, '00': 364, '11': 497, '10': 53}
> Counts for pub 8: {'11': 520, '00': 416, '01': 66, '10': 22}
> Counts for pub 9: {'01': 517, '10': 320, '00': 168, '11': 19}
> Counts for pub 10: {'00': 297, '10': 566, '01': 99, '11': 62}
> Counts for pub 11: {'00': 339, '11': 482, '01': 138, '10': 65}
> Counts for pub 12: {'00': 387, '11': 559, '01': 55, '10': 23}
> Counts for pub 13: {'10': 657, '00': 157, '01': 166, '11': 44}
> Counts for pub 14: {'01': 321, '10': 566, '00': 109, '11': 28}
> Counts for pub 15: {'01': 720, '00': 159, '10': 49, '11': 96}
> Counts for pub 16: {'00': 321, '11': 535, '10': 101, '01': 67}
> Counts for pub 17: {'10': 610, '00': 183, '11': 176, '01': 55}
> Counts for pub 18: {'10': 518, '01': 174, '00': 229, '11': 103}
> Counts for pub 19: {'10': 446, '01': 192, '11': 111, '00': 275}
> Counts for pub 20: {'00': 193, '11': 59, '01': 186, '10': 586}
> Counts for pub 21: {'10': 20, '01': 640, '11': 103, '00': 261}
> Counts for pub 22: {'01': 73, '11': 331, '10': 407, '00': 213}
> Counts for pub 23: {'11': 548, '00': 390, '01': 66, '10': 20}
> Counts for pub 24: {'01': 159, '11': 447, '00': 359, '10': 59}
> Counts for pub 25: {'11': 402, '10': 120, '00': 347, '01': 155}
```

```
> Counts for pub 26: {'10': 682, '00': 261, '01': 51, '11': 30}
> Counts for pub 27: {'01': 510, '10': 358, '00': 125, '11': 31}
> Counts for pub 28: {'01': 306, '10': 411, '00': 244, '11': 63}
> Counts for pub 29: {'01': 167, '10': 230, '11': 348, '00': 279}
> Counts for pub 30: {'00': 340, '01': 424, '11': 58, '10': 202}
> Counts for pub 31: {'11': 103, '10': 76, '01': 518, '00': 327}
> Counts for pub 32: {'10': 381, '01': 530, '11': 8, '00': 105}
> Counts for pub 33: {'11': 529, '00': 394, '10': 26, '01': 75}
> Counts for pub 34: {'11': 499, '00': 333, '01': 133, '10': 59}
> Counts for pub 35: {'01': 407, '00': 343, '11': 200, '10': 74}
> Counts for pub 36: {'00': 282, '01': 98, '10': 625, '11': 19}
> Counts for pub 37: {'00': 357, '11': 402, '10': 119, '01': 146}
> Counts for pub 38: {'11': 232, '01': 358, '00': 373, '10': 61}
> Counts for pub 39: {'00': 370, '01': 161, '11': 430, '10': 63}
> Counts for pub 40: {'11': 425, '10': 110, '00': 337, '01': 152}
> Counts for pub 41: {'11': 541, '01': 60, '00': 408, '10': 15}
> Counts for pub 42: {'00': 313, '01': 157, '11': 443, '10': 111}
> Counts for pub 43: {'01': 586, '10': 261, '00': 138, '11': 39}
> Counts for pub 44: {'10': 140, '01': 187, '11': 387, '00': 310}
> Counts for pub 45: {'01': 772, '00': 179, '11': 57, '10': 16}
> Counts for pub 46: {'11': 511, '01': 93, '00': 308, '10': 112}
> Counts for pub 47: {'00': 394, '10': 15, '11': 548, '01': 67}
> Counts for pub 48: {'00': 280, '10': 448, '01': 156, '11': 140}
> Counts for pub 49: {'01': 16, '10': 651, '00': 212, '11': 145}
> Counts for pub 50: {'00': 418, '11': 506, '01': 80, '10': 20}
> Counts for pub 51: {'00': 332, '11': 450, '10': 104, '01': 138}
> Counts for pub 52: {'00': 405, '11': 528, '01': 66, '10': 25}
> Counts for pub 53: {'00': 308, '11': 403, '01': 188, '10': 125}
> Counts for pub 54: {'01': 80, '11': 550, '00': 376, '10': 18}
> Counts for pub 55: {'11': 498, '00': 348, '01': 130, '10': 48}
> Counts for pub 56: {'10': 309, '11': 144, '00': 282, '01': 289}
> Counts for pub 57: {'11': 528, '00': 400, '01': 75, '10': 21}
> Counts for pub 58: {'11': 298, '00': 447, '01': 227, '10': 52}
> Counts for pub 59: {'11': 566, '00': 376, '01': 62, '10': 20}
> Counts for pub 60: {'10': 78, '00': 357, '01': 152, '11': 437}
> Counts for pub 61: {'11': 45, '10': 542, '01': 109, '00': 328}
> Counts for pub 62: {'11': 95, '10': 446, '00': 255, '01': 228}
> Counts for pub 63: {'11': 437, '00': 340, '10': 91, '01': 156}
> Counts for pub 64: {'00': 115, '01': 661, '10': 239, '11': 9}
> Counts for pub 65: {'11': 58, '10': 643, '01': 215, '00': 108}
> Counts for pub 66: {'11': 530, '00': 404, '01': 71, '10': 19}
> Counts for pub 67: {'11': 516, '00': 428, '01': 62, '10': 18}
> Counts for pub 68: {'01': 328, '00': 309, '11': 249, '10': 138}
> Counts for pub 69: {'11': 510, '10': 24, '00': 402, '01': 88}
> Counts for pub 70: {'01': 290, '10': 526, '00': 144, '11': 64}
> Counts for pub 71: {'11': 472, '00': 385, '10': 50, '01': 117}
> Counts for pub 72: {'11': 553, '00': 398, '01': 60, '10': 13}
> Counts for pub 73: {'10': 520, '00': 287, '01': 149, '11': 68}
```

