Bayesian Network

```
from pgmpy.models import BayesianNetwork
from pgmpy.factors.discrete import TabularCPD
from pgmpy.inference import VariableElimination

# Step 1: Define the structure of the Bayesian Network
model = BayesianNetwork([
    ('B', 'A'), # Burglary causes Alarm
    ('E', 'A'), # Earthquake causes Alarm
    ('A', 'D'), # Alarm causes David to call
    ('A', 'S')  # Alarm causes Sophia to call
])

# Burglary CPD: P(B)
cpd_b = TabularCPD(variable='B', variable_card=2, values=[[0.999], [0.001]])
# Earthquake CPD: P(E)
cpd_e = TabularCPD(variable='E', variable_card=2, values=[[0.998], [0.002]])
# Alarm CPD: P(A | B, E)
cpd_a = TabularCPD(variable='A', variable_card=2,
                   values=[[0.999, 0.71, 0.06, 0.05],
                           [0.001, 0.29, 0.94, 0.95]],
                   evidence=['B', 'E'], evidence_card=[2, 2])
# David's call CPD: P(D | A)
cpd_d = TabularCPD(variable='D', variable_card=2,
                   values=[[0.99, 0.3], [0.01, 0.7]],
                   evidence=['A'], evidence_card=[2])
# Sophia's call CPD: P(S | A)
cpd_s = TabularCPD(variable='S', variable_card=2,
                   values=[[0.95, 0.4], [0.05, 0.6]],
                   evidence=['A'], evidence_card=[2])

# Step 3: Add CPDs to the model
model.add_cpds(cpd_b, cpd_e, cpd_a, cpd_d, cpd_s)

# Check if the model is valid
assert model.check_model()

# Step 4: Perform inference
inference = VariableElimination(model)

# Step 5: Compute the probability P(A | B=False, E=False, D=True, S=True)
result = inference.query(variables=['A'],  # Only query for A
                         evidence={'B': 0, 'E': 0, 'D': 1, 'S': 1})
print(result)
```