Anomaly Detection with Hidden Markov Model Reinforcement Learning

Group 15
CMPT 318
Fall 2023

Amy Cao
Christopher Peer
Khaled Taseen
Manmeet Singh

Two Sections

Part 1: Training of Hidden Markov models for the purpose of

unsupervised intrusion detection by analyzing stream data from a supervisory control system (household electricity consumption data)

Part 2: Implementing Q-Learning for Investment Decisions

Our reinforcement learning model employs the Q-learning algorithm, a model-free reinforcement learning technique used to find the optimal action-selection policy for a given Markov decision process. In our scenario, the agent makes investment decisions across various sectors like Stocks, Real Estate, Commodities, Cryptocurrencies, and Forex, with budgets ranging from 1 to 100 million dollars.

The Q-learning model's state space consists of integer values representing different budget levels. The actions correspond to different investment choices. The rewards are computed based on the profit generated from these investments, with profits being random numbers within specified limits. The Q-table, a central component of Q-learning, is initialized to zeros for all state-action pairs.

Part 1: **Unsupervised intrusion detection**

Part 1: Background

- Supervisory Control Systems are being adopted in many industrial sectors.
- The reliance on automation within critical sectors increases the attack surface for Advanced Persistent Threats (APT).
- APTs try to remain undetected for the longest possible period of time.
- Existing vulnerabilities expose critical infrastructure to adversarial scenarios.
- The attack space is increasing with the expanding digital space.
- Detecting the attack is a type of defense against cyber attacks.
- Anomaly-detection based intrusion detection methods are used for cyber situational awareness in the analysis of automated control processes.

Methodology

1. Clean and process data sets
2. Extract and Select Features using PCA
3. Process Feature Data
4. Train and test HMM to find the top-performing model
5. Make predictions on data confirmed to have anomalies

1. Clean and Process Data Sets

	A	B	C	D	E	F	G	H	I	J
1	Date	Time	Global_activ	Global_reac	Voltage	Global_inte	Sub_meteri	Sub_meteri	Sub_metering_3	
2	16/12/2006	17:24:00	NA	0.418	234.84	18.4	0	1	17	
3	16/12/2006	17:25:00	5.36	0.436	233.63	23	0	1	16	
4	16/12/2006	17:26:00	NA	0.498	233.29	23	0	2	17	
5	16/12/2006	17:27:00	5.388	0.502	233.74	23	0	1	17	
6	16/12/2006	17:28:00	NA	0.528	235.68	15.8	0	1	17	
7	16/12/2006	17:29:00	3.52	0.522	235.02	15	0	2	17	
8	16/12/2006	17:30:00	3.702	0.52	235.09	15.8	0	1	17	
9	16/12/2006	17:31:00	3.7	0.52	235.22	15.8	0	1	17	

	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3	timestamp
1	12/16/2006	17:24:00	5.36	0.418	234.84	18.4	0	1	17	2006-12-16 17:24:00
2	12/16/2006	17:25:00	5.36	0.436	233.63	23	0	1	16	2006-12-16 17:25:00
3	12/16/2006	17:26:00	5.374	0.498	233.29	23	0	2	17	2006-12-16 17:26:00
4	12/16/2006	17:27:00	5.388	0.502	233.74	23	0	1	17	2006-12-16 17:27:00
5	12/16/2006	17:28:00	4.454	0.528	235.68	15.8	0	1	17	2006-12-16 17:28:00
6	12/16/2006	17:29:00	3.52	0.522	235.02	15	0	2	17	2006-12-16 17:29:00
7	12/16/2006	17:30:00	3.702	0.52	235.09	15.8	0	1	17	2006-12-16 17:30:00
8	12/16/2006	17:31:00	3.7	0.52	235.22	15.8	0	1	17	2006-12-16 17:31:00
9	12/16/2006	17:32:00	3.668	0.51	233.99	15.8	0	1	17	2006-12-16 17:32:00