```
> Counts for pub 74: {'11': 419, '10': 137, '00': 319, '01': 149}
> Counts for pub 75: {'00': 238, '01': 738, '11': 27, '10': 21}
> Counts for pub 76: {'01': 425, '00': 365, '10': 163, '11': 71}
> Counts for pub 77: {'11': 154, '01': 479, '00': 333, '10': 58}
> Counts for pub 78: {'01': 831, '00': 119, '10': 22, '11': 52}
> Counts for pub 79: {'00': 374, '11': 471, '01': 124, '10': 55}
> Counts for pub 80: {'10': 527, '01': 189, '00': 300, '11': 8}
> Counts for pub 81: {'11': 81, '10': 505, '01': 165, '00': 273}
> Counts for pub 82: {'00': 297, '11': 59, '10': 537, '01': 131}
> Counts for pub 83: {'10': 639, '00': 115, '01': 210, '11': 60}
> Counts for pub 84: {'10': 490, '01': 213, '00': 238, '11': 83}
> Counts for pub 85: {'11': 544, '00': 399, '01': 67, '10': 14}
> Counts for pub 86: {'00': 321, '10': 149, '11': 147, '01': 407}
> Counts for pub 87: {'01': 519, '00': 313, '10': 117, '11': 75}
> Counts for pub 88: {'00': 412, '11': 526, '10': 23, '01': 63}
> Counts for pub 89: {'00': 297, '10': 123, '01': 474, '11': 130}
> Counts for pub 90: {'10': 355, '11': 132, '01': 275, '00': 262}
> Counts for pub 91: {'00': 176, '10': 672, '11': 153, '01': 23}
> Counts for pub 92: {'00': 359, '01': 74, '10': 30, '11': 561}
> Counts for pub 93: {'00': 380, '11': 349, '01': 237, '10': 58}
> Counts for pub 94: {'00': 186, '10': 538, '11': 232, '01': 68}
> Counts for pub 95: {'01': 526, '11': 66, '00': 195, '10': 237}
> Counts for pub 96: {'00': 200, '10': 777, '01': 14, '11': 33}
> Counts for pub 97: {'00': 384, '11': 549, '01': 73, '10': 18}
> Counts for pub 98: {'00': 474, '11': 395, '01': 117, '10': 38}
> Counts for pub 99: {'11': 558, '01': 58, '00': 390, '10': 18}
> Counts for pub 100: {'00': 400, '11': 527, '01': 77, '10': 20}
> Counts for pub 101: {'11': 529, '00': 415, '01': 61, '10': 19}
> Counts for pub 102: {'10': 582, '01': 220, '11': 56, '00': 166}
> Counts for pub 103: {'01': 578, '00': 334, '11': 102, '10': 10}
> Counts for pub 104: {'11': 389, '01': 183, '10': 140, '00': 312}
> Counts for pub 105: {'10': 377, '00': 299, '01': 281, '11': 67}
> Counts for pub 106: {'00': 374, '11': 550, '01': 75, '10': 25}
> Counts for pub 107: {'00': 354, '01': 155, '11': 448, '10': 67}
> Counts for pub 108: {'00': 408, '11': 515, '01': 86, '10': 15}
> Counts for pub 109: {'11': 424, '00': 381, '10': 55, '01': 164}
> Counts for pub 110: {'00': 301, '10': 608, '01': 109, '11': 6}
> Counts for pub 111: {'00': 76, '01': 786, '11': 67, '10': 95}
> Counts for pub 112: {'11': 538, '00': 403, '01': 67, '10': 16}
> Counts for pub 113: {'10': 621, '00': 150, '01': 162, '11': 91}
> Counts for pub 114: {'10': 721, '00': 204, '11': 45, '01': 54}
> Counts for pub 115: {'01': 65, '00': 411, '11': 525, '10': 23}
> Counts for pub 116: {'01': 618, '00': 139, '10': 230, '11': 37}
> Counts for pub 117: {'11': 547, '00': 395, '01': 64, '10': 18}
> Counts for pub 118: {'00': 396, '01': 72, '11': 545, '10': 11}
> Counts for pub 119: {'11': 427, '01': 336, '00': 217, '10': 44}
> Counts for pub 120: {'10': 132, '01': 734, '00': 135, '11': 23}
> Counts for pub 121: {'01': 185, '10': 655, '00': 154, '11': 30}
```

```
> Counts for pub 122: {'11': 441, '00': 321, '10': 107, '01': 155}
> Counts for pub 123: {'10': 661, '11': 119, '00': 193, '01': 51}
> Counts for pub 124: {'11': 106, '00': 355, '10': 406, '01': 157}
> Counts for pub 125: {'01': 601, '00': 181, '11': 189, '10': 53}
> Counts for pub 126: {'01': 402, '00': 303, '11': 150, '10': 169}
> Counts for pub 127: {'01': 562, '00': 129, '10': 309, '11': 24}
> Counts for pub 128: {'10': 180, '01': 511, '00': 233, '11': 100}
> Counts for pub 129: {'01': 454, '10': 208, '11': 50, '00': 312}
> Counts for pub 130: {'11': 531, '00': 415, '01': 57, '10': 21}
> Counts for pub 131: {'11': 206, '10': 262, '01': 264, '00': 292}
> Counts for pub 132: {'00': 358, '01': 434, '10': 160, '11': 72}
> Counts for pub 133: {'01': 156, '11': 423, '10': 130, '00': 315}
> Counts for pub 134: {'00': 362, '10': 60, '11': 450, '01': 152}
> Counts for pub 135: {'00': 406, '10': 16, '11': 537, '01': 65}
```

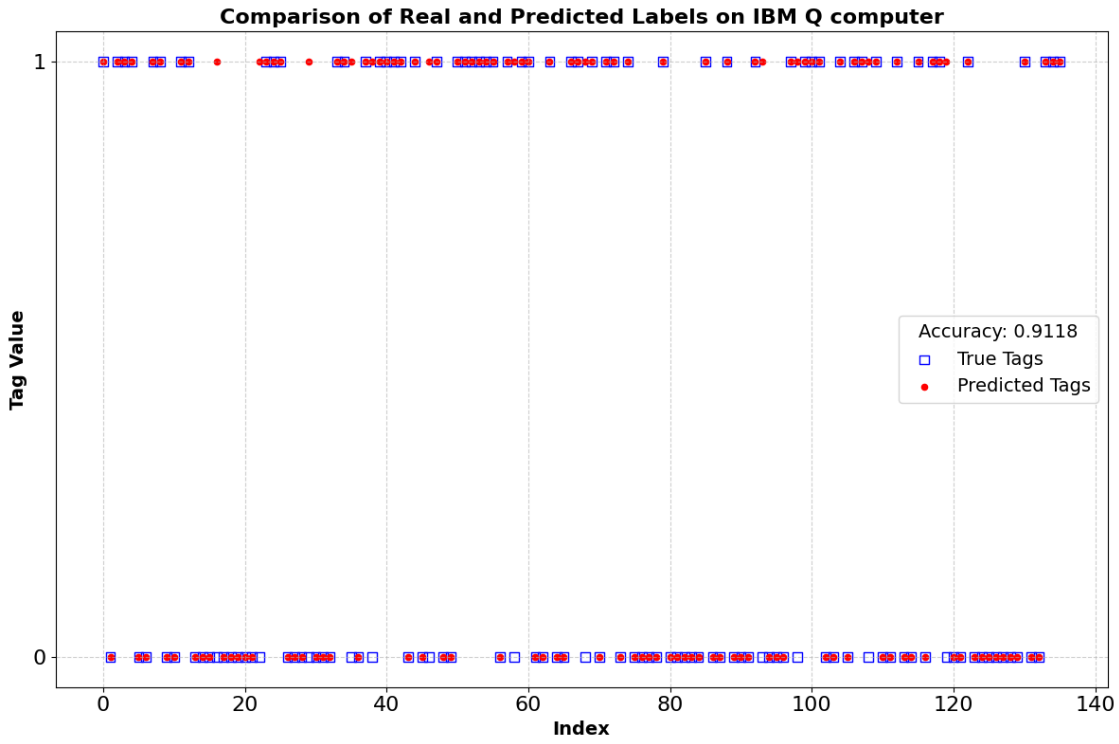
[49]:
```python
def parity(bitstring):
    """Returns 1 if parity of `bitstring` is even, otherwise 0."""
    hamming_weight = sum(int(k) for k in list(bitstring))
    return (hamming_weight+1) % 2