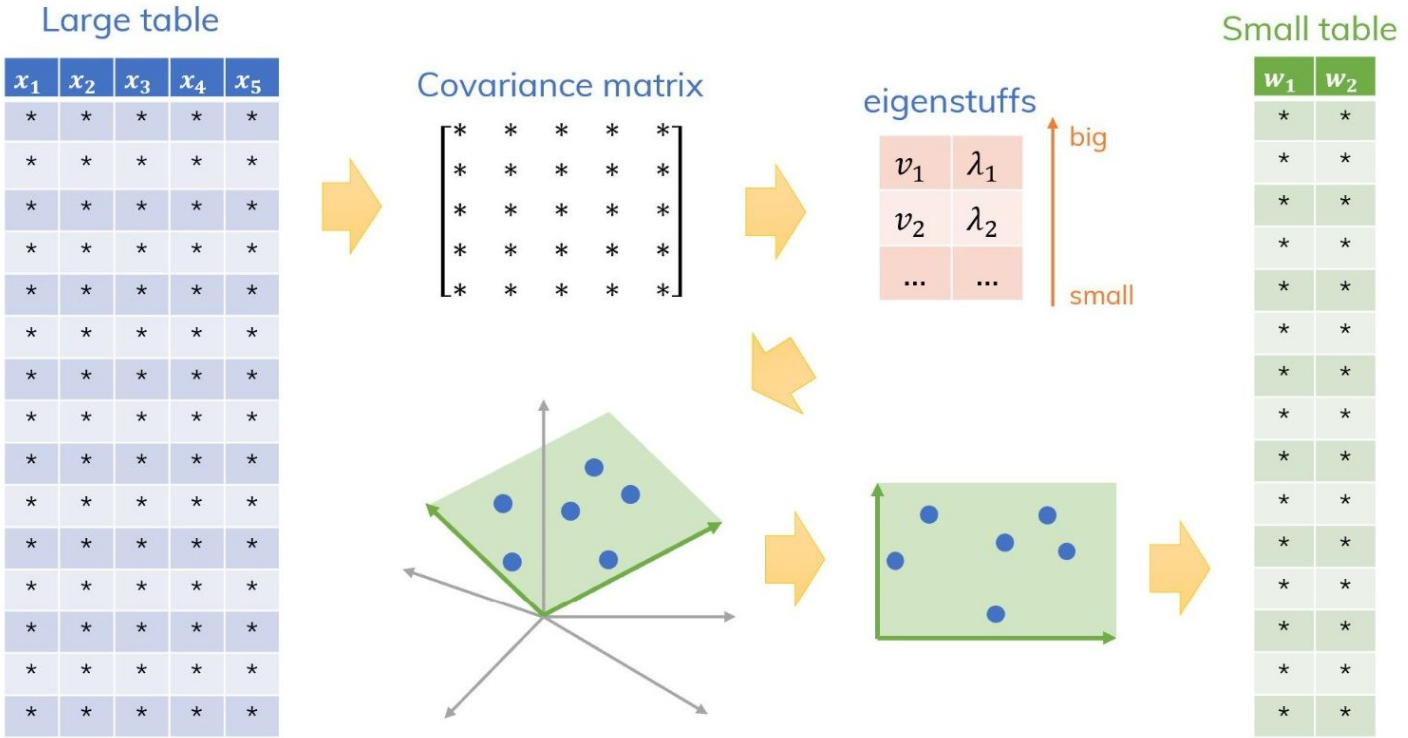
1. Clean and Process Data Sets

	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3	timestamp
1	12/16/2006	17:24:00	5.36	0.418	234.84	18.4	0	1	17	2006-12-16 17:24:00
2	12/16/2006	17:25:00	5.36	0.436	233.63	23	0	1	16	2006-12-16 17:25:00
3	12/16/2006	17:26:00	5.374	0.498	233.29	23	0	2	17	2006-12-16 17:26:00
4	12/16/2006	17:27:00	5.388	0.502	233.74	23	0	1	17	2006-12-16 17:27:00
5	12/16/2006	17:28:00	4.45							
6	12/16/2006	17:29:00	3.5							
7	12/16/2006	17:30:00								
8	12/16/2006	17:31:00								
9	12/16/2006	17:32:00								