def label_probability(results):
    """Converts a dict of bitstrings and their counts,
    to parities and their counts"""
    shots = sum(results.values())
    probabilities = {0: 0, 1: 0}
    for bitstring, counts in results.items():
        label = parity(bitstring)
        probabilities[label] += counts / shots
    return probabilities

#Calculates the accuracy of predicting parities based on train labels.
def calculate_accuracy(train_labels, parity):
    """
    Calculates the accuracy of predicting parities based on train labels.

    Args:
        train_labels (list): A list of true parity labels (0 or 1).
        parity (list): A list of predicted parity values (0 or 1).

    Returns:
        float: The accuracy of parity predictions (0.0 to 1.0).
    """

    if len(train_labels) != len(parity):
        raise ValueError("train_labels and parity must have the same length")
```

```
    correct_predictions = sum(1 for true_label, predicted_label in␣
 ↪zip(train_labels, parity) if true_label == predicted_label)

    accuracy = correct_predictions / len(train_labels)
    return accuracy

# Example usage (assuming TRAIN_LABELS and parity are populated correctly as␣
 ↪lists)
#TRAIN_LABELS = [0, 1, 0, 1, 1, 0]  # Example true labels
#parity = [0, 1, 0, 0, 1, 0]  # Example predicted parity values

accuracy = calculate_accuracy(TRAIN_LABELS[0:136], paridad)
print(f"Accuracy: {accuracy:.4f}")
```

```
Accuracy: 0.9118
```

[66]:
```
import numpy as np
import matplotlib.pyplot as plt

# Sample data (replace with your actual data)
# Assuming TRAIN_LABELS and paridad are already defined
# For demonstration, let's create some dummy data:

# ... (Rest of your code to calculate 'paridad' and 'accuracy') ...

# Calculate accuracy (you should already have this function)
def calculate_accuracy(train_labels, parity):
    # ... (Your implementation of the function) ...
    if len(train_labels) != len(parity):
        raise ValueError("train_labels and parity must have the same length")

    correct_predictions = sum(1 for true_label, predicted_label in␣
 ↪zip(train_labels, parity) if true_label == predicted_label)

    accuracy = correct_predictions / len(train_labels)
    return accuracy

accuracy = calculate_accuracy(TRAIN_LABELS[0:136], paridad)

# Verificar que ambas listas tengan la misma longitud
if len(TRAIN_LABELS[0:136]) != len(paridad):
    raise ValueError("TRAIN_LABELS and parity must have the same length")

# Crear una figura y ejes
fig, ax = plt.subplots(figsize=(12, 8))

# Configurar el gráfico
```

```
indices = np.arange(len(TRAIN_LABELS[0:136]))

# Graficar las etiquetas reales como puntos cuadrados sin relleno
ax.scatter(indices, TRAIN_LABELS[0:136], label='True Tags', edgecolors='b',␣
 ↪facecolors='none', marker='s', s=50)

# Graficar las etiquetas predichas como puntos redondos más pequeños
ax.scatter(indices, paridad, label='Predicted Tags', color='r', marker='o',␣
 ↪s=20)

# Configurar etiquetas y título
ax.set_xlabel('Index', fontsize=14, fontweight='bold')
ax.set_ylabel('Tag Value', fontsize=14, fontweight='bold')
ax.set_title('Comparison of Real and Predicted Labels on IBM Q computer',␣
 ↪fontsize=16, fontweight='bold')

# Configurar la leyenda
# Modificar la leyenda para incluir la precisión
legend = ax.legend(fontsize=14, title=f"Accuracy: {accuracy:.4f}")  # Add␣
 ↪accuracy to legend title
plt.setp(legend.get_title(), fontsize=14)  # Set legend title font size

# Ajustar los ticks del eje y para que sean solo 0 y 1
ax.set_yticks([0, 1])

# Mejorar la visualización de la cuadrícula
ax.grid(True, linestyle='--', alpha=0.6)

# Aumentar el tamaño de los números en los ejes
ax.tick_params(axis='x', labelsize=16)
ax.tick_params(axis='y', labelsize=16)

# Ajustar el espaciado de los elementos
plt.tight_layout()

# Mostrar el gráfico
plt.show()
```



[ ]:
```
total_counts
```

[ ]: 4096

[ ]:
```
counts_data = []
for idx, pub_result in enumerate(result):
        counts = pub_result.data.meas.get_counts()
        counts_data.append(counts)

        print(f" > Counts for pub {idx}: {counts}")
```

```
> Counts for pub 0: {'00': 336, '10': 256, '01': 76, '11': 356}
> Counts for pub 1: {'00': 252, '01': 155, '11': 364, '10': 253}
> Counts for pub 2: {'00': 369, '11': 316, '10': 234, '01': 105}
> Counts for pub 3: {'10': 294, '00': 342, '11': 325, '01': 63}
> Counts for pub 4: {'10': 201, '11': 383, '00': 349, '01': 91}
> Counts for pub 5: {'00': 263, '10': 199, '01': 320, '11': 242}
> Counts for pub 6: {'11': 132, '10': 719, '00': 131, '01': 42}
> Counts for pub 7: {'01': 341, '10': 487, '00': 131, '11': 65}
> Counts for pub 8: {'00': 339, '10': 284, '01': 83, '11': 318}
> Counts for pub 9: {'10': 282, '11': 117, '01': 564, '00': 61}
> Counts for pub 10: {'00': 349, '11': 395, '10': 186, '01': 94}
> Counts for pub 11: {'10': 495, '01': 319, '11': 125, '00': 85}
> Counts for pub 12: {'10': 472, '00': 182, '01': 227, '11': 143}
```

```
> Counts for pub 13: {'00': 508, '01': 215, '10': 270, '11': 31}
> Counts for pub 14: {'11': 350, '10': 240, '01': 97, '00': 337}
> Counts for pub 15: {'11': 687, '00': 243, '01': 68, '10': 26}
> Counts for pub 16: {'11': 404, '00': 363, '10': 161, '01': 96}
> Counts for pub 17: {'01': 498, '10': 349, '00': 140, '11': 37}
> Counts for pub 18: {'11': 324, '00': 312, '10': 306, '01': 82}
> Counts for pub 19: {'01': 77, '00': 418, '11': 280, '10': 249}
> Counts for pub 20: {'00': 366, '11': 353, '10': 215, '01': 90}
> Counts for pub 21: {'10': 357, '11': 340, '00': 247, '01': 80}
> Counts for pub 22: {'10': 252, '11': 204, '01': 323, '00': 245}
> Counts for pub 23: {'10': 523, '01': 55, '11': 256, '00': 190}
> Counts for pub 24: {'10': 266, '01': 91, '11': 345, '00': 322}
> Counts for pub 25: {'01': 456, '11': 243, '10': 226, '00': 99}
> Counts for pub 26: {'10': 490, '11': 170, '01': 138, '00': 226}
> Counts for pub 27: {'00': 329, '11': 519, '01': 87, '10': 89}
> Counts for pub 28: {'10': 489, '01': 473, '00': 20, '11': 42}
> Counts for pub 29: {'10': 465, '00': 77, '01': 417, '11': 65}
> Counts for pub 30: {'10': 553, '01': 409, '00': 22, '11': 40}
> Counts for pub 31: {'01': 584, '10': 337, '00': 45, '11': 58}
> Counts for pub 32: {'10': 515, '00': 128, '01': 357, '11': 24}
> Counts for pub 33: {'01': 286, '11': 41, '10': 561, '00': 136}
> Counts for pub 34: {'11': 373, '01': 124, '10': 228, '00': 299}
> Counts for pub 35: {'10': 719, '00': 121, '11': 151, '01': 33}
> Counts for pub 36: {'10': 182, '01': 157, '11': 398, '00': 287}
> Counts for pub 37: {'10': 556, '11': 36, '01': 355, '00': 77}
> Counts for pub 38: {'11': 376, '00': 339, '10': 227, '01': 82}
> Counts for pub 39: {'00': 356, '11': 337, '10': 264, '01': 67}
> Counts for pub 40: {'01': 70, '11': 249, '10': 313, '00': 392}
> Counts for pub 41: {'11': 376, '01': 89, '10': 197, '00': 362}
> Counts for pub 42: {'11': 30, '10': 586, '00': 72, '01': 336}
> Counts for pub 43: {'10': 278, '11': 93, '01': 588, '00': 65}
> Counts for pub 44: {'10': 194, '00': 631, '11': 169, '01': 30}
> Counts for pub 45: {'10': 248, '11': 356, '00': 336, '01': 84}
> Counts for pub 46: {'10': 482, '01': 438, '00': 49, '11': 55}
> Counts for pub 47: {'10': 542, '01': 387, '11': 47, '00': 48}
> Counts for pub 48: {'01': 102, '11': 373, '00': 345, '10': 204}
> Counts for pub 49: {'10': 509, '01': 411, '11': 62, '00': 42}
> Counts for pub 50: {'01': 234, '10': 656, '11': 78, '00': 56}
> Counts for pub 51: {'10': 683, '00': 123, '11': 174, '01': 44}
> Counts for pub 52: {'01': 361, '10': 475, '11': 77, '00': 111}
> Counts for pub 53: {'01': 543, '10': 252, '11': 178, '00': 51}
> Counts for pub 54: {'10': 257, '11': 296, '00': 388, '01': 83}
> Counts for pub 55: {'11': 270, '01': 359, '00': 194, '10': 201}
> Counts for pub 56: {'11': 398, '10': 258, '01': 157, '00': 211}
> Counts for pub 57: {'01': 377, '10': 524, '00': 103, '11': 20}
> Counts for pub 58: {'01': 381, '11': 28, '10': 520, '00': 95}
> Counts for pub 59: {'10': 484, '00': 112, '01': 378, '11': 50}
> Counts for pub 60: {'11': 404, '10': 188, '00': 351, '01': 81}
```