$$z = (value - mean)/standarddeviation$$

	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
						1	17
						1	16
						2	17
						1	17
						1	17
6	3.52	0.522	235.02	15	0	2	17
7	3.702	0.52	235.09	15.8	0	1	17
8	3.7	0.52	235.22	15.8	0	1	17
9	3.668	0.51	233.99	15.8	0	1	17

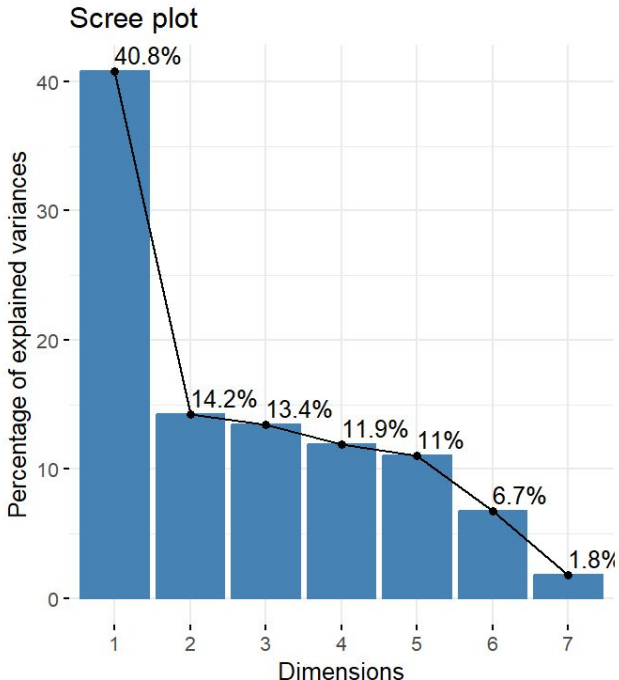
2. Select Features Using PCA

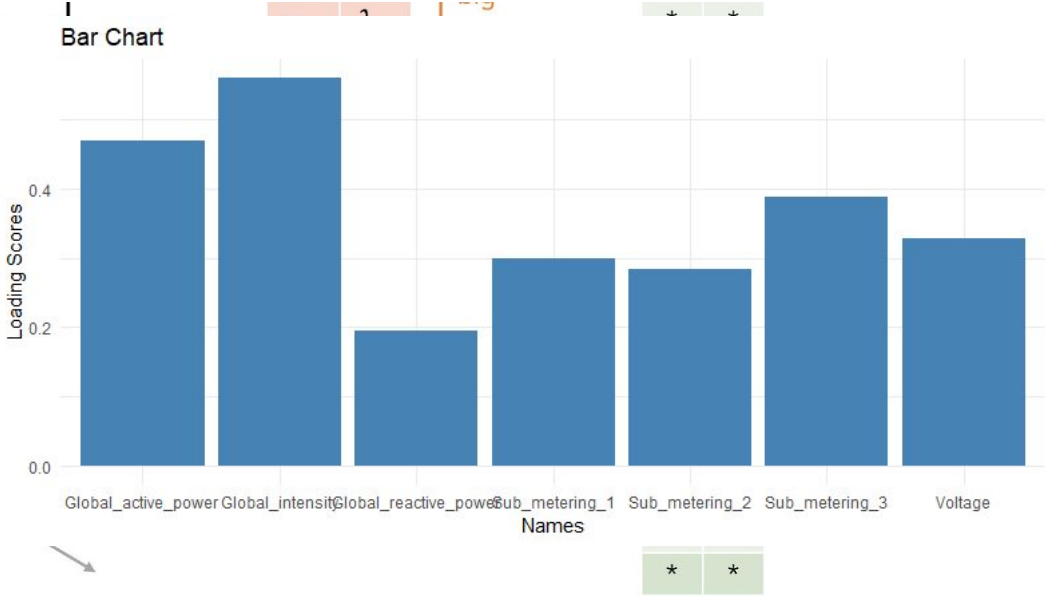


Source: https://datascience.stackexchange.com

2. Select Features Using PCA

```
Importance of components:
                          PC1     PC2     PC3     PC4     PC5     PC6     PC
Standard deviation     1.6899 0.9987 0.9697 0.9138 0.8793 0.68722 0.3553
Proportion of Variance 0.4079 0.1425 0.1343 0.1193 0.1104 0.06747 0.0180
Cumulative Proportion  0.4079 0.5504 0.6847 0.8041 0.9145 0.98196 1.0000
```