```
> Counts for pub 61: {'00': 323, '11': 347, '10': 276, '01': 78}
> Counts for pub 62: {'01': 303, '10': 525, '00': 161, '11': 35}
> Counts for pub 63: {'00': 334, '11': 383, '10': 219, '01': 88}
> Counts for pub 64: {'00': 161, '10': 679, '11': 139, '01': 45}
> Counts for pub 65: {'10': 657, '11': 201, '00': 121, '01': 45}
> Counts for pub 66: {'11': 686, '01': 97, '10': 192, '00': 49}
> Counts for pub 67: {'00': 413, '11': 301, '10': 226, '01': 84}
> Counts for pub 68: {'00': 381, '10': 273, '11': 299, '01': 71}
> Counts for pub 69: {'11': 714, '00': 53, '10': 75, '01': 182}
> Counts for pub 70: {'10': 454, '01': 512, '00': 41, '11': 17}
> Counts for pub 71: {'10': 734, '00': 161, '11': 112, '01': 17}
> Counts for pub 72: {'10': 390, '00': 32, '01': 580, '11': 22}
> Counts for pub 73: {'01': 357, '10': 519, '11': 41, '00': 107}
> Counts for pub 74: {'11': 339, '10': 298, '00': 328, '01': 59}
> Counts for pub 75: {'01': 457, '10': 480, '11': 69, '00': 18}
> Counts for pub 76: {'00': 606, '11': 398, '01': 13, '10': 7}
> Counts for pub 77: {'01': 623, '10': 291, '11': 91, '00': 19}
> Counts for pub 78: {'10': 490, '01': 326, '11': 15, '00': 193}
> Counts for pub 79: {'01': 433, '10': 467, '11': 16, '00': 108}
> Counts for pub 80: {'00': 71, '01': 469, '10': 472, '11': 12}
> Counts for pub 81: {'00': 362, '10': 208, '11': 372, '01': 82}
> Counts for pub 82: {'00': 207, '10': 414, '01': 377, '11': 26}
> Counts for pub 83: {'10': 392, '01': 583, '11': 34, '00': 15}
> Counts for pub 84: {'10': 178, '11': 397, '01': 80, '00': 369}
> Counts for pub 85: {'01': 481, '10': 478, '11': 57, '00': 8}
> Counts for pub 86: {'11': 267, '00': 422, '10': 269, '01': 66}
> Counts for pub 87: {'00': 137, '11': 126, '10': 746, '01': 15}
> Counts for pub 88: {'10': 531, '11': 82, '01': 339, '00': 72}
> Counts for pub 89: {'01': 479, '10': 296, '11': 161, '00': 88}
> Counts for pub 90: {'11': 377, '00': 383, '10': 170, '01': 94}
> Counts for pub 91: {'11': 261, '10': 275, '01': 102, '00': 386}
> Counts for pub 92: {'00': 331, '11': 389, '10': 208, '01': 96}
> Counts for pub 93: {'00': 377, '11': 348, '10': 193, '01': 106}
> Counts for pub 94: {'11': 335, '10': 215, '00': 382, '01': 92}
> Counts for pub 95: {'10': 508, '11': 51, '01': 396, '00': 69}
> Counts for pub 96: {'00': 713, '11': 261, '10': 29, '01': 21}
> Counts for pub 97: {'10': 216, '00': 358, '11': 369, '01': 81}
> Counts for pub 98: {'10': 348, '00': 243, '01': 285, '11': 148}
> Counts for pub 99: {'10': 446, '01': 475, '00': 91, '11': 12}
> Counts for pub 100: {'10': 512, '01': 369, '00': 114, '11': 29}
> Counts for pub 101: {'00': 334, '10': 268, '11': 352, '01': 70}
> Counts for pub 102: {'10': 246, '01': 94, '00': 408, '11': 276}
> Counts for pub 103: {'01': 364, '10': 513, '00': 100, '11': 47}
> Counts for pub 104: {'10': 570, '01': 297, '11': 29, '00': 128}
> Counts for pub 105: {'00': 93, '10': 491, '01': 390, '11': 50}
> Counts for pub 106: {'10': 498, '01': 463, '00': 40, '11': 23}
> Counts for pub 107: {'10': 252, '01': 85, '00': 374, '11': 313}
> Counts for pub 108: {'00': 142, '10': 457, '01': 353, '11': 72}
```

```
> Counts for pub 109: {'10': 225, '01': 619, '11': 161, '00': 19}
> Counts for pub 110: {'00': 362, '11': 411, '01': 92, '10': 159}
> Counts for pub 111: {'01': 583, '10': 333, '11': 67, '00': 41}
> Counts for pub 112: {'10': 458, '01': 434, '00': 117, '11': 15}
> Counts for pub 113: {'01': 74, '10': 302, '00': 344, '11': 304}
> Counts for pub 114: {'10': 792, '00': 140, '11': 83, '01': 9}
> Counts for pub 115: {'11': 329, '10': 291, '00': 330, '01': 74}
> Counts for pub 116: {'01': 74, '00': 384, '10': 274, '11': 292}
> Counts for pub 117: {'11': 362, '10': 192, '00': 376, '01': 94}
> Counts for pub 118: {'00': 347, '11': 323, '10': 265, '01': 89}
> Counts for pub 119: {'10': 201, '00': 416, '11': 303, '01': 104}
> Counts for pub 120: {'11': 325, '00': 325, '10': 288, '01': 86}
> Counts for pub 121: {'10': 381, '01': 568, '11': 39, '00': 36}
> Counts for pub 122: {'01': 363, '10': 509, '00': 114, '11': 38}
> Counts for pub 123: {'01': 538, '10': 455, '00': 18, '11': 13}
> Counts for pub 124: {'10': 305, '11': 76, '01': 618, '00': 25}
> Counts for pub 125: {'11': 332, '10': 274, '00': 341, '01': 77}
> Counts for pub 126: {'01': 88, '00': 327, '11': 365, '10': 244}
> Counts for pub 127: {'10': 522, '01': 362, '00': 102, '11': 38}
> Counts for pub 128: {'11': 268, '01': 91, '00': 410, '10': 255}
> Counts for pub 129: {'01': 264, '10': 628, '11': 44, '00': 88}
> Counts for pub 130: {'10': 316, '00': 373, '11': 263, '01': 72}
> Counts for pub 131: {'11': 546, '01': 352, '10': 99, '00': 27}
> Counts for pub 132: {'11': 456, '01': 104, '10': 163, '00': 301}
> Counts for pub 133: {'10': 481, '00': 135, '01': 360, '11': 48}
> Counts for pub 134: {'00': 208, '10': 344, '11': 373, '01': 99}
> Counts for pub 135: {'01': 93, '10': 281, '00': 367, '11': 283}
> Counts for pub 136: {'10': 411, '01': 229, '00': 215, '11': 169}
> Counts for pub 137: {'11': 466, '00': 341, '01': 99, '10': 118}
> Counts for pub 138: {'01': 367, '00': 275, '11': 149, '10': 233}
> Counts for pub 139: {'11': 198, '00': 200, '10': 580, '01': 46}
> Counts for pub 140: {'01': 310, '10': 579, '11': 82, '00': 53}
> Counts for pub 141: {'10': 590, '01': 331, '11': 64, '00': 39}
> Counts for pub 142: {'11': 471, '10': 160, '00': 316, '01': 77}
> Counts for pub 143: {'11': 345, '00': 331, '01': 101, '10': 247}
> Counts for pub 144: {'10': 143, '00': 327, '11': 448, '01': 106}
> Counts for pub 145: {'10': 195, '01': 297, '00': 188, '11': 344}
> Counts for pub 146: {'10': 261, '11': 322, '00': 370, '01': 71}
> Counts for pub 147: {'01': 540, '00': 84, '10': 266, '11': 134}
> Counts for pub 148: {'10': 659, '01': 211, '00': 70, '11': 84}
> Counts for pub 149: {'00': 342, '01': 74, '11': 335, '10': 273}
> Counts for pub 150: {'01': 178, '11': 741, '10': 55, '00': 50}
> Counts for pub 151: {'10': 483, '00': 125, '11': 69, '01': 347}
> Counts for pub 152: {'01': 85, '00': 385, '11': 282, '10': 272}
> Counts for pub 153: {'00': 366, '01': 210, '11': 37, '10': 411}
> Counts for pub 154: {'01': 80, '11': 288, '00': 385, '10': 271}
> Counts for pub 155: {'11': 327, '10': 207, '00': 386, '01': 104}
> Counts for pub 156: {'10': 260, '00': 346, '11': 324, '01': 94}
```