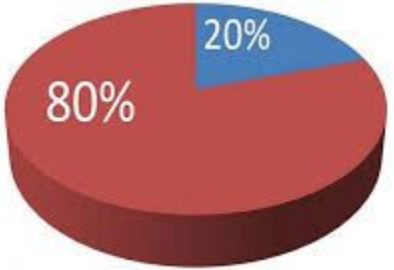


Source: https://datascience.stackexchange.com

3. Process Feature Data

a. Obtain Time Window of Data

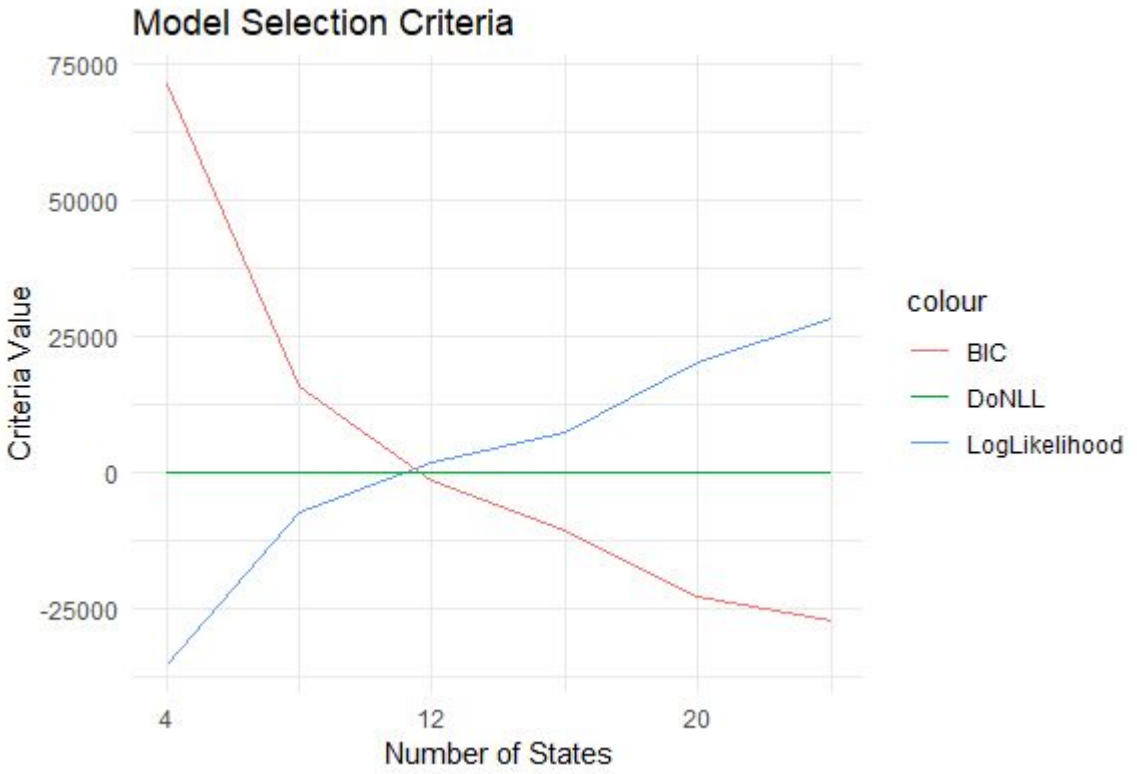
B. And split Data into Train/Test



4. Training HMM to Find Top-Performing Model

By evaluating BIC and Log-likelihood

We



Evaluating on test data

We extracted the response variables from our test data and fit them to a model. Then transferred to parameters from our trained model to the new test model. Then the forward backward function was run to give us forward-backward probabilities (and other model components) from which the log-likelihood of the fitment of the test responses could be extracted. A higher log-likelihood is better for the validity of our model.

1) Test Log Like : 2097.12
2) Normalised Log Like : 0.2008735

As observed the log-likelihood is sufficient and indicates our model is well fit and generalisable to non-anomalous data.

5. Make Predictions on Data Confirmed to Have Anomalies

Log likelihood of the anomalous datasets were as followed, computed in a similar manner to the test data fitment. Low log-likelihood indicates bad fitment on anomalous data

Anom1: -13654.342

Anom2: -25436.975

Anom3: -126573.636

As observed, all these datasets are anomalous. Particularly Anom3 which on some runs wouldn't fit correctly after forward backwards prediction and would return Nan for log-likelihood.

Part 2: **Implementing Q-Learning for Investment Decisions**

Implementing Q-Learning for Investment Decisions

- Objective: To develop a Q-learning model capable of making strategic investment decisions.

- State Space: Budget levels ranging from 1 to 100.

- Actions: Different investment options (Stocks, Real Estate, Commodities, Cryptocurrencies, Forex).

- Rewards: Based on simulated profits from investment decisions.

Dynamics of the Q-Learning Algorithm

- Hyperparameters:
  - Alpha (α): 0.1, balancing new vs. past experiences.
  - Gamma (γ): 0.9, emphasizing the value of future rewards.
- Training Process:
  - Description of generating training data from simulated experiences.