```
> Counts for pub 157: {'11': 264, '10': 536, '00': 147, '01': 77}
> Counts for pub 158: {'00': 246, '10': 387, '11': 324, '01': 67}
> Counts for pub 159: {'10': 495, '01': 350, '00': 109, '11': 70}
> Counts for pub 160: {'11': 333, '01': 81, '00': 354, '10': 256}
> Counts for pub 161: {'10': 781, '11': 87, '00': 137, '01': 19}
> Counts for pub 162: {'00': 398, '11': 245, '10': 308, '01': 73}
> Counts for pub 163: {'00': 124, '10': 502, '11': 89, '01': 309}
> Counts for pub 164: {'01': 584, '10': 320, '11': 62, '00': 58}
> Counts for pub 165: {'10': 222, '11': 357, '00': 364, '01': 81}
> Counts for pub 166: {'10': 606, '01': 230, '00': 148, '11': 40}
> Counts for pub 167: {'01': 300, '11': 271, '00': 262, '10': 191}
> Counts for pub 168: {'01': 490, '10': 448, '11': 73, '00': 13}
> Counts for pub 169: {'00': 327, '01': 183, '10': 195, '11': 319}
> Counts for pub 170: {'01': 439, '11': 88, '10': 419, '00': 78}
> Counts for pub 171: {'01': 351, '10': 505, '00': 103, '11': 65}
> Counts for pub 172: {'10': 497, '01': 412, '11': 55, '00': 60}
> Counts for pub 173: {'11': 424, '00': 362, '10': 145, '01': 93}
> Counts for pub 174: {'10': 511, '01': 352, '00': 117, '11': 44}
> Counts for pub 175: {'11': 87, '01': 471, '10': 440, '00': 26}
> Counts for pub 176: {'10': 362, '01': 386, '11': 131, '00': 145}
> Counts for pub 177: {'00': 383, '11': 371, '10': 190, '01': 80}
> Counts for pub 178: {'10': 229, '00': 366, '11': 349, '01': 80}
> Counts for pub 179: {'10': 706, '01': 96, '00': 147, '11': 75}
> Counts for pub 180: {'01': 527, '10': 424, '00': 14, '11': 59}
> Counts for pub 181: {'00': 412, '10': 267, '11': 266, '01': 79}
> Counts for pub 182: {'10': 450, '11': 25, '01': 373, '00': 176}
> Counts for pub 183: {'10': 253, '00': 354, '11': 345, '01': 72}
> Counts for pub 184: {'00': 135, '10': 744, '01': 22, '11': 123}
> Counts for pub 185: {'11': 301, '00': 406, '10': 225, '01': 92}
> Counts for pub 186: {'11': 58, '10': 653, '01': 227, '00': 86}
> Counts for pub 187: {'11': 364, '10': 366, '00': 200, '01': 94}
> Counts for pub 188: {'10': 192, '11': 339, '01': 244, '00': 249}
> Counts for pub 189: {'00': 461, '11': 530, '01': 23, '10': 10}
> Counts for pub 190: {'00': 111, '10': 712, '11': 165, '01': 36}
> Counts for pub 191: {'00': 339, '01': 107, '11': 447, '10': 131}
> Counts for pub 192: {'10': 206, '00': 369, '01': 91, '11': 358}
> Counts for pub 193: {'11': 370, '01': 116, '00': 370, '10': 168}
> Counts for pub 194: {'01': 354, '10': 463, '11': 71, '00': 136}
> Counts for pub 195: {'00': 66, '11': 51, '10': 502, '01': 405}
> Counts for pub 196: {'01': 584, '11': 146, '00': 57, '10': 237}
> Counts for pub 197: {'00': 700, '11': 128, '10': 157, '01': 39}
> Counts for pub 198: {'10': 395, '11': 93, '01': 486, '00': 50}
> Counts for pub 199: {'11': 404, '00': 346, '01': 104, '10': 170}
```

[ ]:
```python
import matplotlib.pyplot as plt

# Suponiendo que ya tienes los resultados en la variable `result`
```

```python
counts_data = []
for idx, pub_result in enumerate(result):
    counts = pub_result.data.meas.get_counts()
    counts_data.append(counts)
    print(f" > Counts for pub {idx}: {counts}")

# Extraer los recuentos para '00' de cada publicación y calcular la suma total␣
 ↪de recuentos por publicación
counts_00 = []
for counts in counts_data:
    total_counts = sum(counts.values())
    count_00 = counts.get('00', 0)
    count_00_pu = count_00 / total_counts if total_counts != 0 else 0  # Evitar␣
 ↪división por cero
    counts_00.append(count_00_pu)
"""
# Imprimir los recuentos de '00' en p.u. para verificar
for idx, count in enumerate(counts_00):
    print(f"Counts of '00' in p.u. for pub {idx}: {count:.4f}")
"""
```