  - Iterative Q-table update process over multiple epochs.

Insights from the Trained Q-Learning Model

- Q-Table: Represents learned value of actions at different budget levels.

- Derived Policy: Indicates the optimal action for each state.

- Application to Unseen Data:

  - Showcases the model's ability to predict optimal actions for new scenarios.

  - Table with a sample of optimal actions for specific unseen states.

Application on Unseen Data (Action State Table)

TABLE II
OPTIMAL ACTIONS FOR UNSEEN STATES

State	Optimal Action	State	Optimal Action
15	Real Estate	31	Real Estate
16	Stocks	32	Cryptocurrencies
17	Forex	33	Real Estate
18	Forex	34	Real Estate
19	Forex	35	Real Estate
20	Commodities	36	Stocks
21	Commodities	37	Commodities
22	Stocks	38	Commodities
23	Commodities	39	Cryptocurrencies
24	Commodities	40	Forex
25	Cryptocurrencies	41	Forex
26	Stocks	42	Stocks
27	Real Estate	43	Real Estate
28	Cryptocurrencies	44	Real Estate
29	Commodities	45	Stocks
30	Commodities		

Conclusion of Q-Learning Implementation

-Our Q-learning model has established a strategic framework for making investment decisions, tailoring action choices to varying budget levels with notable success.

-By fine-tuning the learning rate (α), discount factor (γ), the model achieved a delicate balance that maximized reward over time.

-The Q-table constructed by the model encapsulates a comprehensive strategy, suggesting optimal investment decisions that are both data-driven and adaptable.

-The model's effectiveness was further underscored when applied to unseen data, showcasing its ability to generalize and potentially inform real-world financial decision-making processes.

-This exploration into reinforcement learning has opened avenues for automating investment strategy formulation, underscoring the transformative potential of machine learning in finance.

-The implementation of the Q-learning model marks a significant milestone in leveraging artificial intelligence for financial applications, setting the stage for future innovations in the sector.