```
> Counts for pub 0: {'00': 393, '11': 538, '01': 74, '10': 19}
> Counts for pub 1: {'00': 388, '11': 563, '01': 53, '10': 20}
> Counts for pub 2: {'01': 472, '00': 305, '10': 151, '11': 96}
> Counts for pub 3: {'00': 399, '10': 303, '01': 280, '11': 42}
> Counts for pub 4: {'11': 425, '01': 166, '00': 361, '10': 72}
> Counts for pub 5: {'11': 534, '00': 410, '01': 57, '10': 23}
> Counts for pub 6: {'00': 236, '11': 240, '10': 400, '01': 148}
> Counts for pub 7: {'11': 447, '10': 126, '00': 311, '01': 140}
> Counts for pub 8: {'01': 535, '00': 187, '10': 237, '11': 65}
> Counts for pub 9: {'11': 378, '10': 377, '00': 191, '01': 78}
> Counts for pub 10: {'00': 332, '10': 110, '01': 146, '11': 436}
> Counts for pub 11: {'00': 342, '11': 452, '10': 70, '01': 160}
> Counts for pub 12: {'01': 294, '11': 245, '10': 441, '00': 44}
> Counts for pub 13: {'00': 399, '11': 537, '01': 71, '10': 17}
> Counts for pub 14: {'10': 499, '00': 312, '11': 85, '01': 128}
> Counts for pub 15: {'01': 811, '00': 151, '11': 30, '10': 32}
> Counts for pub 16: {'01': 140, '11': 444, '00': 369, '10': 71}
> Counts for pub 17: {'10': 479, '11': 38, '00': 281, '01': 226}
> Counts for pub 18: {'11': 537, '00': 396, '01': 71, '10': 20}
> Counts for pub 19: {'10': 310, '00': 220, '01': 328, '11': 166}
> Counts for pub 20: {'01': 682, '00': 94, '10': 236, '11': 12}
> Counts for pub 21: {'00': 366, '11': 553, '01': 82, '10': 23}
> Counts for pub 22: {'10': 724, '00': 107, '01': 80, '11': 113}
> Counts for pub 23: {'00': 347, '11': 510, '01': 121, '10': 46}
> Counts for pub 24: {'10': 441, '01': 248, '00': 237, '11': 98}
> Counts for pub 25: {'10': 76, '11': 432, '01': 155, '00': 361}
```

```
> Counts for pub 26: {'11': 553, '01': 53, '00': 403, '10': 15}
> Counts for pub 27: {'01': 714, '00': 191, '10': 32, '11': 87}
> Counts for pub 28: {'00': 413, '11': 521, '01': 74, '10': 16}
> Counts for pub 29: {'01': 785, '00': 122, '10': 94, '11': 23}
> Counts for pub 30: {'01': 865, '00': 113, '11': 35, '10': 11}
> Counts for pub 31: {'11': 576, '01': 68, '00': 360, '10': 20}
> Counts for pub 32: {'00': 359, '11': 494, '01': 131, '10': 40}
> Counts for pub 33: {'00': 339, '11': 462, '01': 141, '10': 82}
> Counts for pub 34: {'00': 375, '11': 538, '01': 85, '10': 26}
> Counts for pub 35: {'11': 554, '00': 379, '01': 72, '10': 19}
> Counts for pub 36: {'10': 275, '01': 584, '00': 140, '11': 25}
> Counts for pub 37: {'01': 704, '00': 117, '10': 160, '11': 43}
> Counts for pub 38: {'10': 453, '00': 247, '01': 237, '11': 87}
> Counts for pub 39: {'00': 126, '01': 515, '11': 323, '10': 60}
> Counts for pub 40: {'10': 496, '00': 276, '01': 161, '11': 91}
> Counts for pub 41: {'11': 530, '00': 396, '10': 16, '01': 82}
> Counts for pub 42: {'10': 433, '00': 243, '11': 145, '01': 203}
> Counts for pub 43: {'11': 467, '00': 382, '10': 52, '01': 123}
> Counts for pub 44: {'11': 519, '00': 397, '01': 82, '10': 26}
> Counts for pub 45: {'10': 462, '01': 162, '11': 105, '00': 295}
> Counts for pub 46: {'11': 524, '00': 406, '01': 60, '10': 34}
> Counts for pub 47: {'01': 285, '00': 261, '11': 341, '10': 137}
> Counts for pub 48: {'11': 411, '00': 342, '01': 158, '10': 113}
> Counts for pub 49: {'01': 536, '10': 332, '00': 138, '11': 18}
> Counts for pub 50: {'00': 390, '01': 122, '11': 459, '10': 53}
> Counts for pub 51: {'11': 542, '00': 388, '01': 72, '10': 22}
> Counts for pub 52: {'11': 451, '00': 325, '01': 137, '10': 111}
> Counts for pub 53: {'01': 94, '10': 425, '11': 291, '00': 214}
> Counts for pub 54: {'00': 394, '11': 532, '01': 78, '10': 20}
> Counts for pub 55: {'10': 60, '00': 343, '01': 520, '11': 101}
> Counts for pub 56: {'00': 271, '10': 326, '01': 405, '11': 22}
> Counts for pub 57: {'11': 162, '10': 258, '01': 323, '00': 281}
> Counts for pub 58: {'11': 381, '01': 193, '00': 317, '10': 133}
> Counts for pub 59: {'10': 450, '00': 281, '11': 143, '01': 150}
> Counts for pub 60: {'00': 368, '01': 72, '11': 562, '10': 22}
> Counts for pub 61: {'10': 668, '11': 183, '01': 46, '00': 127}
> Counts for pub 62: {'11': 557, '00': 391, '01': 56, '10': 20}
> Counts for pub 63: {'10': 106, '01': 139, '00': 331, '11': 448}
> Counts for pub 64: {'11': 428, '01': 167, '10': 113, '00': 316}
> Counts for pub 65: {'11': 526, '00': 419, '01': 62, '10': 17}
> Counts for pub 66: {'11': 528, '00': 352, '10': 40, '01': 104}
> Counts for pub 67: {'01': 171, '10': 436, '00': 391, '11': 26}
> Counts for pub 68: {'11': 33, '10': 775, '01': 70, '00': 146}
> Counts for pub 69: {'01': 243, '10': 446, '11': 109, '00': 226}
> Counts for pub 70: {'10': 614, '01': 148, '00': 210, '11': 52}
> Counts for pub 71: {'01': 70, '00': 405, '11': 526, '10': 23}
> Counts for pub 72: {'00': 379, '11': 562, '01': 63, '10': 20}
> Counts for pub 73: {'11': 428, '00': 370, '01': 164, '10': 62}
```

```
> Counts for pub 74: {'11': 572, '00': 352, '01': 79, '10': 21}
> Counts for pub 75: {'00': 162, '01': 191, '10': 605, '11': 66}
> Counts for pub 76: {'11': 79, '01': 489, '00': 287, '10': 169}
> Counts for pub 77: {'01': 76, '11': 526, '00': 394, '10': 28}
> Counts for pub 78: {'10': 294, '00': 226, '01': 451, '11': 53}
> Counts for pub 79: {'11': 443, '01': 142, '00': 374, '10': 65}
> Counts for pub 80: {'11': 534, '00': 403, '01': 71, '10': 16}
> Counts for pub 81: {'00': 299, '01': 163, '10': 472, '11': 90}
> Counts for pub 82: {'11': 508, '00': 426, '01': 67, '10': 23}
> Counts for pub 83: {'00': 172, '01': 685, '10': 148, '11': 19}
> Counts for pub 84: {'10': 232, '00': 374, '01': 361, '11': 57}
> Counts for pub 85: {'11': 434, '00': 357, '01': 154, '10': 79}
> Counts for pub 86: {'01': 74, '00': 249, '10': 690, '11': 11}
> Counts for pub 87: {'00': 379, '11': 541, '01': 81, '10': 23}
> Counts for pub 88: {'10': 129, '00': 313, '01': 487, '11': 95}
> Counts for pub 89: {'10': 378, '01': 238, '11': 87, '00': 321}
> Counts for pub 90: {'11': 440, '01': 149, '10': 72, '00': 363}
> Counts for pub 91: {'00': 270, '01': 590, '10': 114, '11': 50}
> Counts for pub 92: {'01': 72, '11': 537, '00': 398, '10': 17}
> Counts for pub 93: {'00': 550, '10': 233, '11': 96, '01': 145}
> Counts for pub 94: {'11': 529, '00': 404, '01': 71, '10': 20}
> Counts for pub 95: {'00': 397, '11': 543, '01': 64, '10': 20}
> Counts for pub 96: {'11': 451, '10': 90, '00': 341, '01': 142}
> Counts for pub 97: {'11': 241, '01': 344, '00': 425, '10': 14}
> Counts for pub 98: {'10': 748, '00': 193, '11': 55, '01': 28}
> Counts for pub 99: {'00': 534, '10': 211, '11': 228, '01': 51}
> Counts for pub 100: {'00': 432, '11': 499, '10': 27, '01': 66}
> Counts for pub 101: {'11': 70, '01': 614, '00': 283, '10': 57}
> Counts for pub 102: {'11': 526, '00': 408, '01': 72, '10': 18}
> Counts for pub 103: {'00': 339, '10': 351, '01': 260, '11': 74}
> Counts for pub 104: {'01': 322, '00': 380, '11': 306, '10': 16}
> Counts for pub 105: {'01': 524, '00': 87, '10': 403, '11': 10}
> Counts for pub 106: {'01': 331, '10': 310, '11': 32, '00': 351}
> Counts for pub 107: {'10': 515, '11': 36, '01': 313, '00': 160}
> Counts for pub 108: {'10': 269, '01': 468, '11': 81, '00': 206}
> Counts for pub 109: {'00': 298, '01': 398, '10': 247, '11': 81}
> Counts for pub 110: {'11': 313, '01': 515, '00': 173, '10': 23}
> Counts for pub 111: {'11': 460, '00': 314, '01': 136, '10': 114}
> Counts for pub 112: {'11': 56, '01': 225, '10': 638, '00': 105}
> Counts for pub 113: {'01': 418, '00': 266, '11': 129, '10': 211}
> Counts for pub 114: {'00': 335, '10': 583, '01': 63, '11': 43}
> Counts for pub 115: {'00': 363, '11': 454, '10': 56, '01': 151}
> Counts for pub 116: {'11': 17, '01': 642, '10': 241, '00': 124}
> Counts for pub 117: {'00': 381, '10': 262, '01': 332, '11': 49}
> Counts for pub 118: {'01': 412, '10': 490, '00': 108, '11': 14}
> Counts for pub 119: {'00': 329, '11': 383, '01': 218, '10': 94}
> Counts for pub 120: {'11': 548, '00': 401, '01': 58, '10': 17}
> Counts for pub 121: {'11': 534, '01': 58, '00': 415, '10': 17}
```

```
> Counts for pub 122: {'11': 399, '00': 141, '10': 429, '01': 55}
> Counts for pub 123: {'01': 351, '11': 125, '10': 258, '00': 290}
> Counts for pub 124: {'00': 364, '11': 513, '10': 60, '01': 87}
> Counts for pub 125: {'11': 450, '00': 348, '10': 60, '01': 166}
> Counts for pub 126: {'10': 762, '00': 203, '01': 17, '11': 42}
> Counts for pub 127: {'10': 693, '00': 137, '11': 172, '01': 22}
> Counts for pub 128: {'01': 706, '00': 184, '10': 82, '11': 52}
> Counts for pub 129: {'00': 266, '10': 604, '01': 127, '11': 27}
> Counts for pub 130: {'00': 379, '11': 552, '01': 72, '10': 21}
> Counts for pub 131: {'11': 567, '00': 359, '01': 71, '10': 27}
> Counts for pub 132: {'11': 520, '00': 403, '01': 79, '10': 22}
> Counts for pub 133: {'00': 170, '01': 760, '10': 80, '11': 14}
> Counts for pub 134: {'01': 155, '00': 367, '11': 423, '10': 79}
> Counts for pub 135: {'11': 316, '00': 283, '01': 182, '10': 243}
```

[ ]:
```
'\n# Imprimir los recuentos de \'00\' en p.u. para verificar\nfor idx, count in
enumerate(counts_00):\n    print(f"Counts of \'00\' in p.u. for pub {idx}:
{count:.4f}")\n'
```

[ ]: