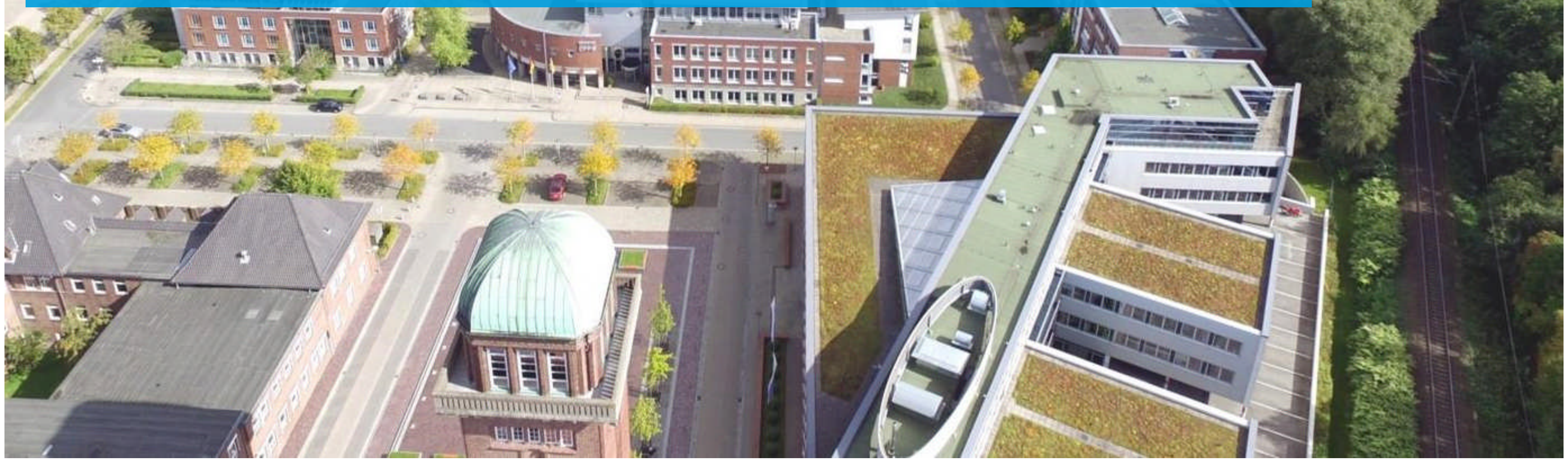
Offline Reinforcement Learning Agents for Adaptive Reactive Power Control with Renewable Energy Sources

email: tejashri.bhatt@offis.de

Tejashri Bhatt, Stephan Balduin, Eric MSP Veith

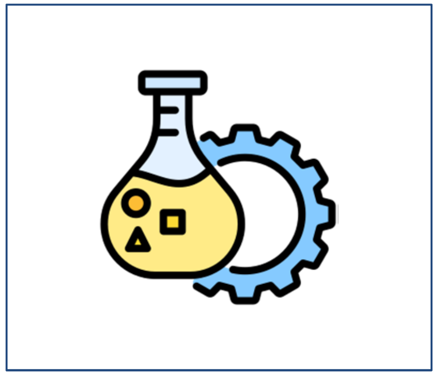
11th March 2025



Background

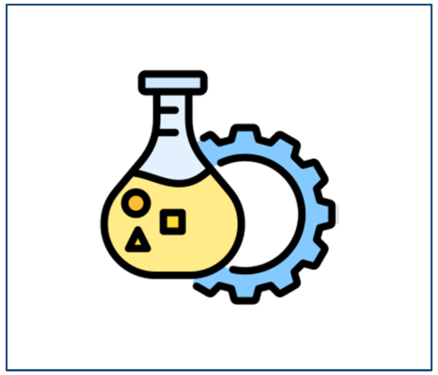
tejashribhatt@gmail.com

Background



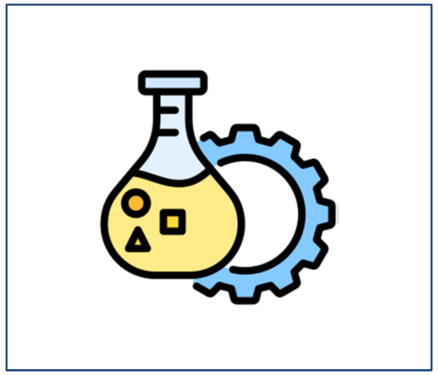
tejashribhatt@gmail.com

Background



tejashribhatt@gmail.com

Background

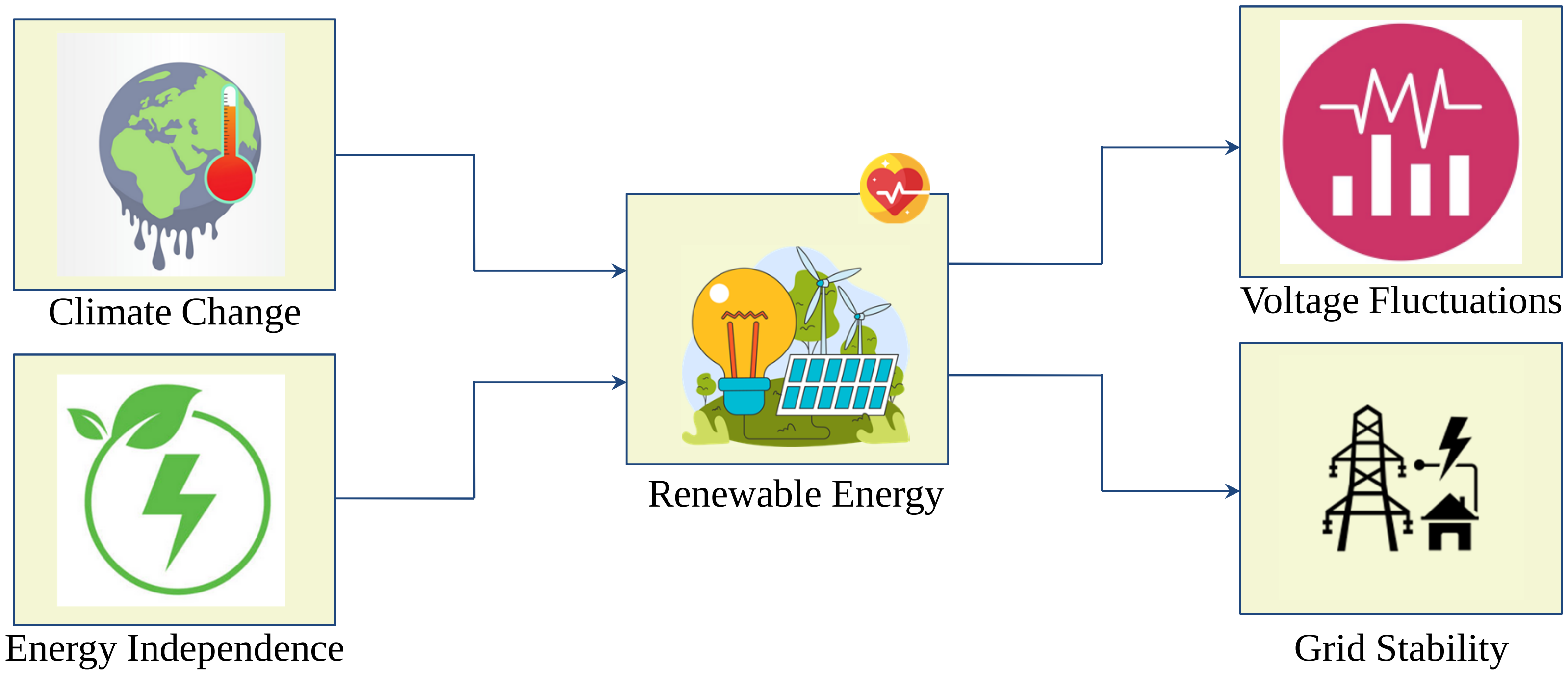


tejashribhatt@gmail.com

Table of Contents

Table of Contents

Motivation


Climate Change
Energy Independence
Renewable Energy
Voltage Fluctuations
Grid Stability

Research Question

Research Question

Reactive power control
via
ANN agent

Research Question

Reactive power control
via
ANN agent

Influence of ANN
on
experiment performance

Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Research Question

Reactive power control
via
ANN agent

Influence of ANN
on
experiment performance

Influence of BCO
on
sample efficiency

Research Question

Reactive power control
via
ANN agent

Influence of ANN
on
experiment performance

Influence of BCO
on
sample efficiency

Use 'expert' knowledge of an operator in executing reactive power control for a PV farm.

Research Question

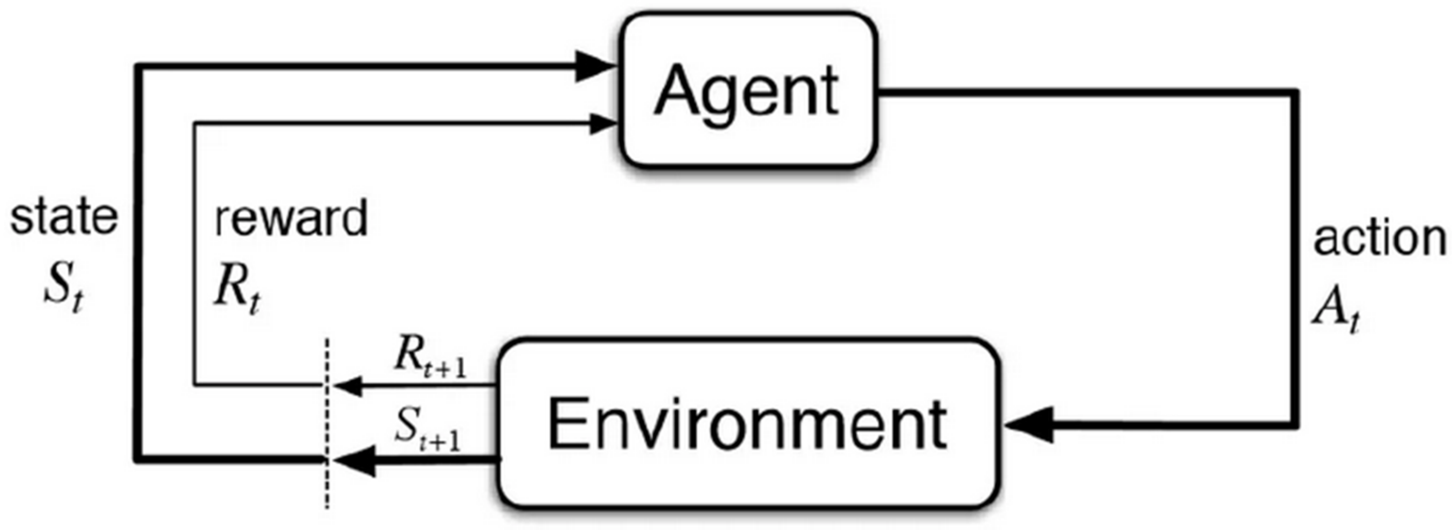
Reactive power control via ANN agent	Influence of ANN on experiment performance	Influence of BCO on sample efficiency

Use 'expert' knowledge of an operator in executing reactive power control for a PV farm.

Enhance grid resilience by empowering operators to make informed decisions.

Table of Contents

Reinforcement Learning



Sutton and Barto - "Learning what to do, through trial and error "

Algorithms used

- Soft-Actor Critic
- Behavior Cloning from Observation

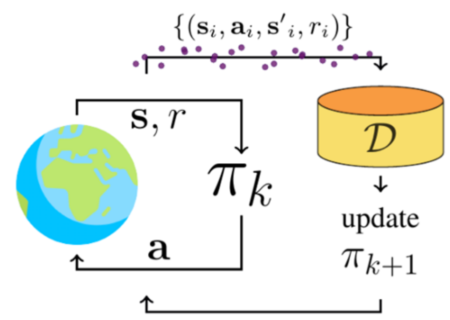
Soft-Actor Critic - Application of Reinforcement Learning

Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Off-Policy

Soft-Actor Critic - Application of Reinforcement Learning

Off-Policy



off-policy reinforcement learning

Soft-Actor Critic - Application of Reinforcement Learning

Off-Policy

Objective:
Max (Reward + Entropy)

Soft-Actor Critic - Application of Reinforcement Learning

Off-Policy

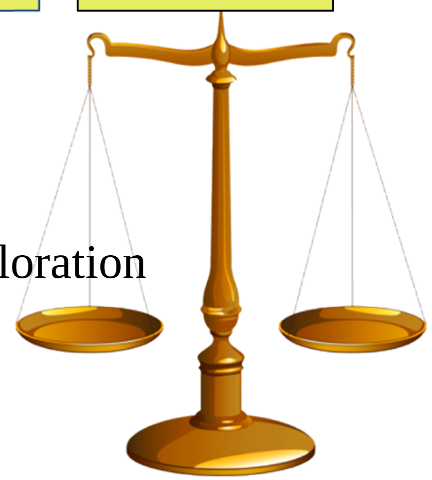
Objective:
Max (Reward + Entropy)

Introduction → Algorithms → System Description → Methodology → Evaluation → Conclusion

Soft-Actor Critic - Application of Reinforcement Learning

Off-Policy

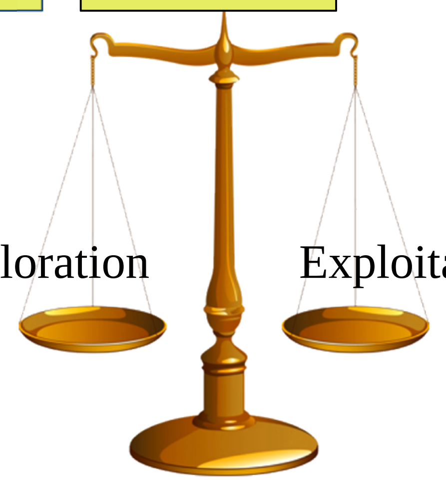
Objective:
Max (Reward + Entropy)



Soft-Actor Critic - Application of Reinforcement Learning

Off-Policy

Objective:
Max (Reward + Entropy)



Exploration

Exploitation

Soft-Actor Critic - Application of Reinforcement Learning

Off-Policy

Objective:
Max (Reward + Entropy)

Actor-Critic algorithm

Soft-Actor Critic - Application of Reinforcement Learning

Off-Policy

Objective:
Max (Reward + Entropy)

Actor-Critic algorithm

Drawback:

- Starts with random policy $\boldsymbol{\pi}$ initialization

OFFIS

Off-Policy	Objective: Max (Reward + Entropy)	Actor-Critic algorithm

Drawback:

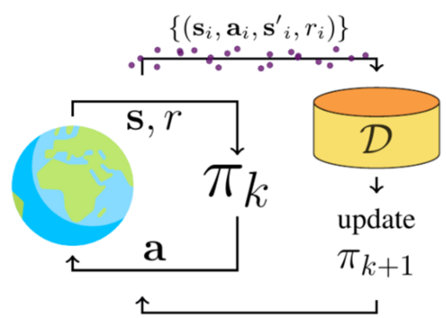
- Starts with random policy π initialization
- High number of initial interactions to reach good π

Soft-Actor Critic - Application of Reinforcement Learning

Off-Policy

Objective:
Max (Reward + Entropy)

Actor-Critic algorithm



off-policy reinforcement learning

$\boldsymbol{\pi}$ random initialization!

Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Soft-Actor Critic - Application of Reinforcement Learning

Off-Policy

Objective:
Max (Reward + Entropy)

Actor-Critic algorithm

Drawback:

- Starts with random policy π initialization
- High number of initial interactions to reach good π

Solution:
Behavior Cloning from Observation (BCO)

BCO - Application of Reinforcement Learning

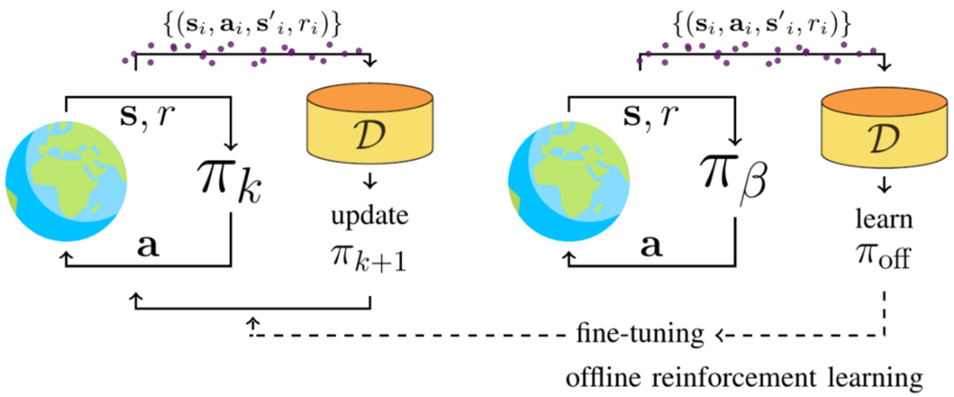
Introduction
Algorithms
System Description
Methodology
Evaluation
Conclusion

BCO - Application of Reinforcement Learning

Offline

BCO - Application of Reinforcement Learning

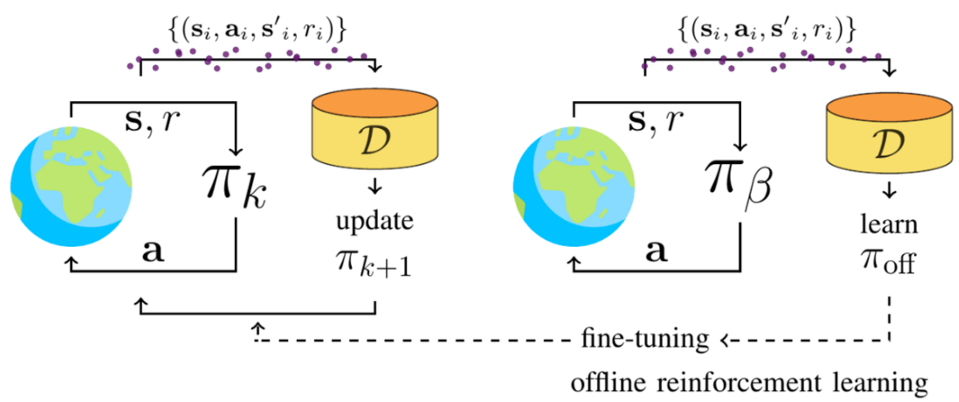
Offline



BCO - Application of Reinforcement Learning

Offline

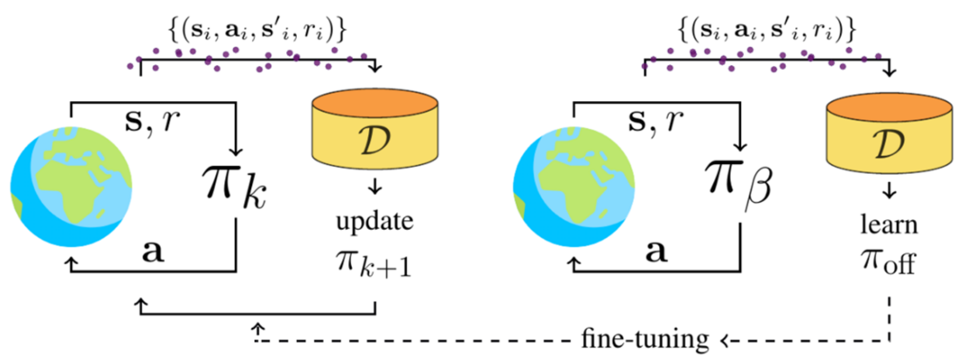
Initial $\boldsymbol{\pi}$ learnt offline



BCO - Application of Reinforcement Learning

Offline

Initial $\boldsymbol{\pi}$ learnt offline → No real-time interactions for good $\boldsymbol{\pi}$



offline reinforcement learning

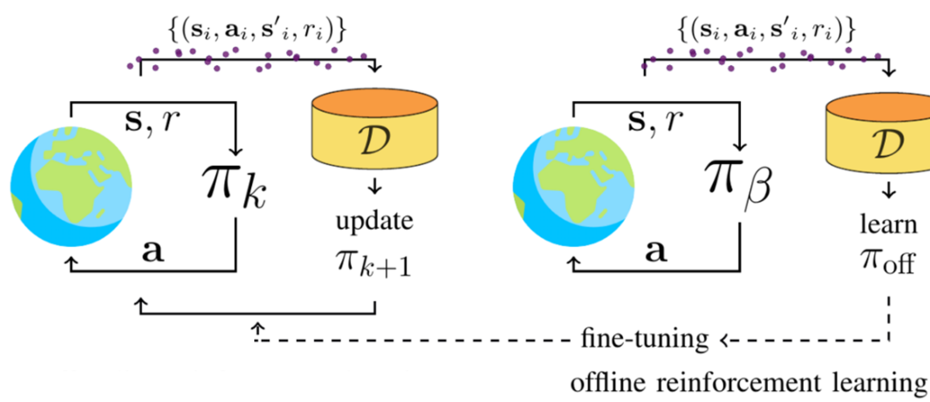
Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

BCO - Application of Reinforcement Learning

Offline

Initial $\boldsymbol{\pi}$ learnt offline → No real-time interactions for good $\boldsymbol{\pi}$

Head start!



BCO - Application of Reinforcement Learning

Offline

Objective:
Maximum reward

BCO - Application of Reinforcement Learning

Offline

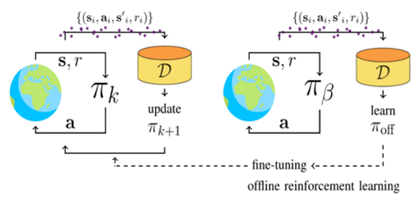
Objective:
Maximum reward

Drawback:
Good quality data

BCO - Application of Reinforcement Learning

Offline

Objective:
Maximum reward



Drawback:
Good quality data

BCO - Application of Reinforcement Learning

Offline

Objective:
Maximum reward

Drawback:
Good quality data

Solution:
Q-Controller equation!

BCO - Application of Reinforcement Learning

Offline

Objective:
Maximum reward

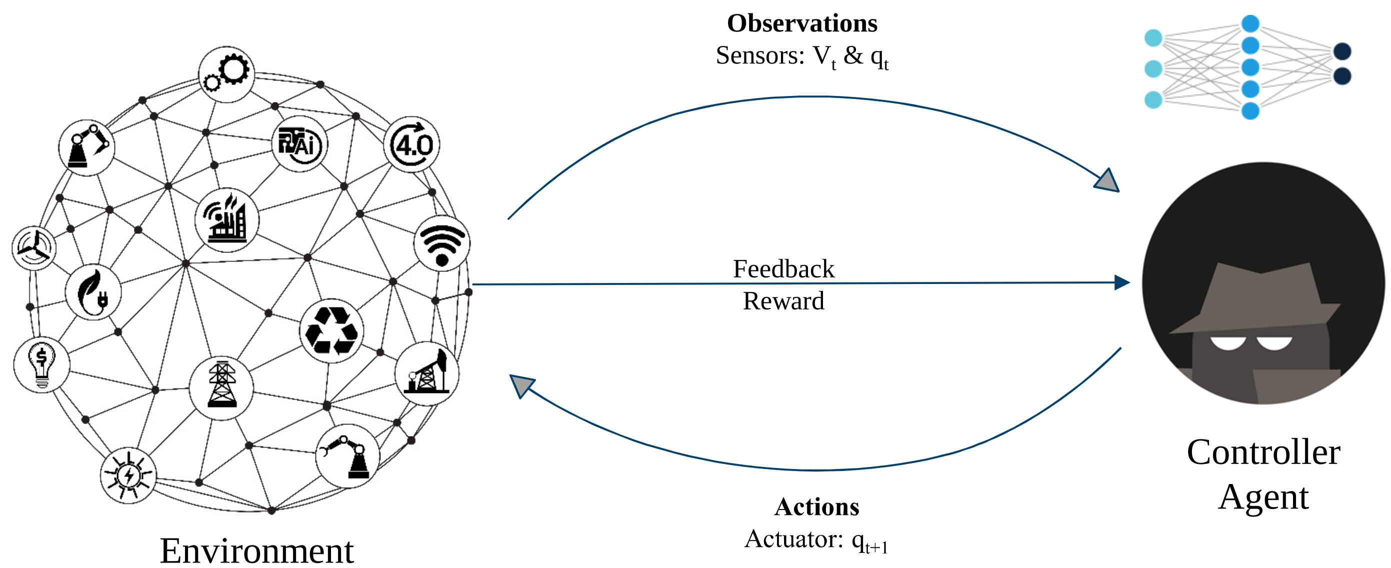
Drawback:
Good quality data

Solution:
Q-Controller equation!

$$q_{t+1} = [q_t - D(V_t - 1)]^+$$

Table of Contents

Set-Up



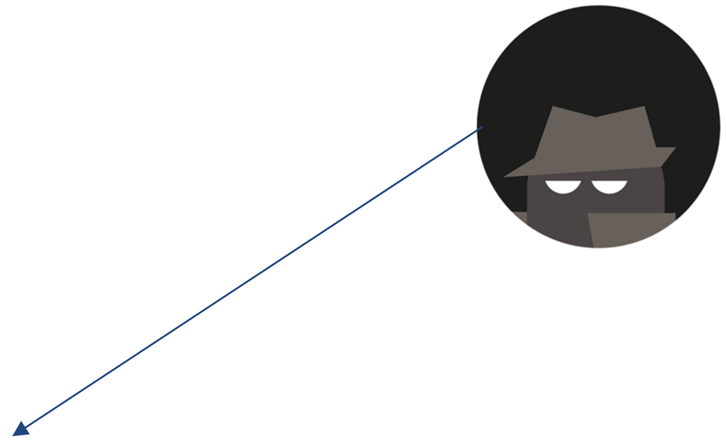
Observations
Sensors: V_t & q_t
Feedback
Reward
Actions
Actuator: q_{t+1}
Environment
Controller Agent

Scenarios

Controlling Agent

Scenarios

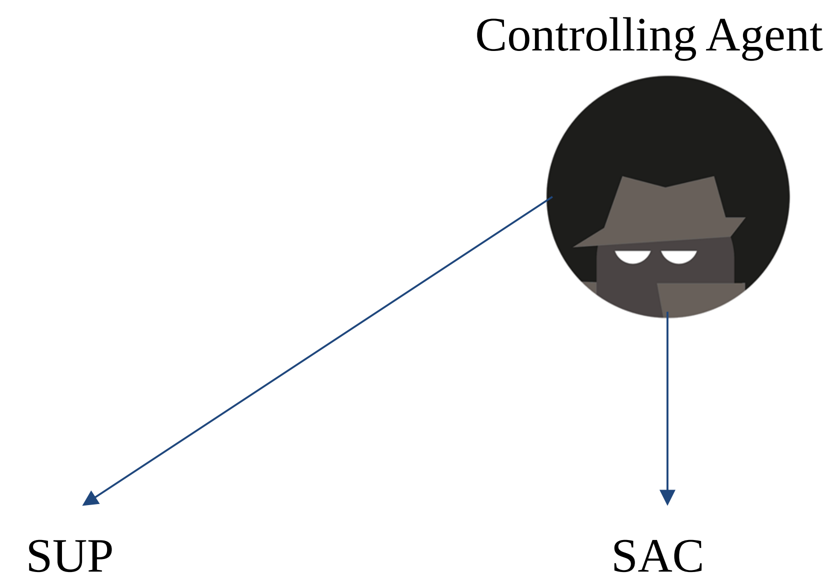
Controlling Agent



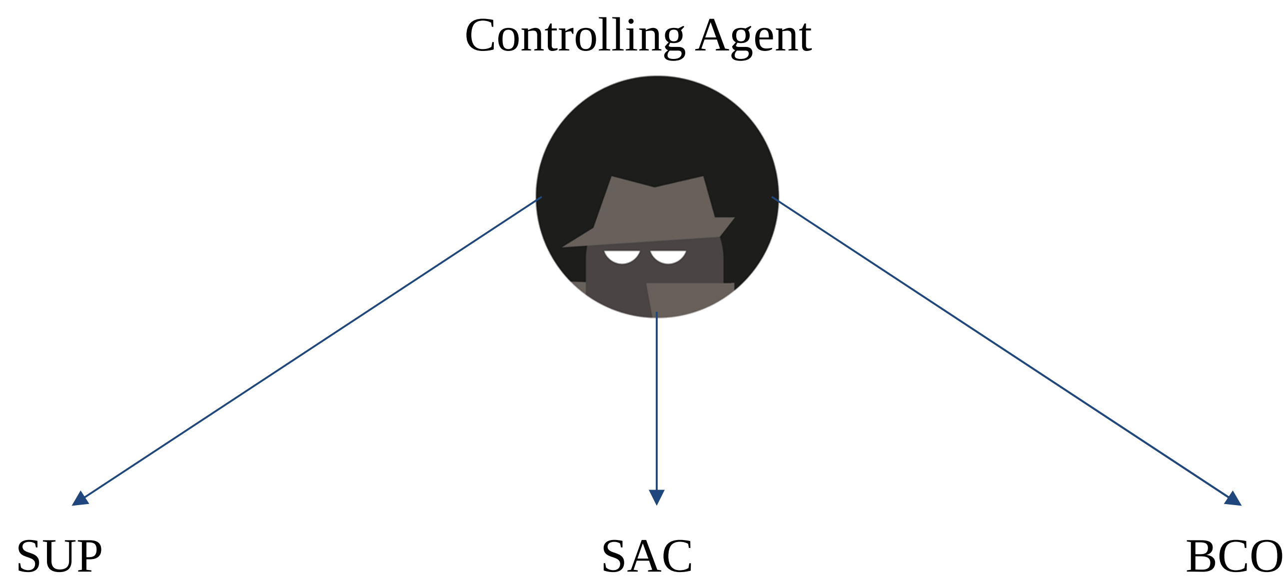
SUP

Introduction Algorithms System Description Methodology Evaluation Conclusion

Scenarios



Scenarios



Scenarios

Objective Function
- Voltage levels of all buses on the grid
- Voltage level of observed bus
- Operational buses unaffected by grid code violations

SUP SAC BCO

Scenarios

Environment

- Weather Data: Bremen 2020
- 20 kV MV Grid
- 14 Buses- Total Power Capacity of 2000 kW
  - Households (load + PV generation)
  - Supermarket (load)
  - Small Hotel (load)

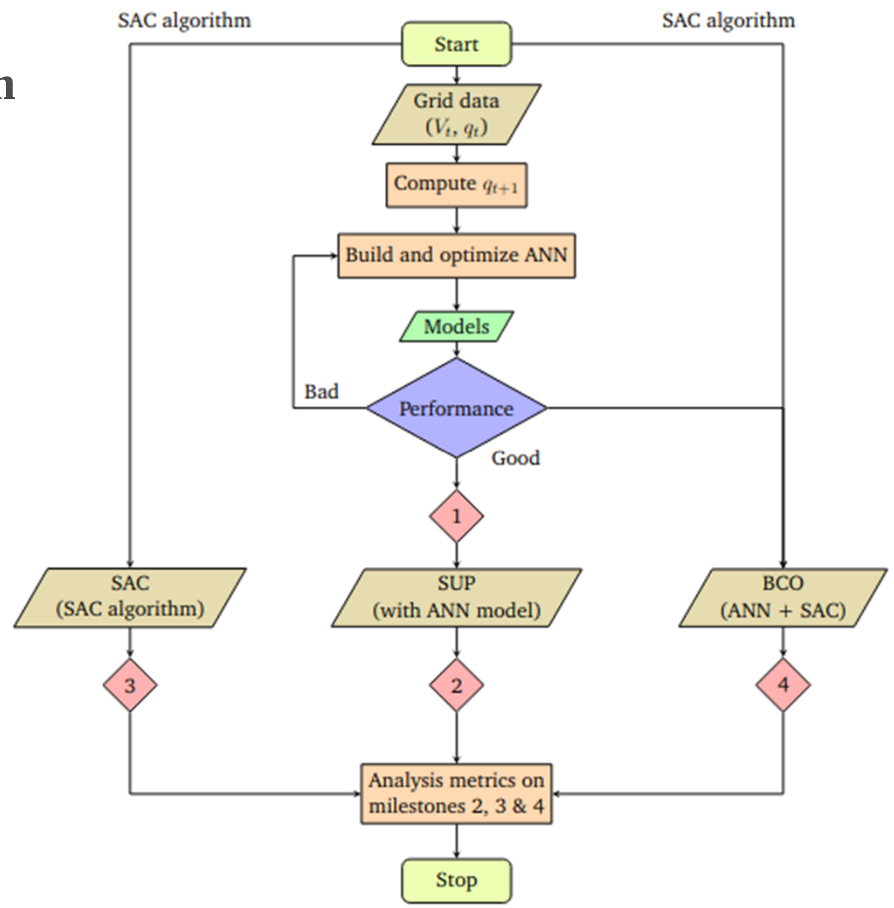
SUP

SAC

BCO

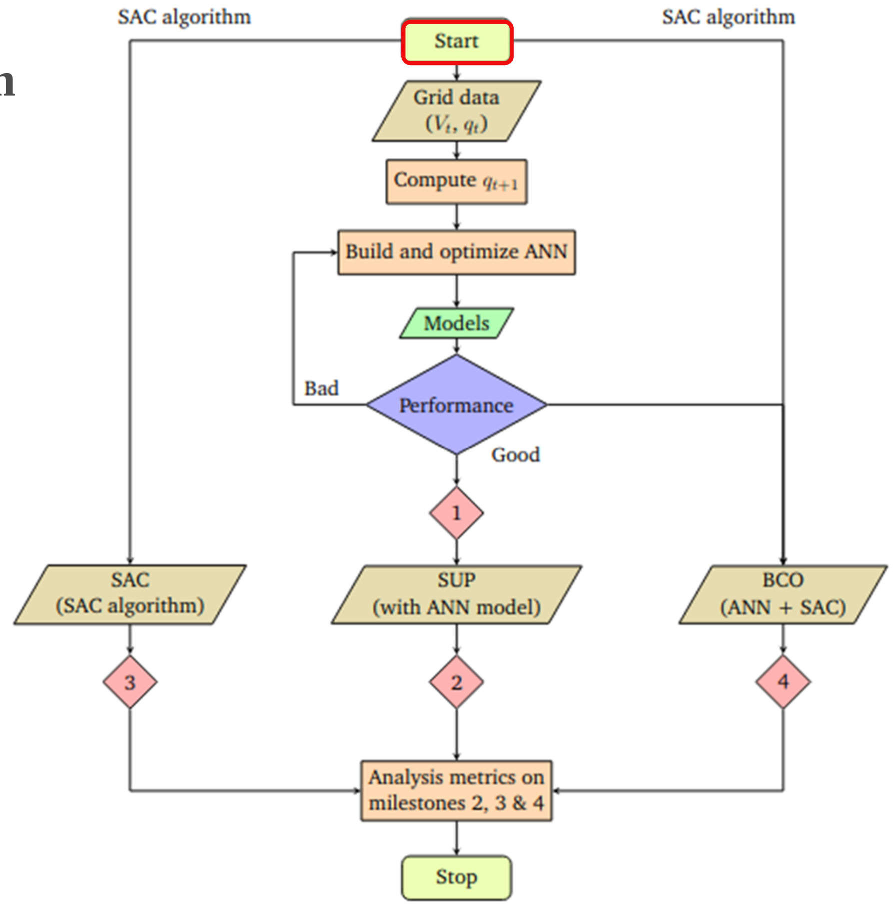
Table of Contents

Process Flow Diagram

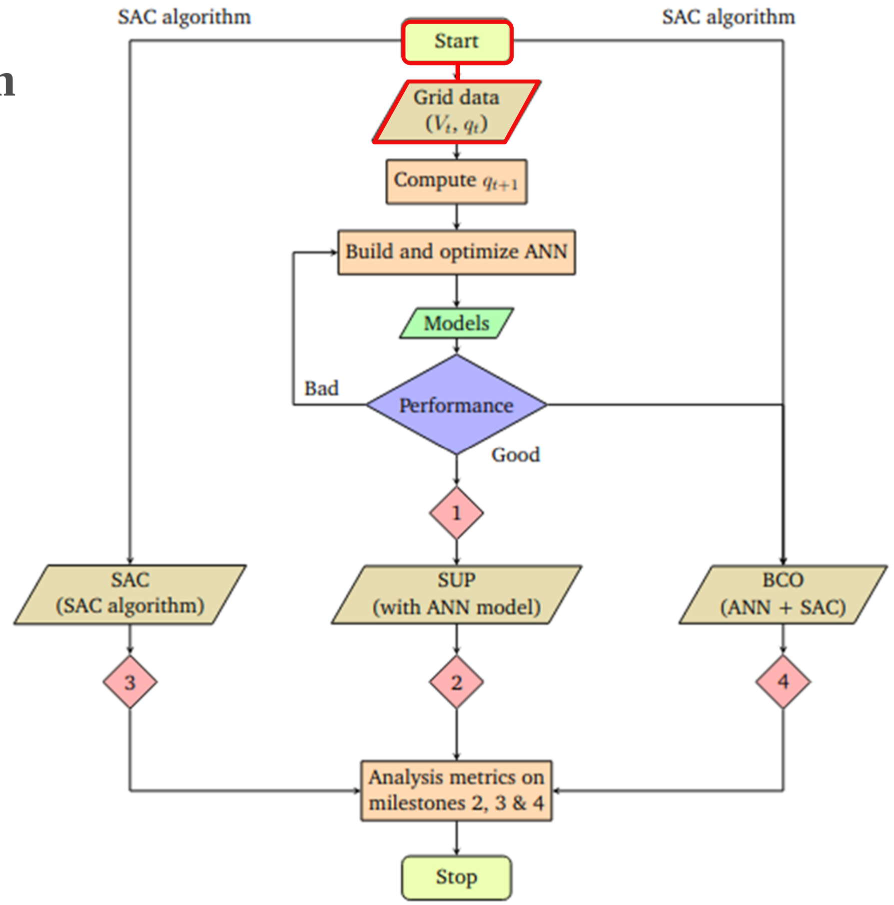


Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Process Flow Diagram



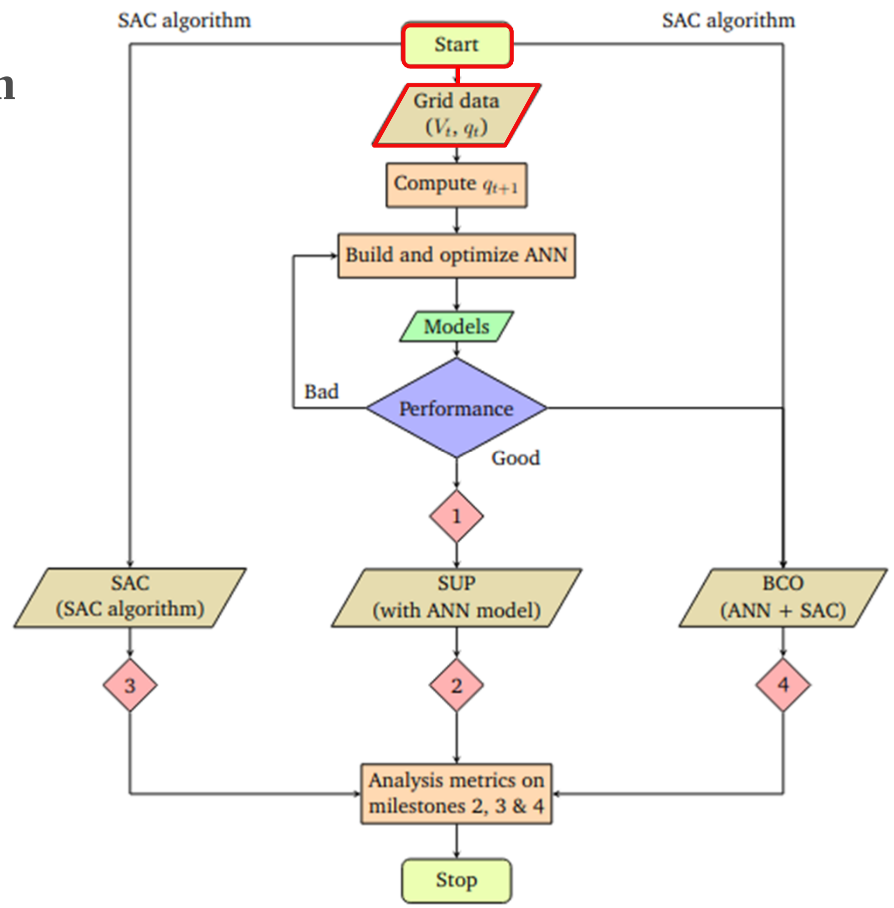
Process Flow Diagram



Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

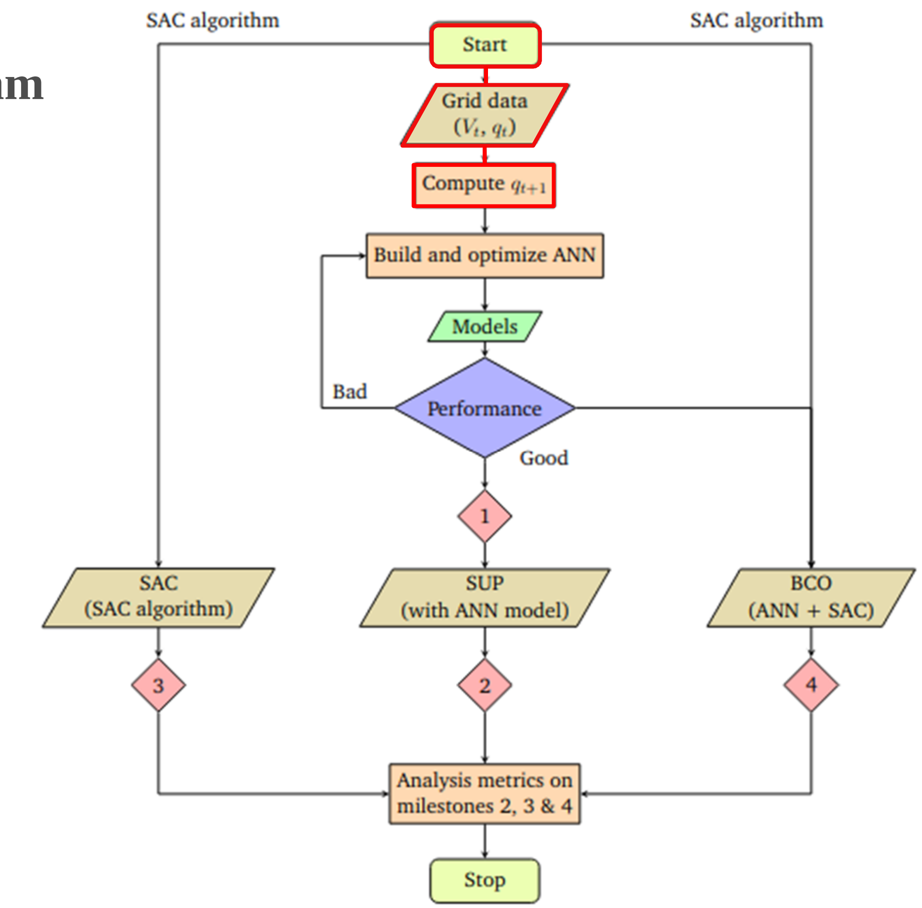
Process Flow Diagram

MIDAS



Introduction → Algorithms → System Description → Methodology → Evaluation → Conclusion

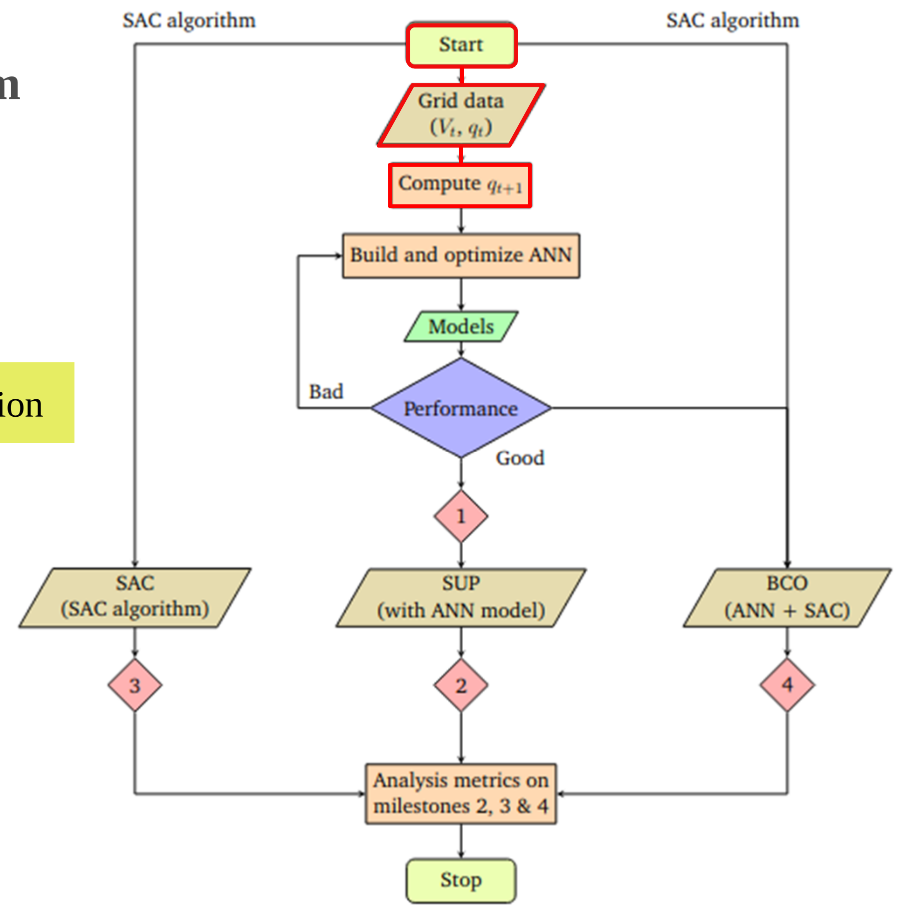
Process Flow Diagram



Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Process Flow Diagram

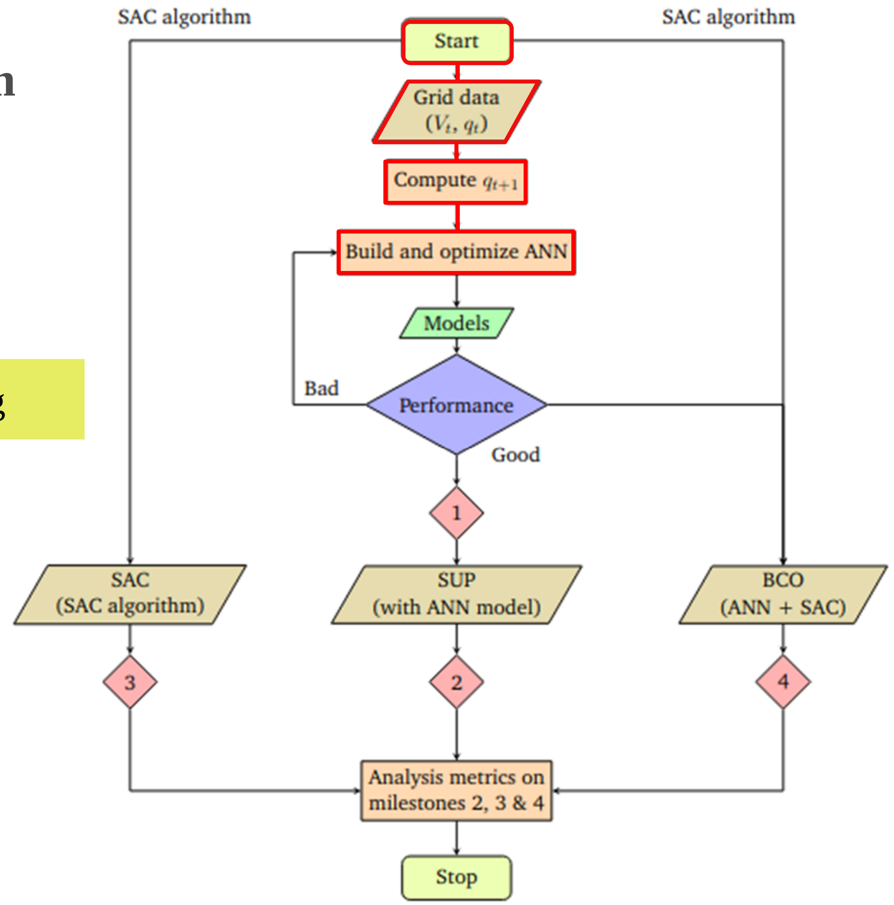
Q-Controller equation



Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

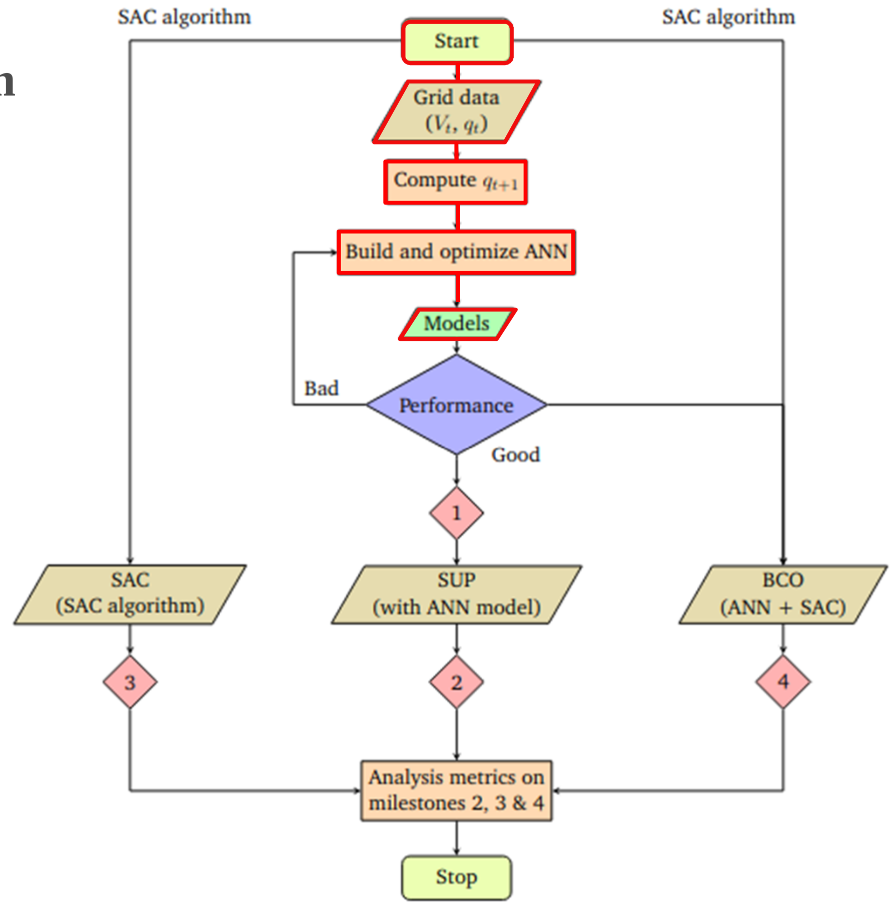
Process Flow Diagram

Supervised learning



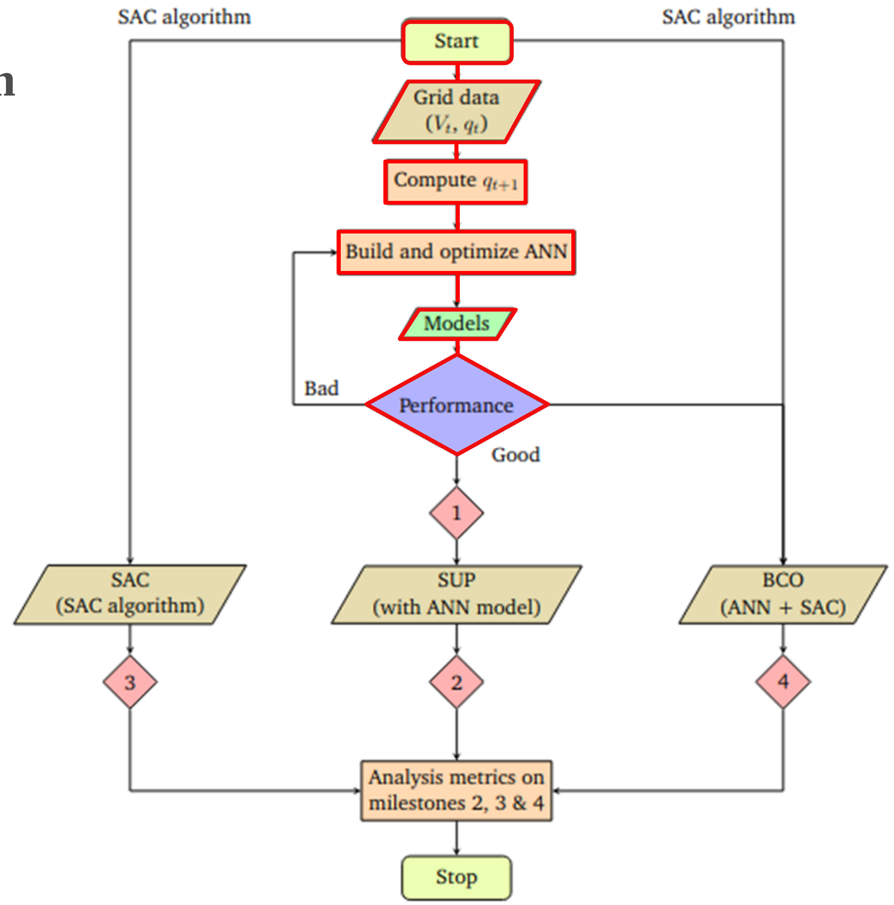
Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Process Flow Diagram

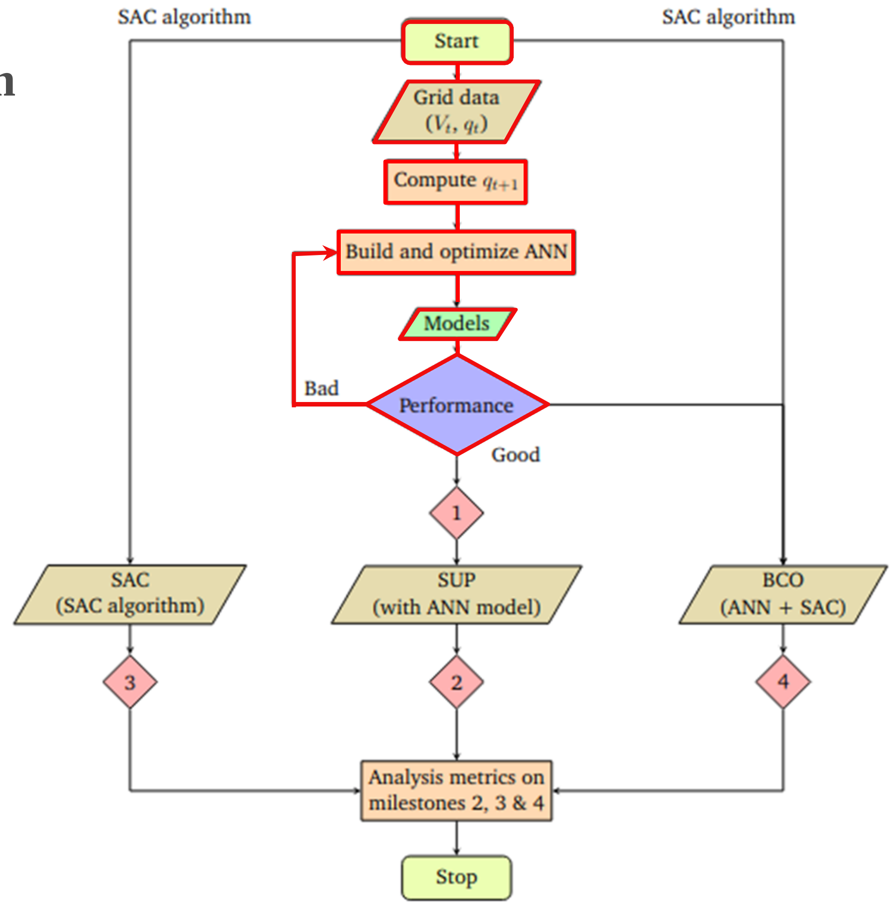


Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

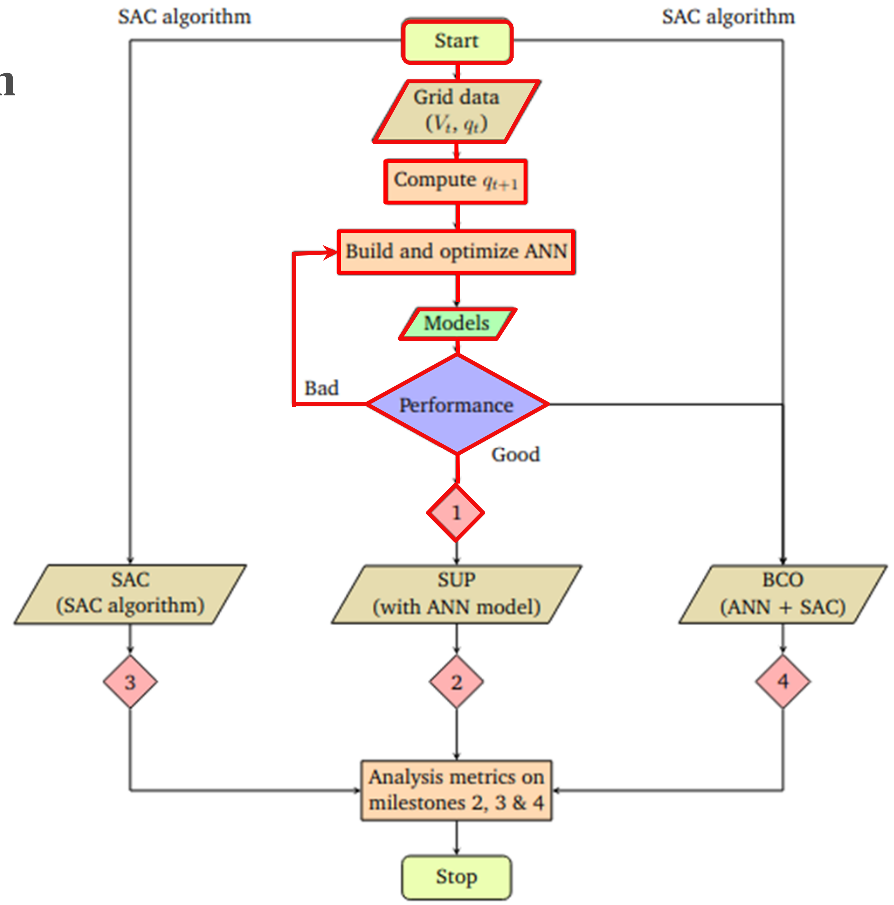
Process Flow Diagram



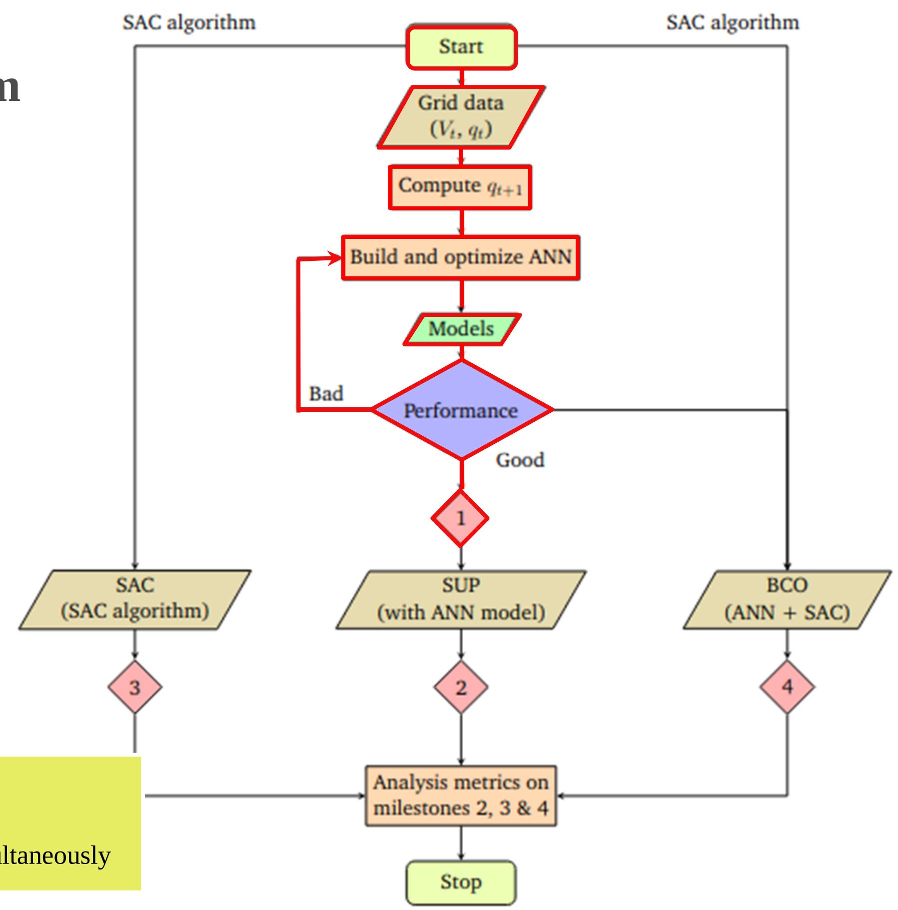
Process Flow Diagram



Process Flow Diagram



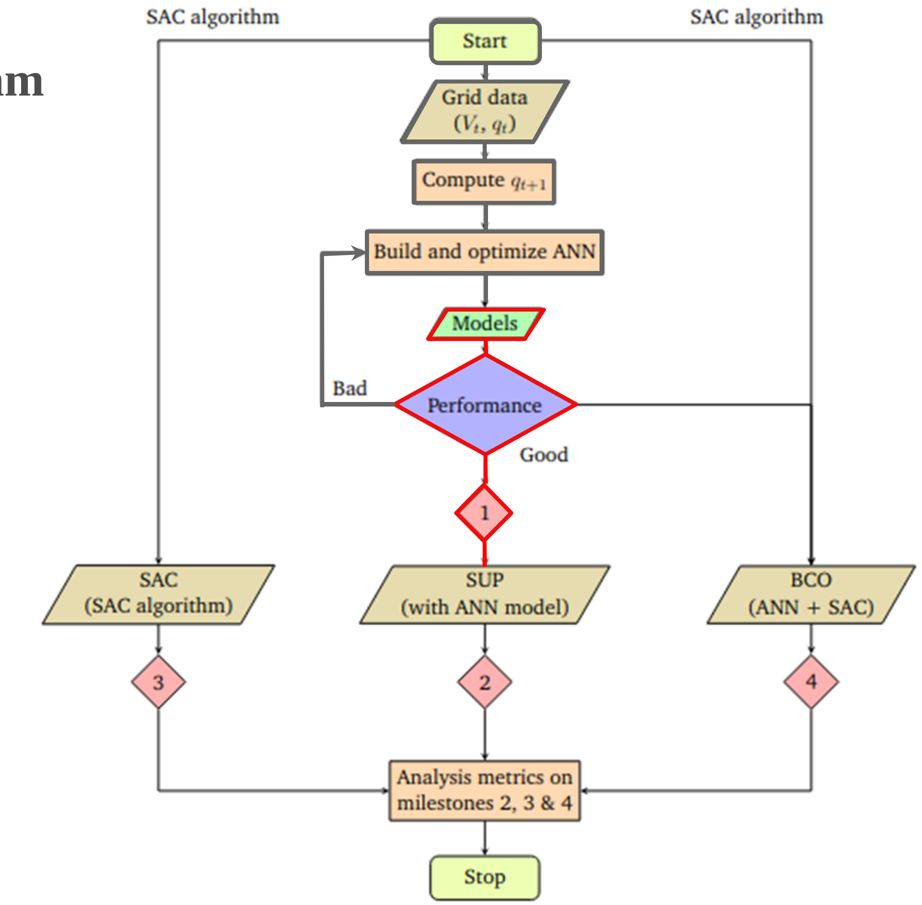
Process Flow Diagram



Cases:

- **Single Bus:** Only one bus is controlled
- **Two Buses:** Two buses are controlled simultaneously

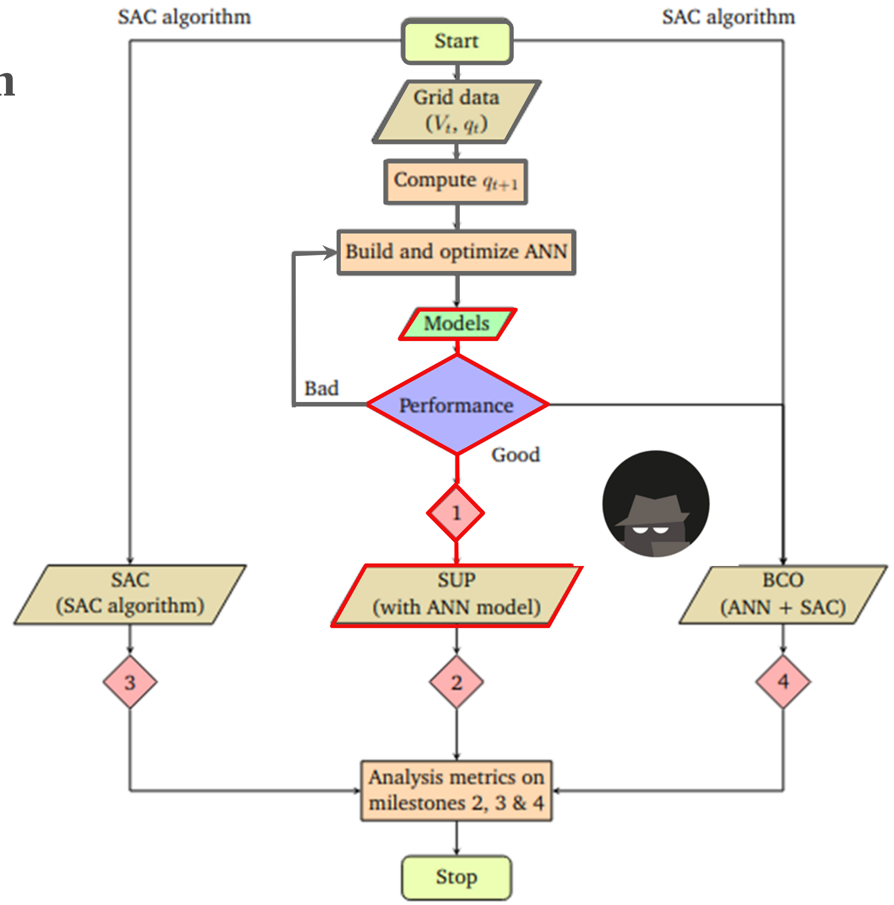
Process Flow Diagram



Experiment building

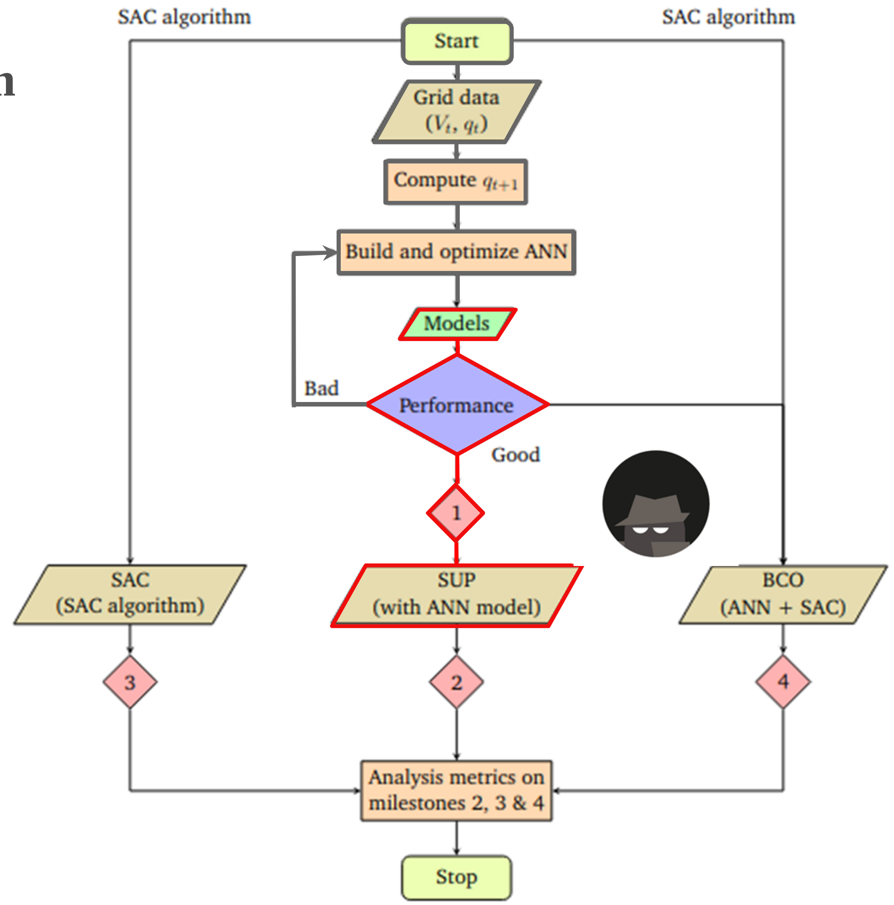
Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Process Flow Diagram



Process Flow Diagram

Only ANN

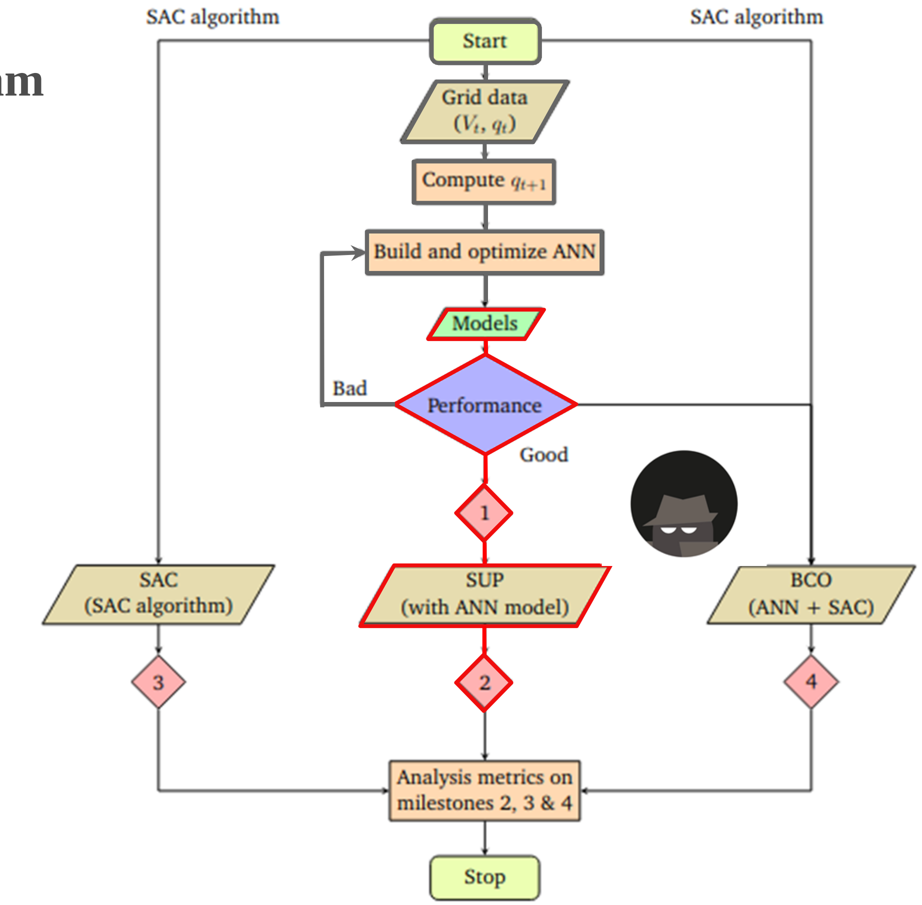


Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Process Flow Diagram

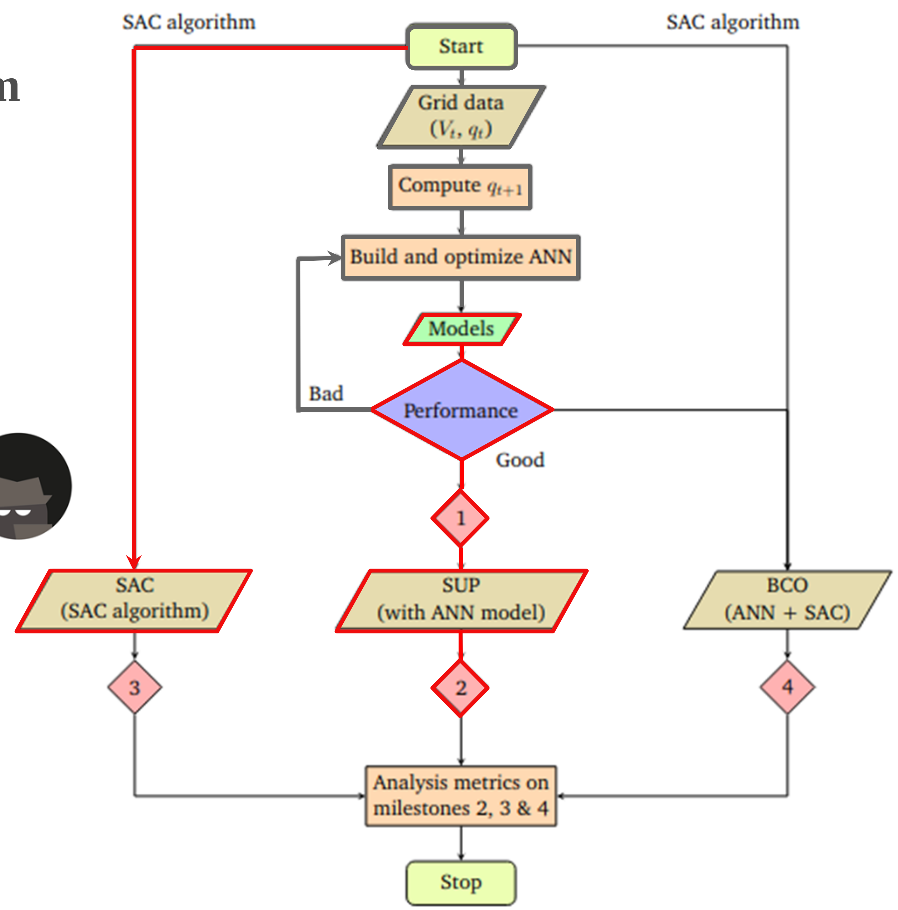
Only ANN

No further learning!



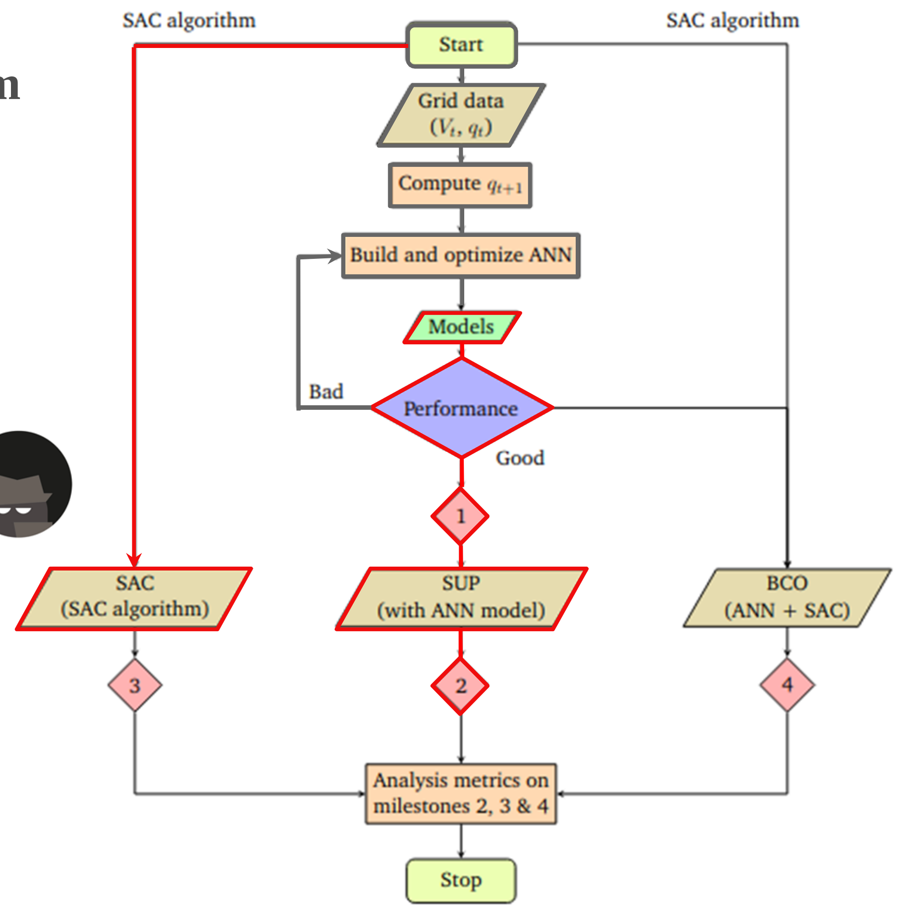
Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Process Flow Diagram



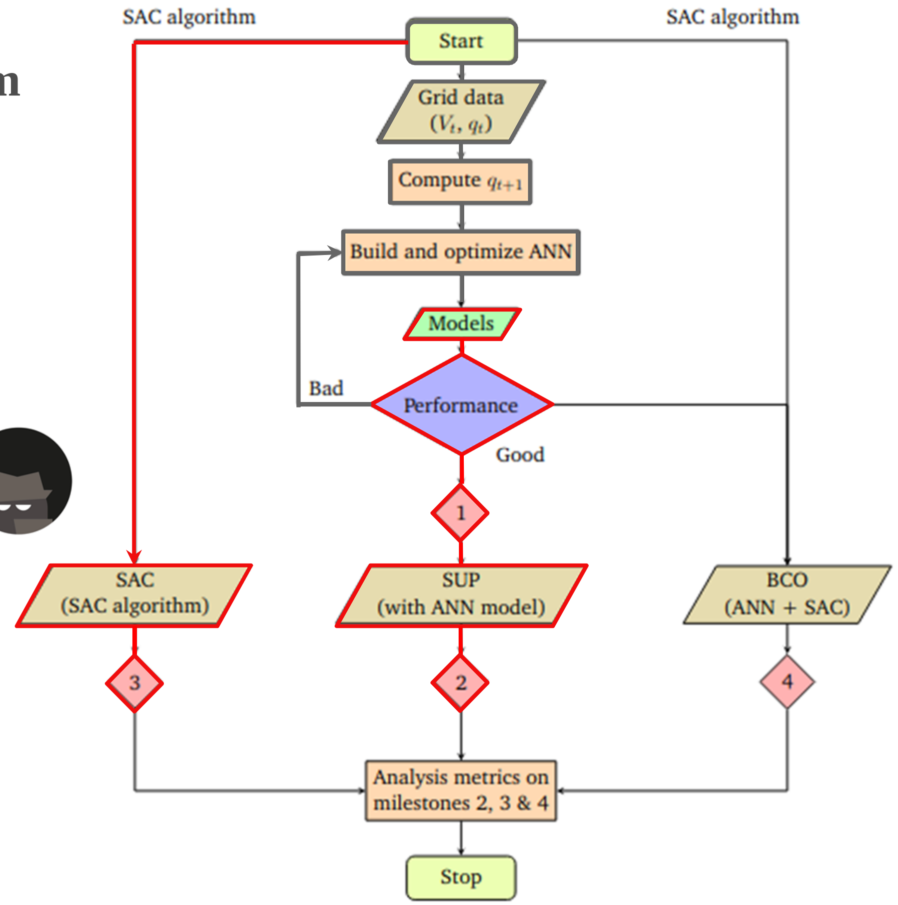
Process Flow Diagram

Only SAC



Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Process Flow Diagram

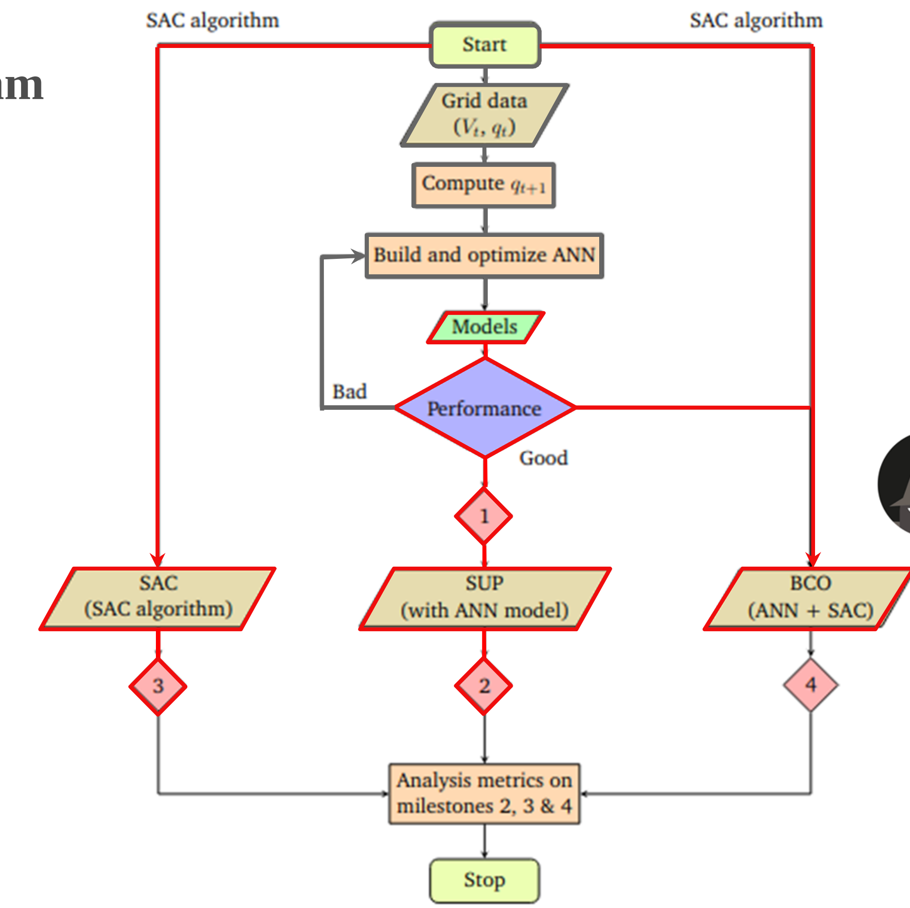


Only SAC

Off policy RL learning

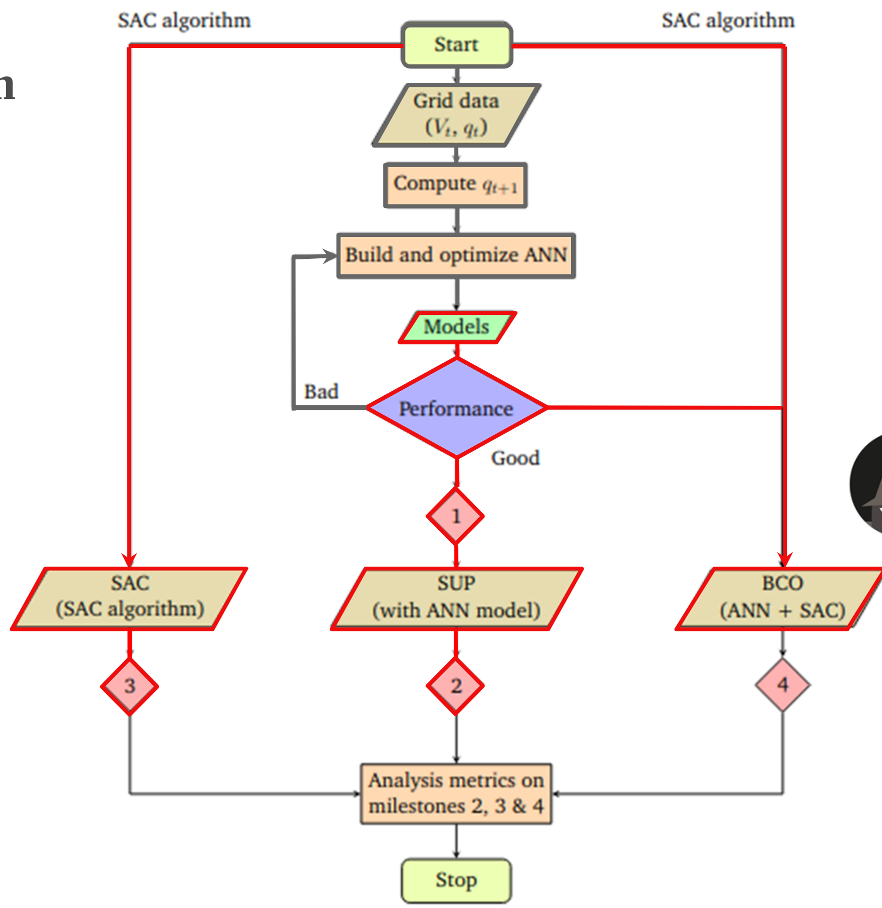
Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Process Flow Diagram



Process Flow Diagram

SAC + ANN = BCO

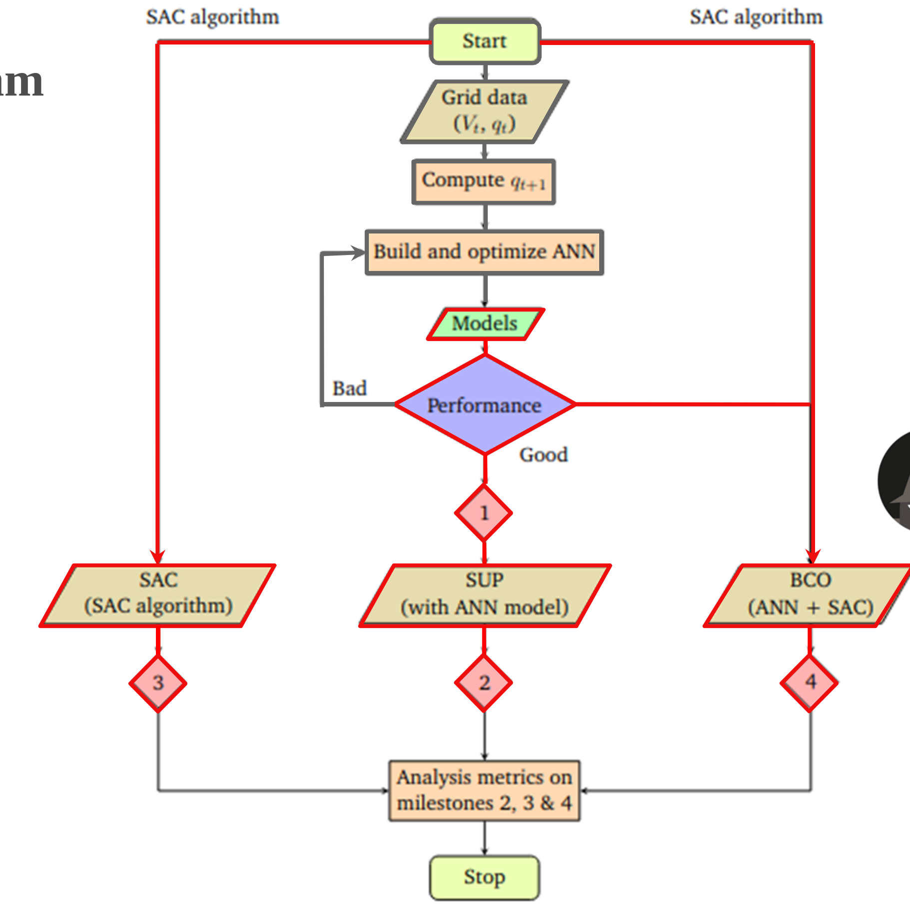


Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

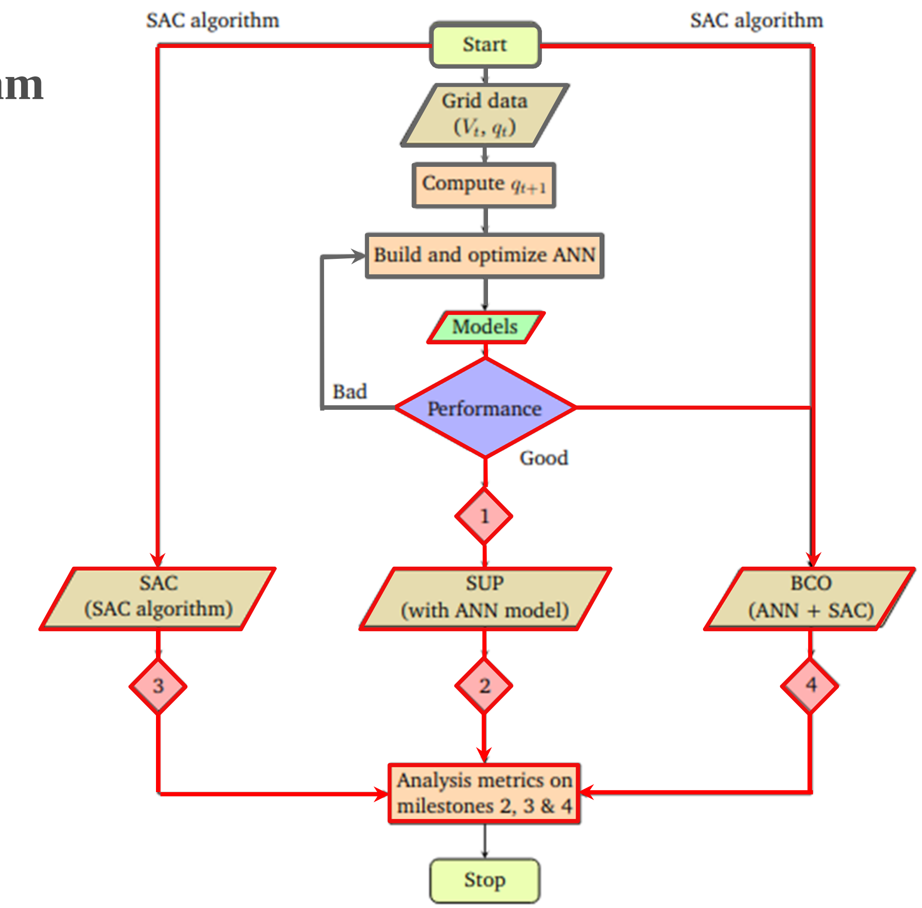
Process Flow Diagram

SAC + ANN = BCO

Offline RL learning



Process Flow Diagram



Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Process Flow Diagram

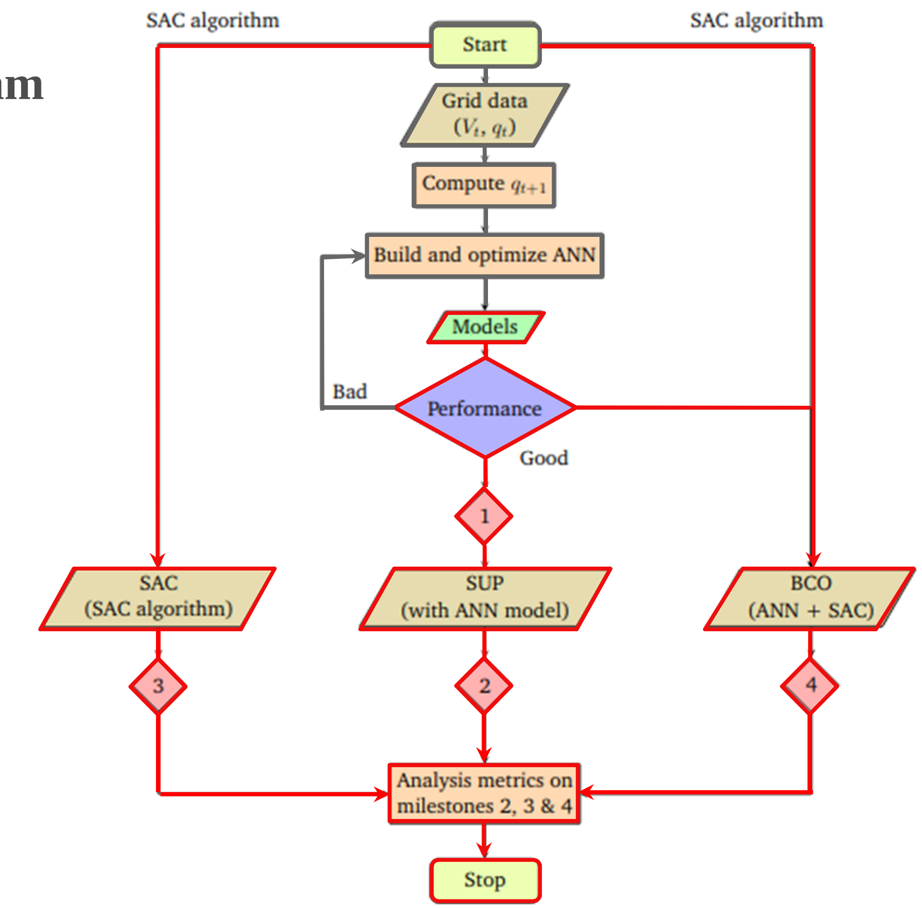


Table of Contents

Two Step Evaluation

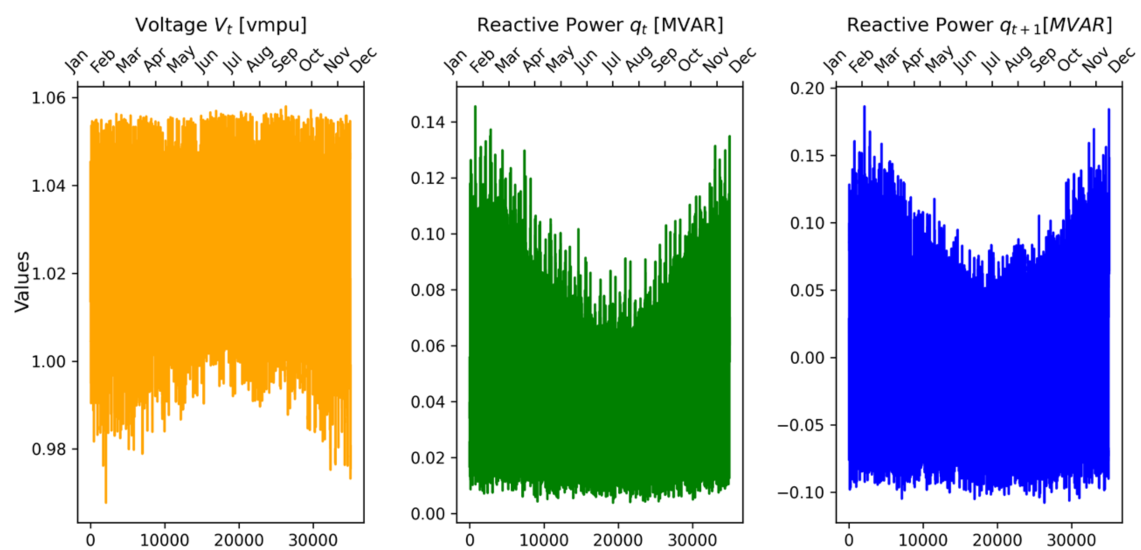
Neural Network
Experiments

Two Step Evaluation

Neural Network

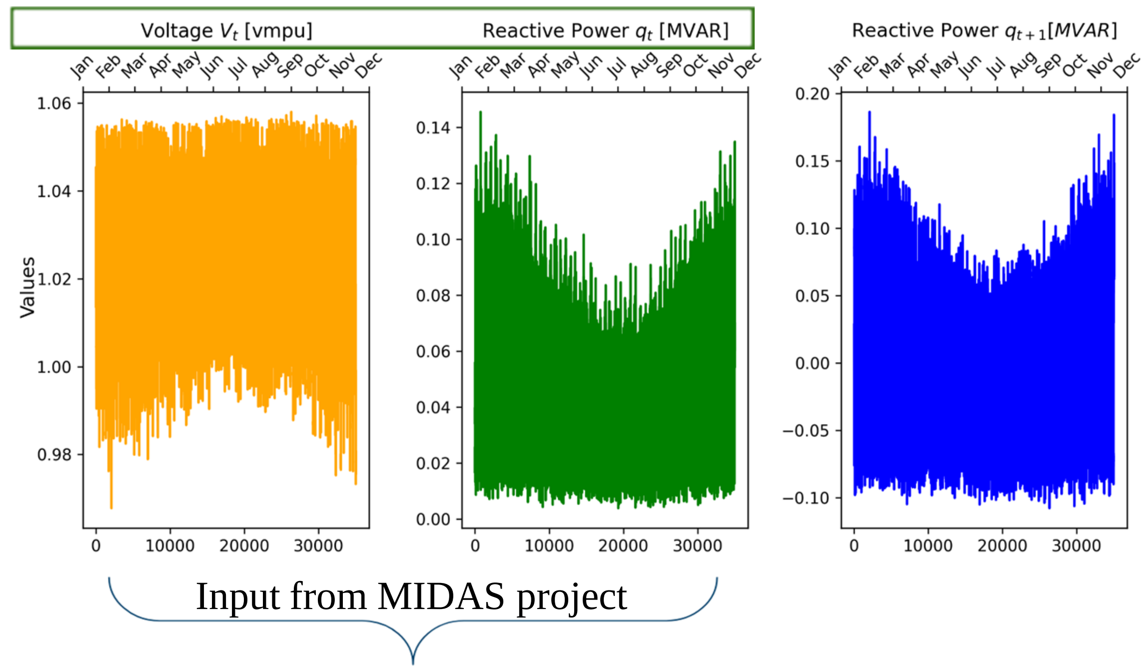
Experiments

Neural Network Optimization : Validation of equation used



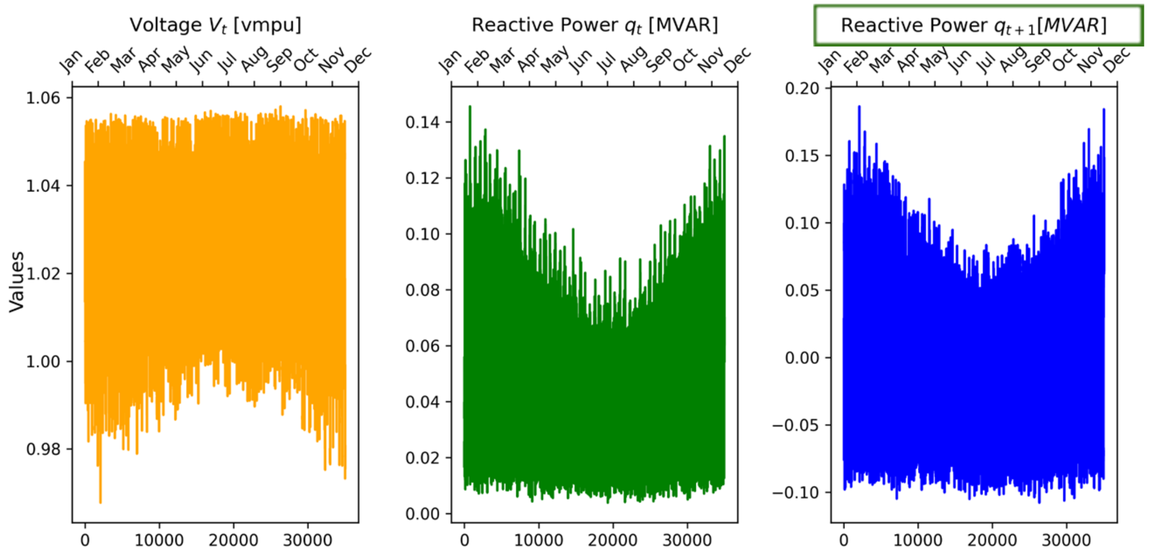
Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Neural Network Optimization : Validation of equation used



Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Neural Network Optimization : Validation of equation used

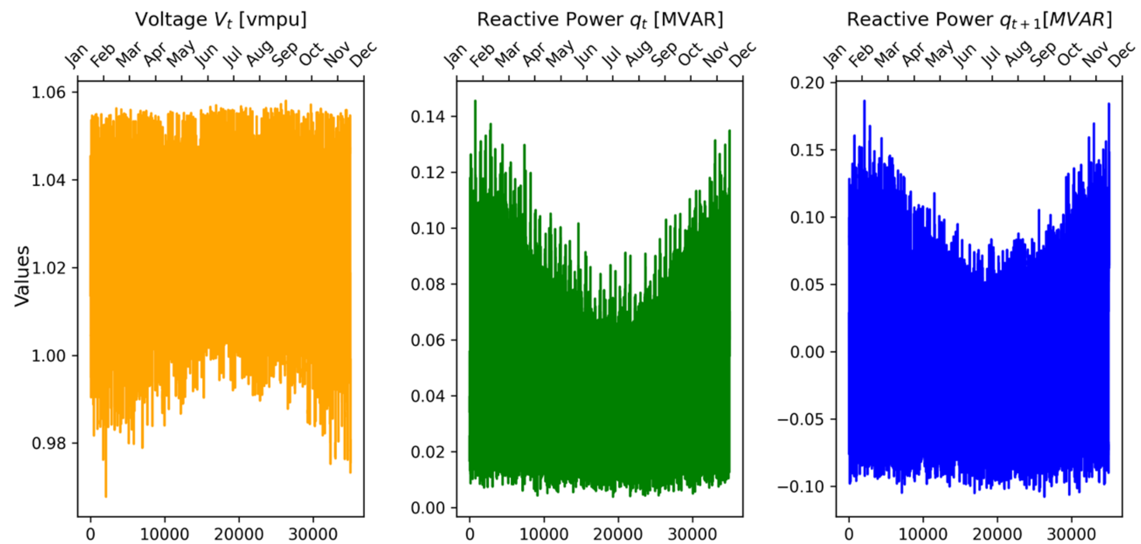


$$q_{t+1} = [q_t - D(V_t - 1)]^+$$

Calculated set point

Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Neural Network Optimization : Validation of equation used



Trend check

Supervised learning

ANN

Optimize

Analyze

14 Models for 14 Datasets

Two Step Evaluation

Neural Network
Experiments

Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Two Step Evaluation

Neural Network

Experiments

- Single Bus
- Two Buses

Experiments

Cases

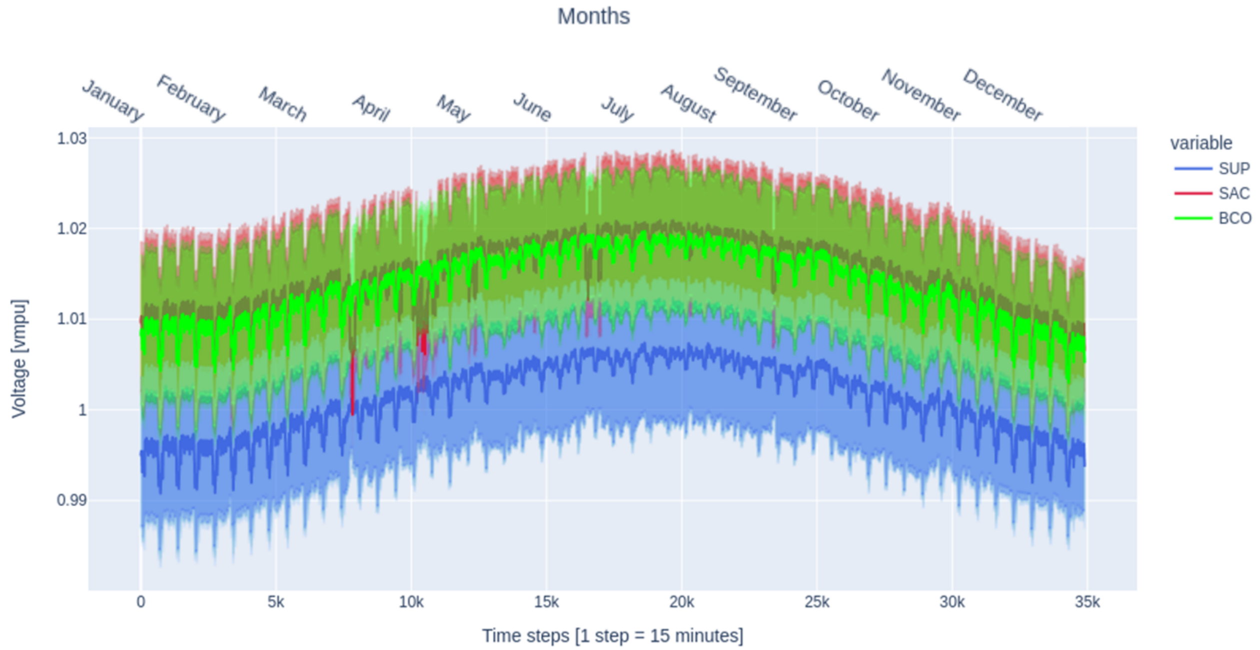
- Single Bus
- Two Buses

Experiments

Cases

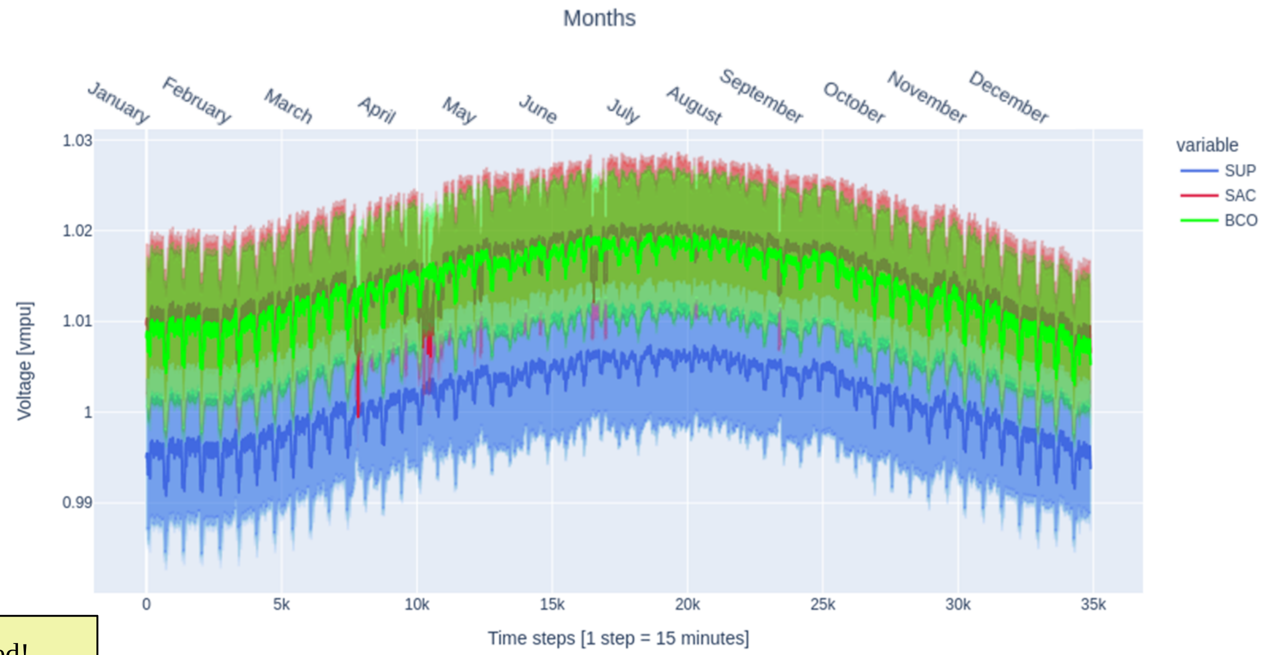
- Single Bus
- Two Buses

Single Bus - Voltage performance



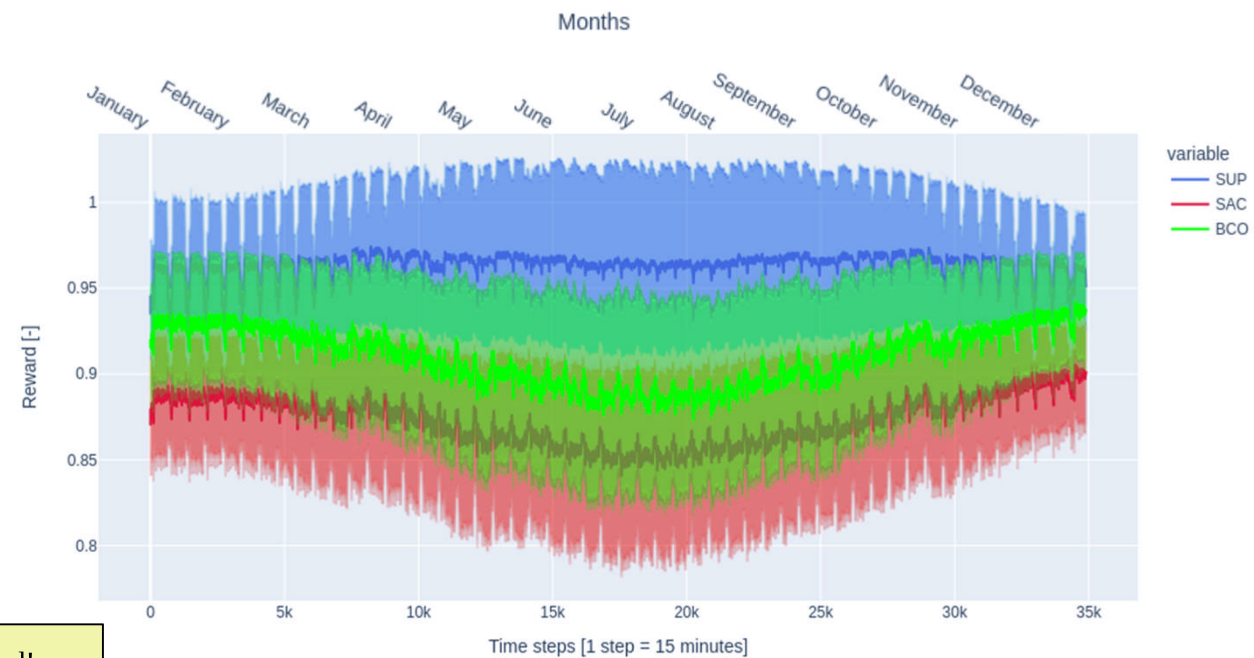
Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Single Bus - Voltage performance



Objective function is followed!

Single Bus - Reward performance



Objective function is followed!

BCO > SAC!

Single Bus Analysis

Sr. No.	Metrics	SUP	SAC	BCO
1	Voltage performance	$0.95 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$
2	Reward performance	$0.54 \leq R_{b,t} \leq 1.00$	$0.33 \leq R_{b,t} \leq 1.00$	$0.46 \leq R_{b,t} \leq 1.00$
3	Sample efficiency			
3.1	Model [Data points utilized]	5000	N/A	30000
3.2	SAC algorithm	N/A		
	Slope	N/A	-0.0020	-0.0024
	AUC	N/A	22.45	22.75

Single Bus Analysis

OFFIS

Sr. No.	Metrics	SUP	SAC	BCO
1	Voltage performance	$0.95 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$
2	Reward performance	$0.54 \leq R_{b,t} \leq 1.00$	$0.33 \leq R_{b,t} \leq 1.00$	$0.46 \leq R_{b,t} \leq 1.00$
3	Sample efficiency			
3.1	Model [Data points utilized]	5000	N/A	30000
3.2	SAC algorithm	N/A		
	Slope	N/A	-0.0020	-0.0024
	AUC	N/A	22.45	22.75

Voltage[vmpu] → $0.85 \leq V_{b,t} < 1.15$ ∀ b in Buses, time
Reward → $0.90 \leq R_{b,t} \leq 1.0$ ∀ b in Buses, time

Single Bus Analysis

Sr. No.	Metrics	SUP	99.9%	SAC	BCO
1	Voltage performance	$0.95 \leq V_{b,t} \leq 1.05$		$0.0 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$
2	Reward performance	$0.54 \leq R_{b,t} \leq 1.00$		$0.33 \leq R_{b,t} \leq 1.00$	$0.46 \leq R_{b,t} \leq 1.00$
3	Sample efficiency				
3.1	Model [Data points utilized]	5000	97.0%	N/A	30000
3.2	SAC algorithm	N/A			
	Slope	N/A		-0.0020	-0.0024
	AUC	N/A		22.45	22.75

Voltage[vmpu] → $0.85 \leq V_{b,t} < 1.15$ ∀ b in Buses, time
Reward → $0.90 \leq R_{b,t} \leq 1.0$ ∀ b in Buses, time

Single Bus Analysis

Sr. No.	Metrics	SUP		SAC		BCO
			99.9%		99.9%	
1	Voltage performance	$0.95 \leq V_{b,t} \leq 1.05$		$0.0 \leq V_{b,t} \leq 1.05$		$0.0 \leq V_{b,t} \leq 1.05$
2	Reward performance	$0.54 \leq R_{b,t} \leq 1.00$		$0.33 \leq R_{b,t} \leq 1.00$		$0.46 \leq R_{b,t} \leq 1.00$
3	Sample efficiency					
			97.0%		47.0%	
3.1	Model [Data points utilized]	5000		N/A		30000
3.2	SAC algorithm	N/A				
	Slope	N/A		-0.0020		-0.0024
	AUC	N/A		22.45		22.75

Voltage[vmpu] → $0.85 \leq V_{b,t} < 1.15$ ∀ b in Buses, time
Reward → $0.90 \leq R_{b,t} \leq 1.0$ ∀ b in Buses, time

Single Bus Analysis

Sr. No.	Metrics	SUP		SAC		BCO	
			99.9%		99.9%		99.9%
1	Voltage performance	$0.95 \leq V_{b,t} \leq 1.05$		$0.0 \leq V_{b,t} \leq 1.05$		$0.0 \leq V_{b,t} \leq 1.05$	
2	Reward performance	$0.54 \leq R_{b,t} \leq 1.00$		$0.33 \leq R_{b,t} \leq 1.00$		$0.46 \leq R_{b,t} \leq 1.00$	
3	Sample efficiency		97.0%		47.0%		65.6%
3.1	Model [Data points utilized]	5000		N/A		30000	
3.2	SAC algorithm	N/A					
	Slope	N/A		-0.0020		-0.0024	
	AUC	N/A		22.45		22.75	

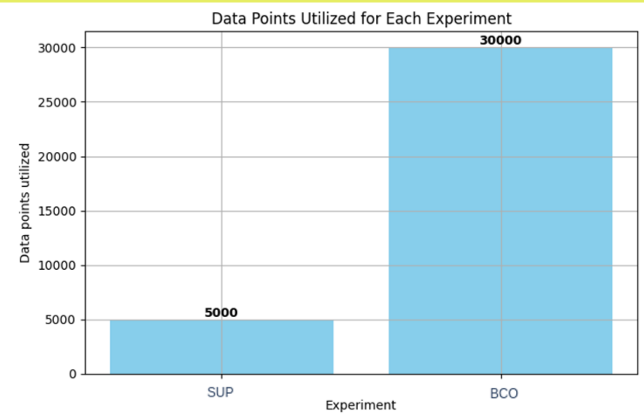
Voltage[vmpu] → $0.85 \leq V_{b,t} < 1.15$ ∀ b in Buses, time
Reward → $0.90 \leq R_{b,t} \leq 1.0$ ∀ b in Buses, time

Single Bus Analysis

Sr. No.	Metrics	SUP	SAC	BCO
1	Voltage performance	$0.95 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$
2	Reward performance	$0.54 \leq R_{b,t} \leq 1.00$	$0.33 \leq R_{b,t} \leq 1.00$	$0.46 \leq R_{b,t} \leq 1.00$
3	Sample efficiency			
3.1	Model [Data points utilized]	5000	N/A	30000
3.2	SAC algorithm	N/A		
	Slope	N/A	-0.0020	-0.0024
	AUC	N/A	22.45	22.75

Single Bus Analysis

Sr. No.	Metrics	SUP	SAC	BCO
1	Voltage performance	$0.95 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$
2	Reward performance	$0.54 \leq R_{b,t} \leq 1.00$	$0.33 \leq R_{b,t} \leq 1.00$	$0.46 \leq R_{b,t} \leq 1.00$
3	Sample efficiency			
3.1	Model [Data points utilized]	5000	N/A	30000
3.2	SAC algorithm	N/A		
	Slope	N/A		
	AUC	N/A		



Single Bus Analysis

Sr. No.	Metrics	SUP	SAC	BCO
1	Voltage performance	$0.95 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$
2	Reward performance	$0.54 \leq R_{b,t} \leq 1.00$	$0.33 \leq R_{b,t} \leq 1.00$	$0.46 \leq R_{b,t} \leq 1.00$
3	Sample efficiency			
3.1	Model [Data points utilized]	5000	N/A	30000
3.2	SAC algorithm	N/A		

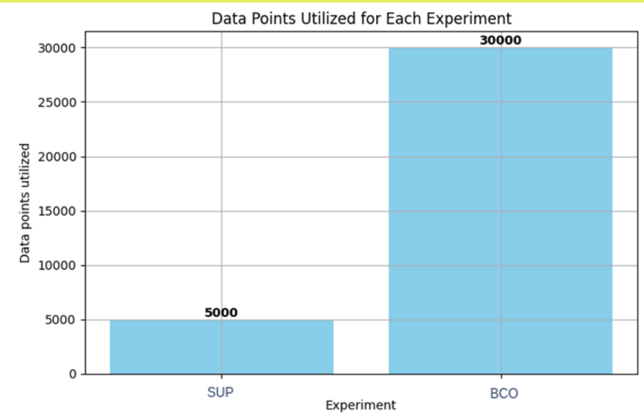
SAC's ability to enhance exploration

↓

A comparatively complex task

↓

Richer dataset to capture underlying patterns effectively



Single Bus Analysis

Sr. No.	Metrics	SUP	SAC	BCO
1	Voltage performance	$0.95 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$
2	Reward performance	$0.54 \leq R_{b,t} \leq 1.00$	$0.33 \leq R_{b,t} \leq 1.00$	$0.46 \leq R_{b,t} \leq 1.00$
3	Sample efficiency			
3.1	Model [Data points utilized]	5000	N/A	30000
3.2	SAC algorithm	N/A		
	Slope	N/A	-0.0020	-0.0024
	AUC	N/A	22.45	22.75

Single Bus Analysis

Sr. No.	Metrics	SUP	SAC	BCO
1	Voltage performance	$0.95 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$
2	Reward performance	$0.54 \leq R_{b,t} \leq 1.00$	$0.33 \leq R_{b,t} \leq 1.00$	$0.46 \leq R_{b,t} \leq 1.00$
3	Sample efficiency			
3.1	Model [Data points utilized]	5000	N/A	30000
3.2	SAC algorithm	N/A		
	Slope	N/A	-0.0020	-0.0024
	AUC	N/A	22.45	22.75

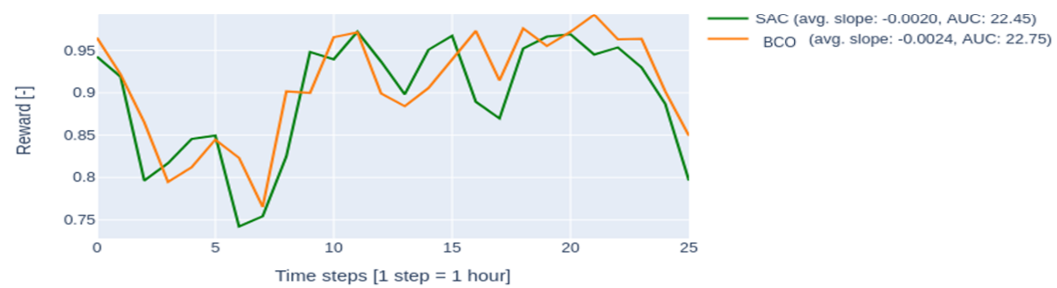
$$\eta_s = \frac{dR}{dt}$$

Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Single Bus Analysis

Sr. No.	Metrics	SUP	SAC	BCO
1	Voltage performance	$0.95 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$
2	Reward performance	$0.54 \leq R_{b,t} \leq 1.00$	$0.33 \leq R_{b,t} \leq 1.00$	$0.46 \leq R_{b,t} \leq 1.00$
3	Sample efficiency			
3.1	Model [Data points utilized]	5000	N/A	30000
3.2	SAC algorithm	N/A		
	Slope	N/A	-0.0020	-0.0024
	AUC	N/A	22.45	22.75

$$\eta_s = \frac{dR}{dt}$$

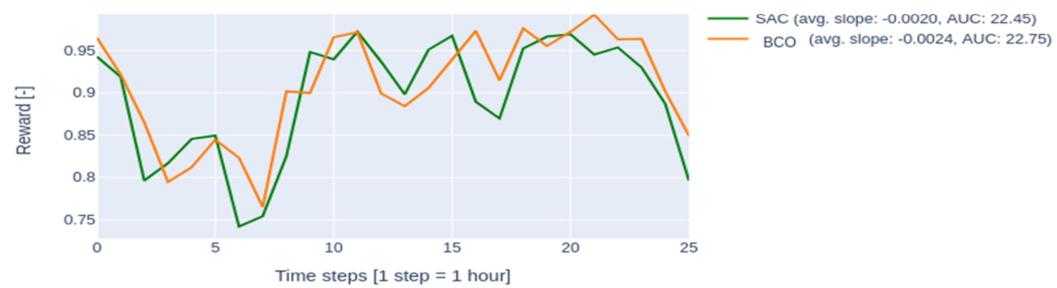


Single Bus Analysis

Sr. No.	Metrics	SUP	SAC	BCO
1	Voltage performance	$0.95 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$	$0.0 \leq V_{b,t} \leq 1.05$
2	Reward performance	$0.54 \leq R_{b,t} \leq 1.00$	$0.33 \leq R_{b,t} \leq 1.00$	$0.46 \leq R_{b,t} \leq 1.00$
3	Sample efficiency			
3.1	Model [Data points utilized]	5000	N/A	30000
3.2	SAC algorithm	N/A		
	Slope	N/A	-0.0020	-0.0024
	AUC	N/A	22.45	22.75

+20 %
+1.34 %

$$\eta_s = \frac{dR}{dt}$$



Experiments

Cases

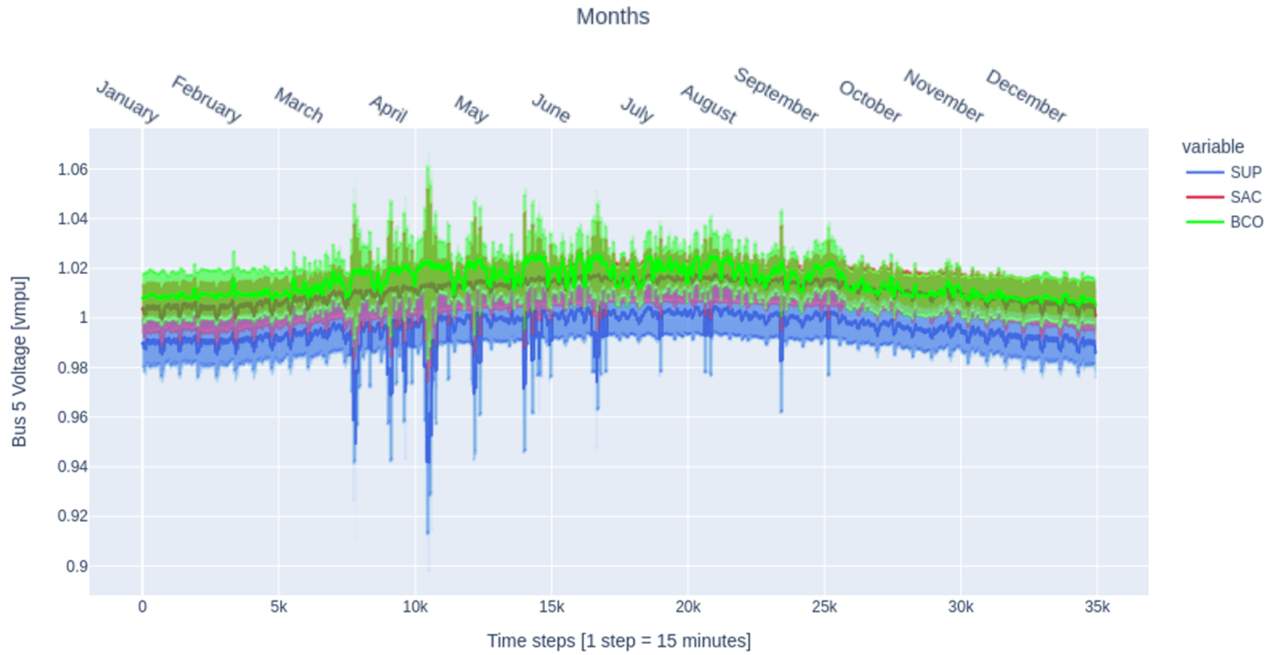
- Single Bus
- Two Buses

Experiments

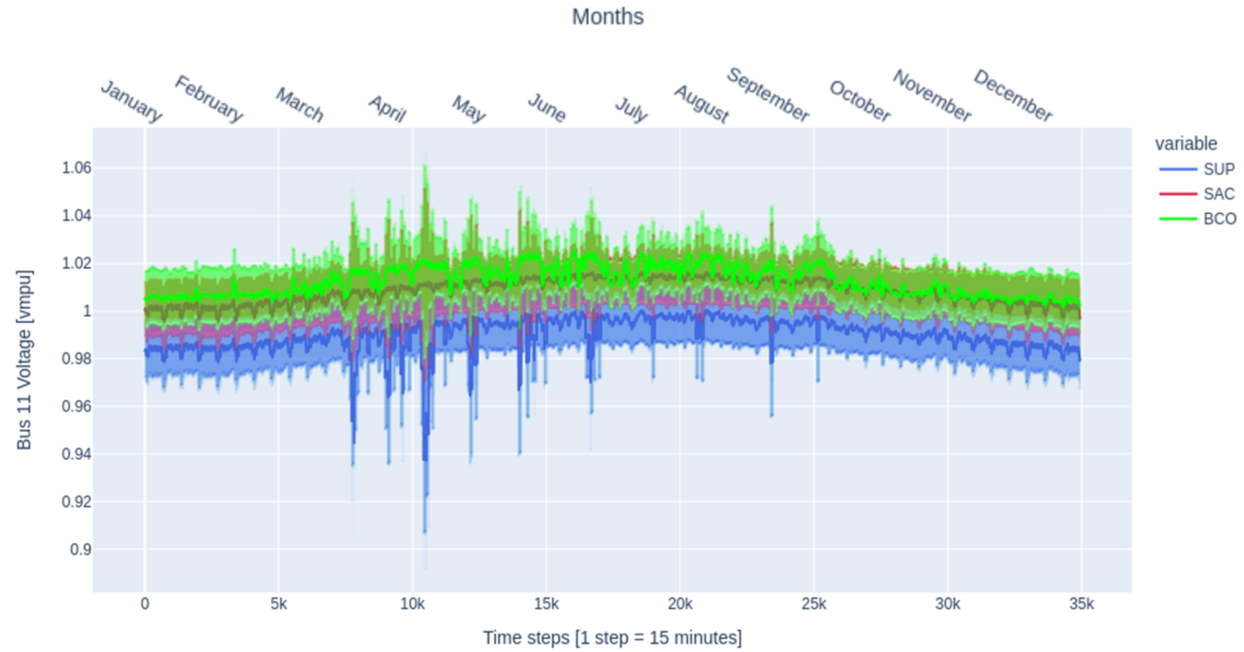
Cases

- Single Bus
- Two Buses

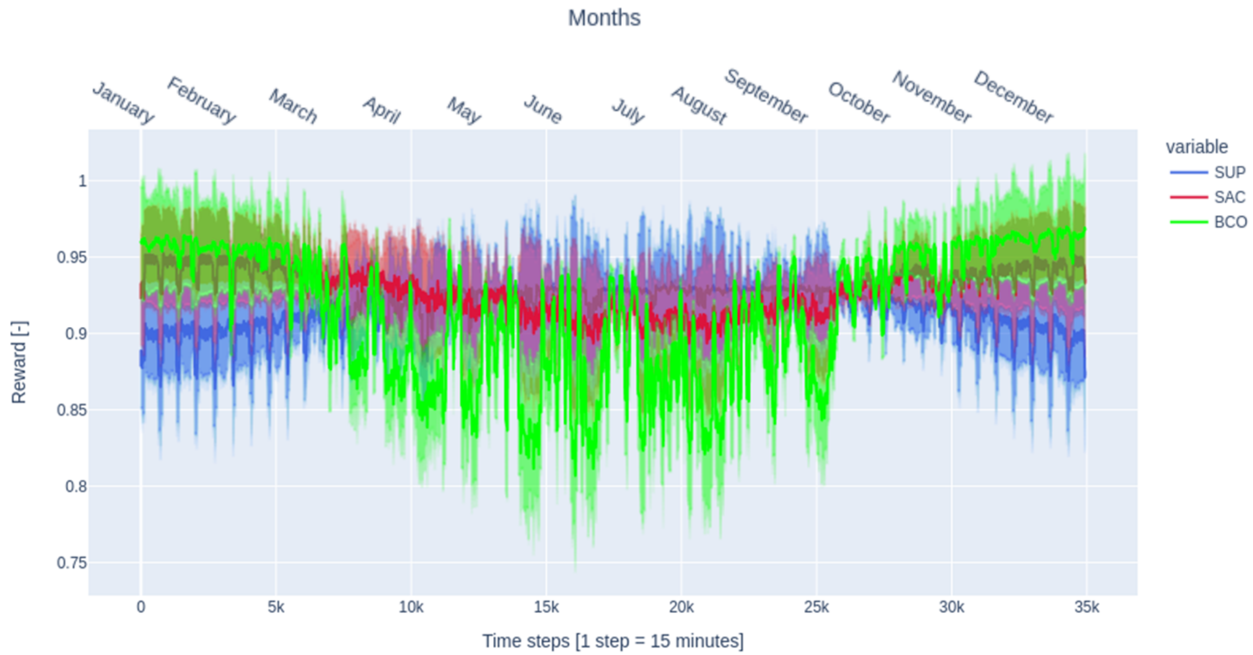
Two Buses - Bus 5 Voltage performance



Two Buses - Bus 11 Voltage performance



Two Buses - Reward performance



Two Buses - Analysis

Criteria defined for controlling two buses:

Voltage[vmpu] → $0.85 \leq V_{b,t} < 1.15$ ∀ b in Buses, time
Reward → $0.90 \leq R_{b,t} \leq 1.0$ ∀ b in Buses, time

Two Buses - Analysis

Criteria defined for controlling two buses:

Voltage[vmpu] → $0.85 \leq V_{b,t} < 1.15$ ∀ b in Buses, time
Reward → $0.90 \leq R_{b,t} \leq 1.0$ ∀ b in Buses, time

Sr. No.	Metrics	SUP	SAC	BCO
1	Voltage performance	$0.0 \leq V_{b,t} \leq 1.05$	$0.96 \leq V_{b,t} \leq 1.03$	$0.98 \leq V_{b,t} \leq 1.06$
2	Reward performance	$0.49 \leq R_{b,t} \leq 0.99$	$0.78 \leq R_{b,t} \leq 1.00$	$0.53 \leq R_{b,t} \leq 1.00$
3	Occurrences within limits			
	Voltage [vmpu] $(0.85 \leq V_{b,t} < 1.15)$	99.83 %	100.00 %	100.00 %
	Reward $(0.9 \leq R_{b,t} < 1)$	70.80 %	70.59 %	81.56 %

Two Buses - Analysis

Criteria defined for controlling two buses:

Voltage[vmpu] → $0.85 \leq V_{b,t} < 1.15$ ∀ b in Buses, time
Reward → $0.90 \leq R_{b,t} \leq 1.0$ ∀ b in Buses, time

Sr. No.	Metrics	SUP	SAC	BCO
1	Voltage performance	$0.0 \leq V_{b,t} \leq 1.05$	$0.96 \leq V_{b,t} \leq 1.03$	$0.98 \leq V_{b,t} \leq 1.06$
2	Reward performance	$0.49 \leq R_{b,t} \leq 0.99$	$0.78 \leq R_{b,t} \leq 1.00$	$0.53 \leq R_{b,t} \leq 1.00$
3	Occurrences within limits			
	Voltage [vmpu] $(0.85 \leq V_{b,t} < 1.15)$	99.83 %	100.00 %	100.00 %
	Reward $(0.9 \leq R_{b,t} < 1)$	70.80 %	70.59 %	81.56 %

+10.97 %

Table of Contents

Conclusion

	EVALUATION CRITERIA	SUP	SAC	BCO
SINGLE BUS	Voltage Stability	🙂	🙂	🙂
	Reward Collection	🙂	🙁	😐
	Sample Efficiency: Model	🙂		🙁
	Sample Efficiency: SAC algorithm		🙁	🙂
TWO BUSES	Voltage Stability	🙁	🙂	🙂
	Reward Collection	😐	🙁	🙂

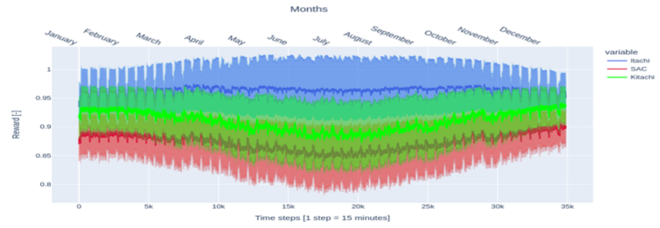
Legend	
Best	🙂
OK	😐
Worst	🙁

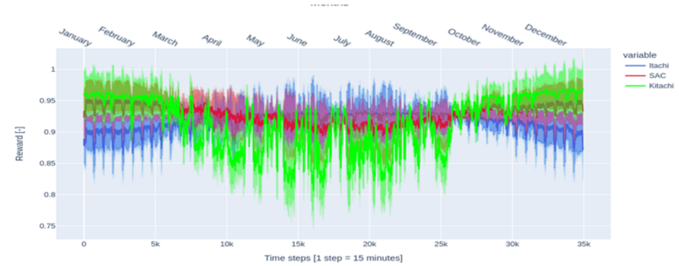
Thank you!

Reactive Power Control
via
ANN agent

Influence of ANN on
experiment performance

Influence of BCO on
sample efficiency





Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Constraints and Limitations

1. **PalaestrAI** framework is utilized for implementing the reactive power controller.

1. The choice of the **SAC** algorithm for policy formulation and comparison in this study is motivated by its compatibility with continuous action spaces. SAC is selected for its efficient learning capabilities, leveraging entropy maximization and stability.

1. Research by Haarnoja et al. demonstrates that SAC outperforms other state-of-the-art model-free deep RL methods like the off-policy Deep Deterministic Policy Gradient (DDPG) algorithm and the on-policy Proximal Policy Optimization (PPO) algorithm [17]. This suggests that using stochastic, entropy-maximizing RL algorithms can offer improved robustness and stability.

1. **BCO** is selected because of the relative simplicity of the approach and availability of high-quality data from the MIDAS project, ensuring reliability. Although Advantage Weighted Actor-Critic (AWAC)offers a method to incorporate prior data and reduce learning time, it is most advantageous when the prior data is suboptimal [28]. Since this thesis relies on data from a reliable and optimal source, behavioral cloning is the preferred method.

1. Simulation time for each experiment scenario is set at **one year**, with each year requiring one hour for simulation alone. Both training and testing will demand additional time.

Motivation: *Why Voltage Control?*

1. Voltage levels impact the performance and longevity of customer and power system equipment. Operating outside the designated voltage range can lead to inefficiencies and damage. Low voltages can impair the performance of devices like light bulbs and induction motors, while high voltages can cause equipment damage.

1. Reactive power utilization imposes demands on transmission and generation resources. Minimizing reactive-power flows is necessary to optimize the transfer of real power across congested transmission interfaces. Excessive reactive power production can also restrict a generator’s capacity to supply real power.

1. The movement of reactive power within the transmission system results in real-power losses. Addressing these losses requires additional capacity and energy, which adds to operational costs and reduces overall system efficiency.

Motivation: *Reactive power management is COMPLEX!*

1. Real power can travel long distances efficiently, while reactive power needs to be dispersed across the power system.

1. The system's reactive power requirements evolve over time due to variations in:
   a. Generation
   b. Transmission configurations &
   c. Load levels

1. For example:
   a. During periods of low load, excess reactive power generated by the system must be absorbed
   b. Under heavy load, additional reactive power must be supplied
   c. Reactive losses surpass real losses at both low and high line loading, substantially reducing the transmitted real power if uncompensated

Motivation: *Why reactive power through operator?*

1. Reactive power management rely on centralized approaches, primarily overseen by the system operator.

1. This centralized control is critical due to its requirement for a comprehensive understanding of system needs and the ability to strategically deploy resources.

1. While suppliers, such as generators with reactive-power capabilities, lack autonomy in determining voltage-control needs, the system operator possesses the necessary information to make informed decisions.

1. Moreover, customer choices in load patterns and generation do not provide adequate insight into reactive-power requirements.

1. This highlights the necessity of the system operator's role in resource deployment.

1. *The limited transportability of reactive power compared to real power highlights the* ***potential benefits of distributed generators*** *providing reactive power control at strategic locations.*

Managing Reactive Power

1. Similar to real power, ensuring the balance of reactive power throughout the system is essential.

1. A mismatch in reactive power, unlike real power, can lead to voltage collapse rather than loss of synchronicity.

1. Reactive losses on a transmission line can be positive or negative, depending on the dominance of inductive or capacitive reactance, unlike real power losses, which are consistently positive as they represent physical heat dissipated into the environment.

1. However, operational considerations for balancing reactive power differ from those for real power.

1. Rather than instructing generators to produce a specific amount of reactive power, they are directed to ***maintain*** *a certain* ***voltage magnitude*** *at their buses*, adjusted through the generator field current.

1. This approach simplifies power flow analysis, as specifying voltage magnitude effectively ensures balanced reactive power without explicitly knowing the total required amount.

Why PV *and not conventional reactive power compensation techniques*

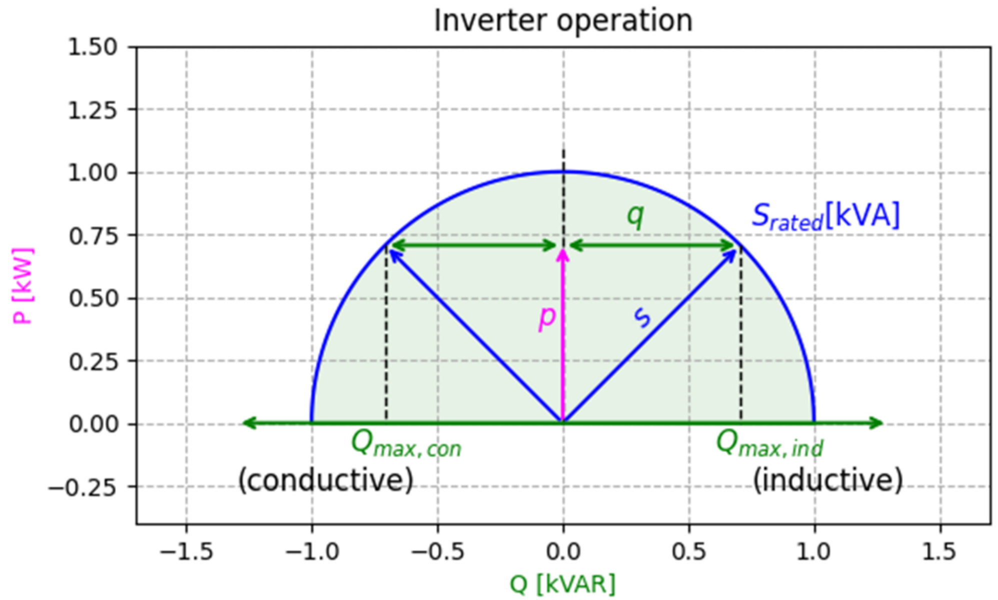
1. Inverter has full control over reactive power, similar to conventional devices like STATCOMs.

1. Cost of inverter has reduced at higher rate than traditional var compensation devices.

1. Distributed generation resources, dispersed throughout power system, can provide reactive power in a distributed manner as well.
    a. Reactive power compensation should be done locally, near the reactive loads to avoid transmission losses (Enhanced Efficiency).
    b. Diverse combinations of reactive injections and optimizing system operation (Flexibility).
    c. No need for extra installation for reactive power management (cost).
    d. (Scalability) Higher PV penetration possible with seamless system upgrades (no more a drawback!).
    e. Reliability: Dependency on extensive capacitor banks increases grid vulnerability to equipment failures and cyber attacks. In contrast, distributed control systems with limited communication between smaller components offer enhanced resilience against cyber threats.

Limitations of PV inverter capabilities

$$|q_g| \leq \sqrt{S^2 - (p_g)^2} \equiv q_{\max}$$



$$s \sim 1.1 p_{g,max}$$

Optimal for achieving sufficient reductions in distribution losses

$$|q_g| \leq 0.45 p_{g,max}$$

Limitations of PV inverter vs Wind inverter capabilities

PV Inverters:

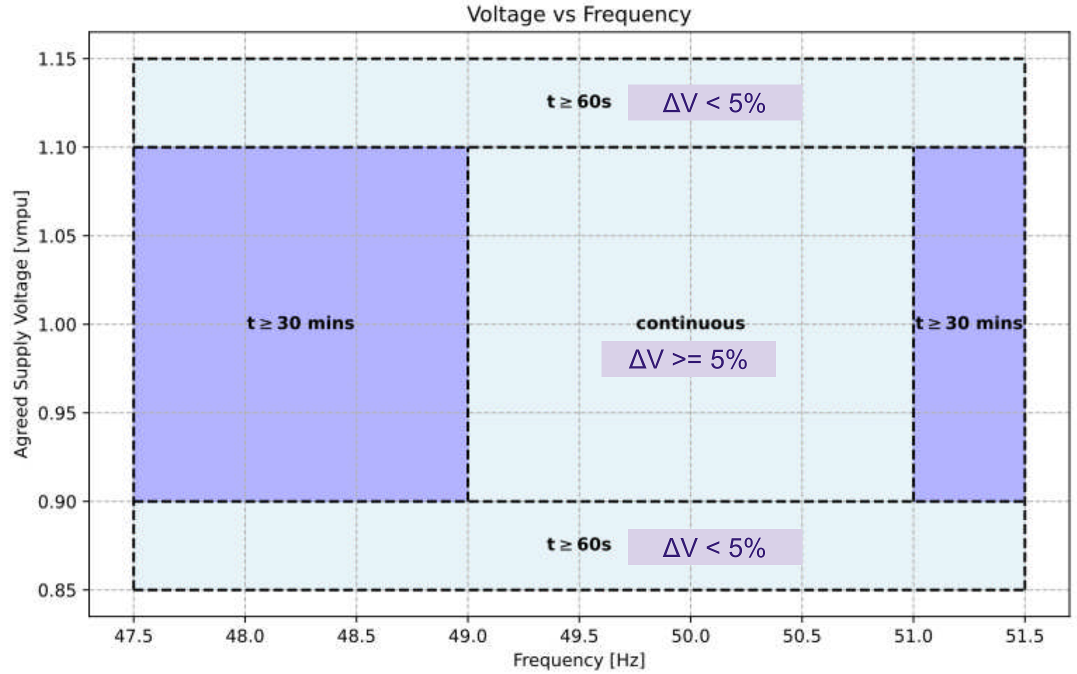
1. Solar Irradiance Variability: PV inverters must handle rapid changes in solar irradiance due to passing clouds, which can cause fluctuations in power output.
2. Maximum Power Point Tracking (MPPT): PV inverters need efficient MPPT algorithms to optimize the energy harvest from solar panels under varying conditions.

Wind Inverters:

1. Wind Speed Variability: Wind inverters must handle the variability in wind speed, which can lead to fluctuations in power generation.
2. Turbine Dynamics: Wind inverters may need to work with turbine control systems (e.g., for blade pitch adjustment) to optimize performance and protect the system under extreme conditions.

Grid Code

VDE-AR-N 4110



Generation system must be connected to the grid for at least 60s.
Each set-point provided by the grid operator must be attainable within 4 minutes.

Association for Electrical, Electronic and Information Technologies
(Verband der Elektrotechnik, Elektronik und Informationstechnik)

Operational status of bus (Grid Code DIN 50160 for MV):

Sr. No.	Grid constraints for medium voltage grid	Limits
1	Bus voltage gradients ΔV must be within $\leq$	0.1 vmpu/min
2	Loads need to sustain voltage fluctuations ΔV of	0.02 vmpu/min
3	Generators must endure voltage ΔV changes of	0.05 vmpu/min
4	Line load must be $\leq$	100%

Literature Review: ***Bus***

Bus in power system analysis, refers to a reference point representing an electrically distinct node, where different components of the system converge.

It is equivalent to a single point in the circuit and marks the location of either a power-generating generator or a power-consuming load.

Literature Review: ***Reinforcement Learning***

Markov Decision Process provides a mathematical framework to characterize an ideal environment in RL, enabling the formulation of theoretical insights into the problem.

MDP formulates the challenge of acquiring knowledge through interactions to accomplish an objective.

The complete MDP can be represented by a 6-tuple $M = (S, A, T, d_0, r, \gamma)$,

S	- state space,
A	- action space,
$T(s_{t+1} \| s_t, a_t)$	- transition distribution,
$d_0(s_0)$	- initial state distribution,
$r(s_t, a_t)$	- reward function,
$\gamma \in (0, 1]$	- discount factor.

This is applied on the future rewards to factor in its importance in current timeline.

Literature Review: *Reinforcement Learning*

In the context of a MDP, the goal is to determine a policy $\pi(a_t|s_t)$, representing the likelihood of taking action at given the current state s_t.

Whereas in RL, the focus shifts to identifying an optimal policy $\pi(a|s)^*$ that maximizes the expected return across all trajectories generated by the policy.

Literature Review: *Reinforcement Learning*

Let Y = (X_t)t∈N be a family of random variables, where $X_t \in S$. Y is a Markov chain if the following condition holds:

$$P(X_{t+1} = s_{jt+1} | X_t = s_{jt}, X_{t-1} = s_{jt-1}, \ldots, X_0 = s_{j0}) = P(X_{t+1} = s_{jt+1} | X_t = s_{jt}).$$

This condition states that the probability of X_{t+1} being in state s_{jt+1} , given the sequence of previous states up to time t (X_t , X_{t-1}, . . . , X_0) = is equal to the probability of X_{t+1} being in state s_{jt+1} , given only the current state X_t .

This property characterizes a **Markov chain as having no memory** of its past states beyond the current state.

Literature Review:
Q- Controller Equation for Reactive Power Set Point

Objective: Minimize voltage mismatch in distributed system in Volt-Var Mode

Equation: $$q_{t+1} = [q_t - D(V_t - 1)]^+$$

Where,

q_t	:	Reactive power at time step *t*
q_{t+1}	:	Reactive power at time step *t+1*
D	:	Diagonal matrix
V_t	:	Voltage at time step *t*
$[\cdot]+$	:	Projection if value exceeds range $[q^g, q^{-g}]$

Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Literature Review: *Bellman's optimality principle*

It is employed to determine the optimal policy for maximizing cumulative rewards.

This principle asserts that the optimal expected future cumulative reward for a given state s can be defined as the maximum expected sum of rewards achievable by selecting the best action in that state.

This is mathematically formalized by the Bellman optimality equation:

$V^*(s) = \max[R(s, a) + \gamma E_{P\,(s'|s,a)}\, [V^*(s')]] =: \max_a Q^*(s, a)$

In this equation, the maximization is performed over all possible actions a available in state s, where:

- $V^*(s)$ - optimal value function for state s.
- $R(s, a)$ - immediate reward obtained by taking action a in state s.
- γ - discount factor that determines the importance of future rewards relative to immediate rewards.
- $E_{P\,(s'|s,a)}\, [V^*(s')]$ - expected value of the optimal value function for the successor state s'.
- $P\,(s'|s, a)$ - state-transition function, which specifies the probability of transitioning to state s' given the current state s and action a.
- $Q^*(s, a)$ - optimal action-value function for the state-action pair (s, a).

Literature Review: SAC (Application of Reinforcement Learning)

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \left(R(s_t, a_t, s_{t+1}) + \alpha H \left(\pi(\cdot | s_t) \right) \right) \right]$$

$$H(P) = \mathbb{E}_{x \sim P}[-\log P(x)].$$

Literature Review: SAC (Key parameters)

1. **fc_dims:** Dimensions of the hidden layers of the agent's actor and critic networks. "fc" stands for "fully connected".

1. **update_after:** Specifies the number of environment interactions before starting gradient descent updates. This ensures that the replay buffer (a place where experiences of agent are stored) is adequately filled with diverse experiences before initiating the training process, affecting the initial delay in training.

1. **batch_size:** Defines the size of mini-batches used in each stochastic gradient descent update. It specifies the number of experiences sampled from the replay buffer to compute each update of the neural network weights, affecting the precision and efficiency of the gradient updates.

1. **update_every:** Determines the frequency of gradient descent updates after the initial delay specified by update_after. It controls how often the agent's policy and value function are updated based on experiences stored in the replay buffer, influencing the tempo of learning.

Literature Review: *Inverter Control Modes*

According to IEEE standard 1547-2018 [31], DERs are required to possess specific reactive power control functionalities, which include the following modes, each of which can be activated individually:

1. Constant power factor mode
2. **Voltage-reactive power (Volt-VAR) mode**
3. Active power-reactive power (Watt-VAR) mode
4. Constant reactive power mode

Literature Review: *Voltage-Reactive Power Mode*

In this mode, the DER actively regulates its reactive power output based on voltage levels, adhering to a voltage-reactive power piece-wise linear characteristic.

The mode includes autonomous adjustment of reference voltage and characteristics within specified parameters.

Literature Review:
Difference between State-value and Action-value functions

State-Value Function V_π (s): represents the expected return (cumulative future rewards) the agent can obtain from a given state s under a certain policy π. In other words, it quantifies the desirability of being in a particular state s and following a specific policy thereafter.

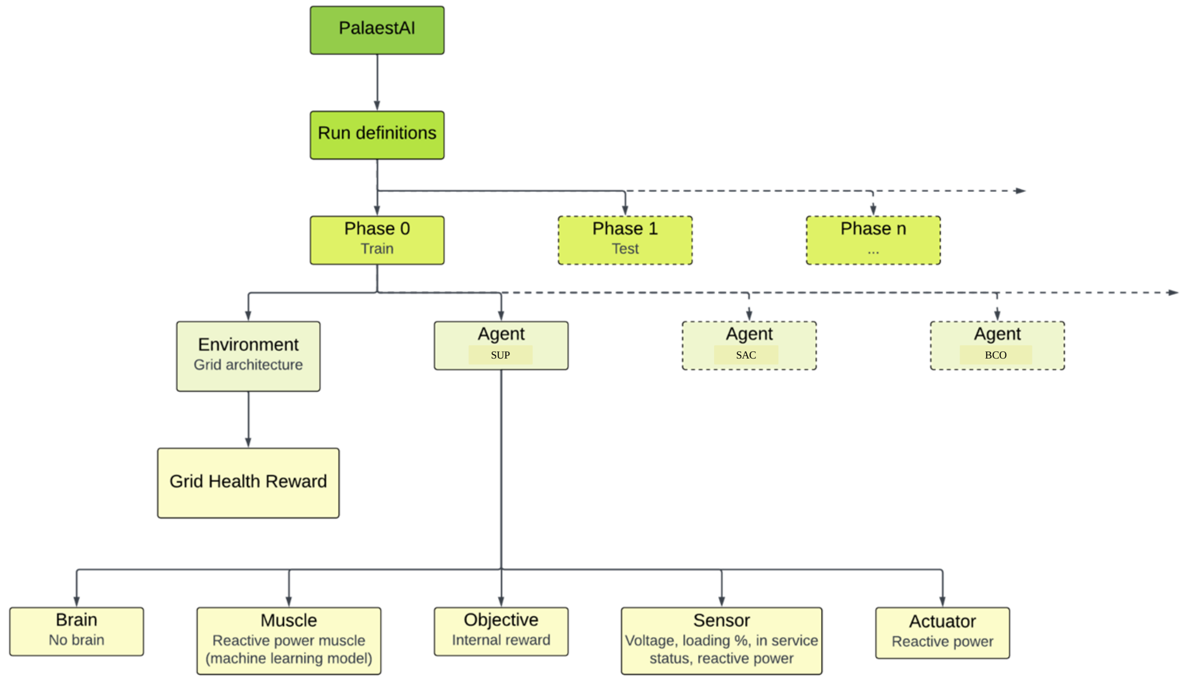
Action-Value Function Q_π (s, a): represents the expected return the agent can obtain by taking action a in state s and then following a certain policy π. It quantifies the desirability of taking a particular action a in a specific state s and following a specific policy thereafter[36]

Literature Review:
Difference between Deterministic (batch) & stochastic gradient descent

Deterministic (batch) gradient descent uses:
- entire dataset for each update and
- is more stable
- but slower

While **stochastic gradient descent** uses:
- a single example (or a mini-batch) for each update,
- which is faster and
- can handle larger datasets
- but introduces more noise into the optimization process.

System Description: *Agent*



System Description: *Agent*

Objective function

- Voltage levels of all buses
- Voltage of observed bus
- Operational buses unaffected by grid code violations

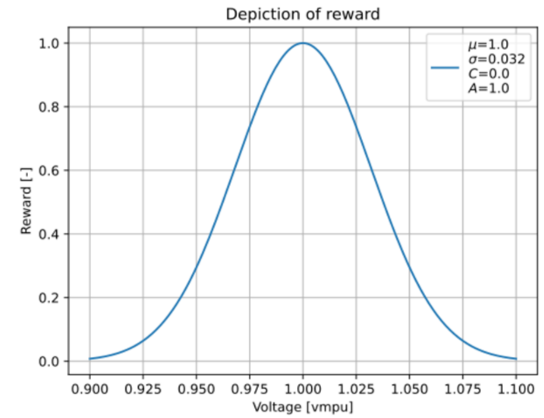
$$
\begin{aligned}
P_\Omega(\boldsymbol{m}^{(t)}) &= \alpha \cdot G_\Omega(\boldsymbol{x} = \boldsymbol{m}^{(t)}_{|V|}) \\
&\quad + \beta \cdot G_\Omega(\boldsymbol{x} = \Psi_\Omega(\boldsymbol{m}^{(t)}_{|V|})) \\
&\quad + \gamma \cdot \left\{ \sum_b \left[\!\left[\Psi_\Omega(\boldsymbol{m}^{(t)}_{|V|}) \right]\!\right]_b \right\} \left\{ |\boldsymbol{m}^{(t)}_{|V|}| \sum_b d^{-1} \right\}^{-1}
\end{aligned}
$$

No. of operational bus * distance from transformer

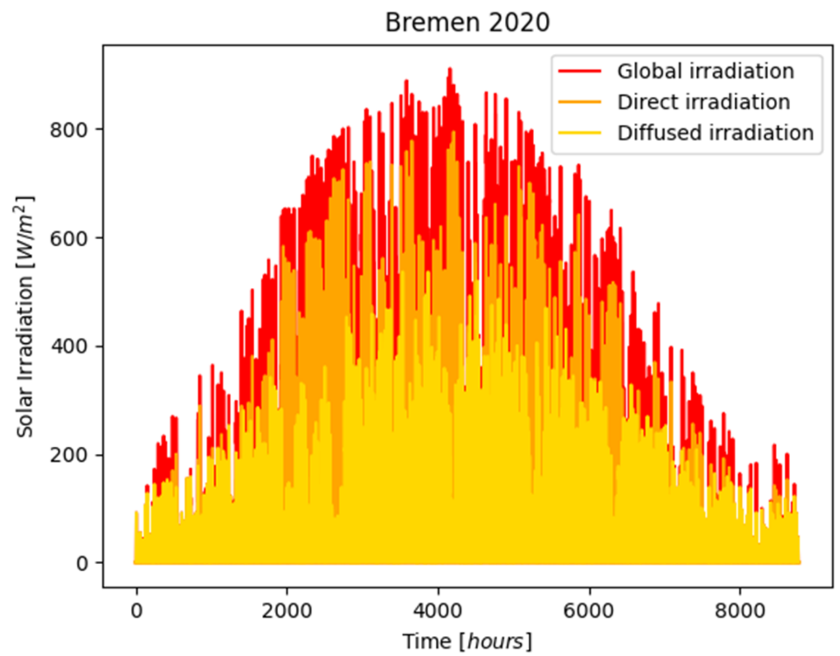
System Description: *Agent*

Objective function

- Voltage levels of all buses
- Voltage of observed bus
- Operational buses unaffected by grid code violations



System Description: *Weather Bremen 2020*

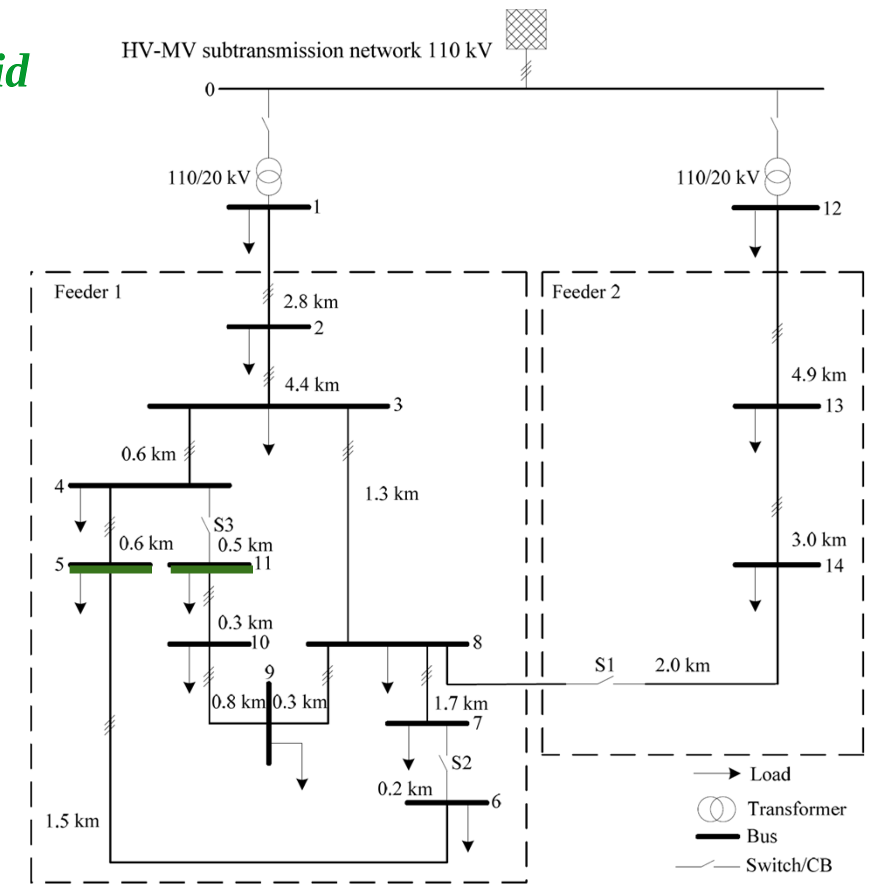


System Description: *Grid*

Bus number	No. of houses	Load [MWh/a]	Load Avg [kW]	Load Peak [kW]	Generation PV [kW]
1	41	130	14.8	39.9	700
	67	323	36.9	148.2	
	114	446	51	141.8	
	132	618	70.6	195	
	103	421	48.1	146.4	
2	67	323	36.9	148.2	700
	57	223	25.5	70.9	
	66	306	35.3	97.5	
	103	421	48.1	146.4	
3	82	260	29.6	79.8	800
	114	446	51	141.8	
	103	421	48.1	146.4	
4	114	446	51	141.8	900
	103	421	48.1	146.4	
5	82	260	29.6	79.8	600
	57	223	25.5	70.9	
6	82	260	29.6	79.8	800
	67	323	36.9	148.2	
	114	446	51	141.8	
	103	421	48.1	146.4	
7	41	130	14.8	39.9	400
	57	223	25.5	70.9	
	66	306	35.3	97.5	
8	67	323	36.9	148.2	600
	57	223	25.5	70.9	
	132	618	70.6	195	
	103	421	48.1	146.4	
9	82	260	29.6	79.8	600
	67	323	36.9	148.2	
	57	223	25.5	70.9	
	132	618	70.6	195	
	103	421	48.1	146.4	
10	41	130	14.8	39.9	600
	67	323	36.9	148.2	
	57	223	25.5	70.9	
	132	618	70.6	195	
	103	421	48.1	146.4	
11	-	-	-	800	-
12	-	-	-	600	-
13	-	-	-	400	-

Sr.No.	Facility	[MWh/a]	Avg [kW]	Peak [kW]
1	Super Market	2343.541	0.268	0.568
2	Small Hotel	1147.85	0.131	0.27

System Description: *Grid*



Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

Why SAC, and not any other algorithm

1. **State-of-the-Art Performance:**
   - *High Performance:* SAC is known for its high performance on continuous action space tasks, often outperforming other algorithms in terms of learning efficiency and final performance.
   - *Robustness:* SAC demonstrates robustness and stability in training, making it a reliable choice for comparing against other algorithms.
2. **Exploration and Exploitation Balance:**
   - *Entropy Regularization:* SAC uses an entropy term in its objective function, encouraging exploration by preventing the policy from becoming too deterministic too quickly. This balance between exploration and exploitation can lead to better overall performance.
3. **Sample Efficiency:**
   - *Off-Policy Learning:* SAC is an off-policy algorithm, meaning it can reuse past experiences stored in a replay buffer. This significantly improves sample efficiency compared to on-policy algorithms, which require new data for each update.
4. **Scalability:**
   - *Scalability to High Dimensions:* SAC can handle high-dimensional state and action spaces, making it suitable for complex tasks with large observation and action spaces.

Why SAC, and not any other algorithm

5. **Stability:**
   - *Stabilized Training:* SAC incorporates techniques such as clipped double Q-learning and slow delayed updates of target networks, which help stabilize training by reducing the overestimation bias common in value-based methods.
6. **Wide Adoption and Benchmarking:**
   - *Benchmarking:* SAC is widely used and benchmarked in the RL community, providing a solid reference point for comparison. Its performance on standard benchmarks can help validate the effectiveness of other algorithms.

Other possible algorithms

Model-Free Algorithms

1. **Deep Q-Network (DQN)**
2. **Twin Delayed Deep Deterministic Policy Gradient (TD3)**
3. Proximal Policy Optimization (PPO)
4. Trust Region Policy Optimization (TRPO)
5. A3C (Asynchronous Advantage Actor-Critic)

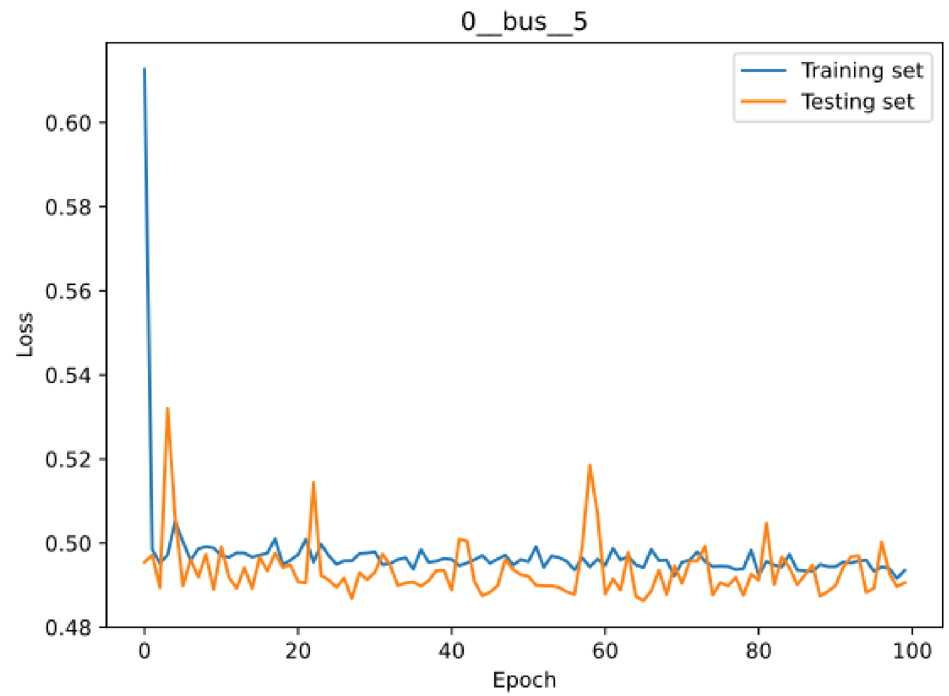
Grid Architecture

OFFIS

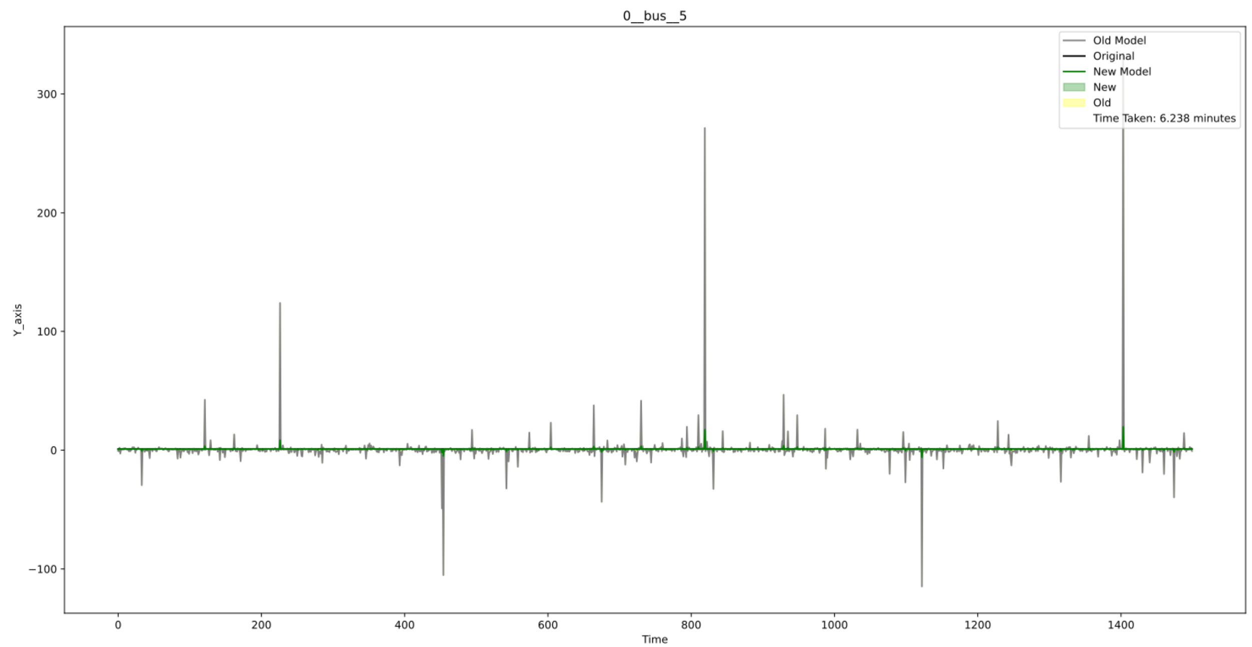
1. **Why Bus 5?**
   a. **Intermediate position:** Bus 5 is in the middle of the grid. It is in feeder 1 and is affected by the all the buses along the line, bus 2, 3 and 4. Therefore, victim of all the changes on other buses.
   b. **Impact of Bus 5**: is primarily only on one other bus. This way we can see the impact of changes on this bus 5 on ONLY one other bus. Not having too much influence on the behavior of the other buses.

1. **Why Bus 5 and 11?**
   - **Parallel buses:** Chosen for comparison because it runs parallel to Bus 5, providing a comparative node that can offer insights into voltage behavior across different, yet parallel, paths of the same feeder.

Neural Network Optimization : 5000 Data Point Model



Neural Network Optimization : 5000 Data Point Model



Neural Network Optimization : 5000 Data Point Model

Description	Default [-]	Optimized [-]
Activation function	ReLU	Linear
Learning rate	0.001	0.0355
Number of neurons	10	4
Number of layers	3	3
Batch size	10	23
Epoch	100	100

Neural Network Optimization : To analyze the models

OFFIS

Mean

$$\mu = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Coefficient of Determination

$$R^2 = 1 - \frac{SS_E}{SS_T}$$

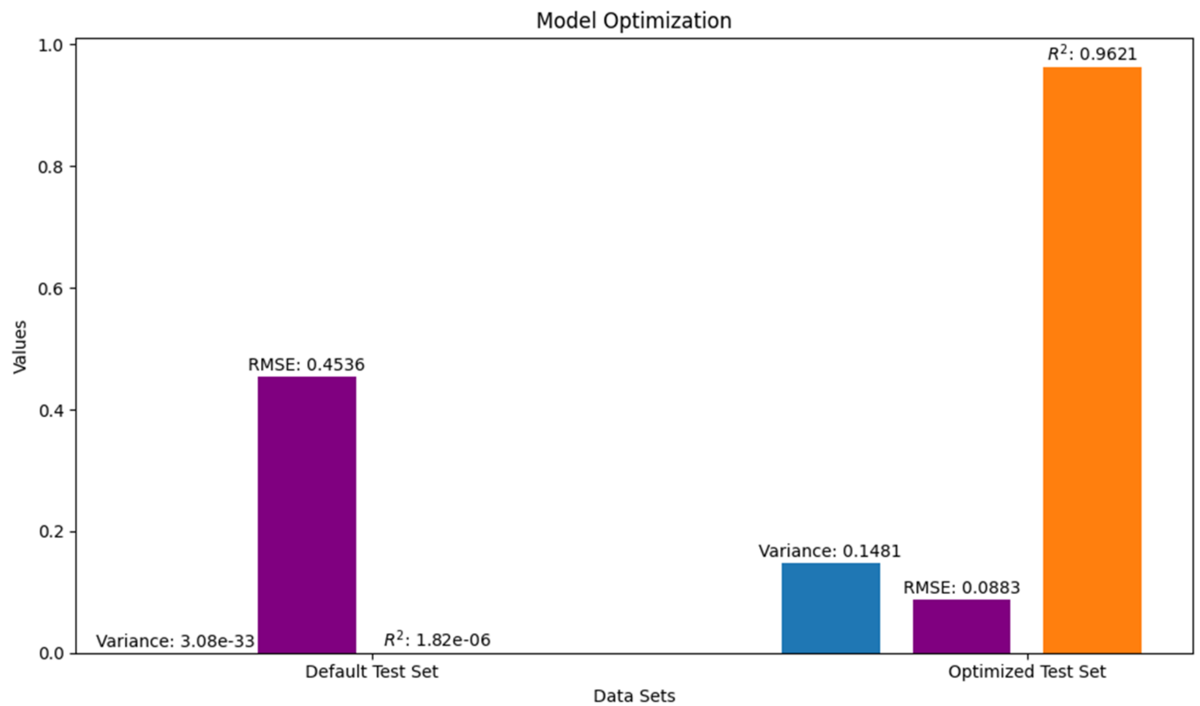
Variance

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mu)^2$$

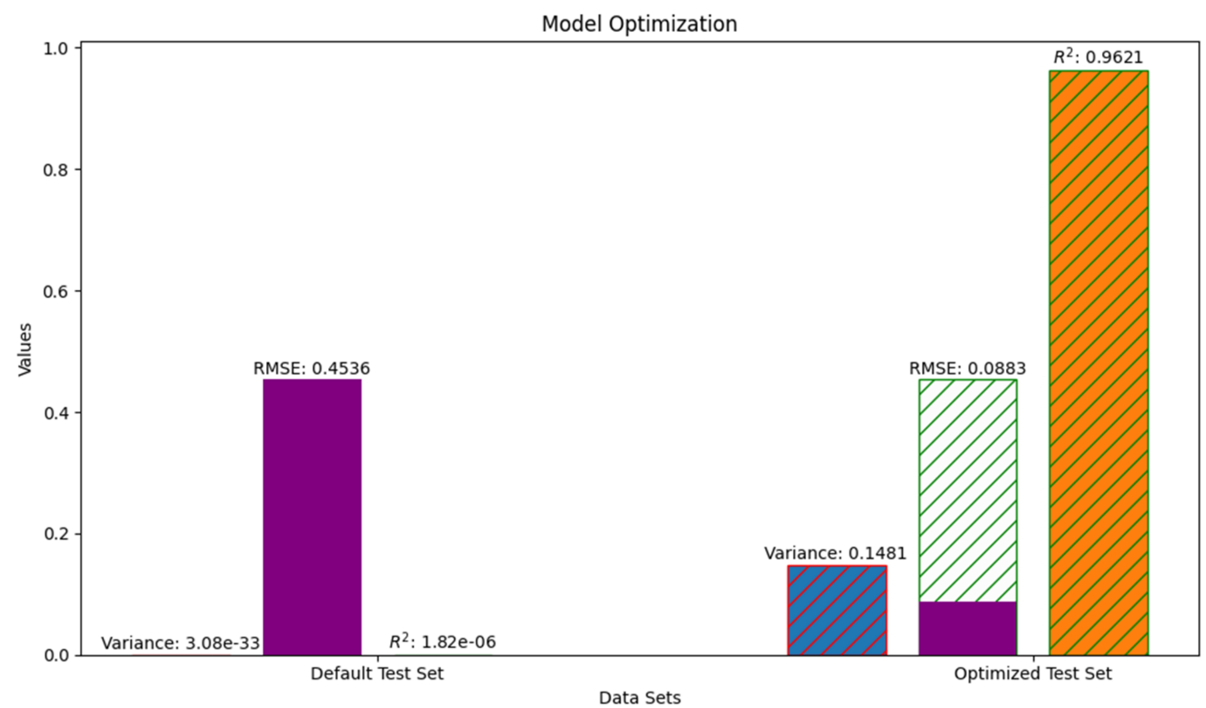
Root of MSE

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Neural Network Optimization



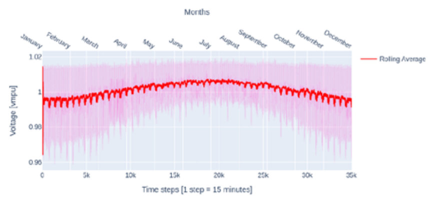
Neural Network Optimization



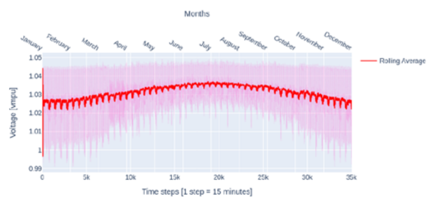
Variance
Accuracy
Good Fit
Precision

Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

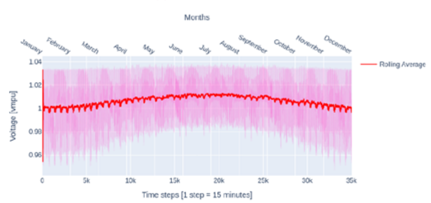
Sample Efficient Model for SUP vs BCO: *Voltage*



(a) 5000-Data-Point Model

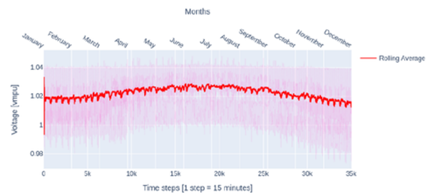


(b) 10000-Data-Point Model

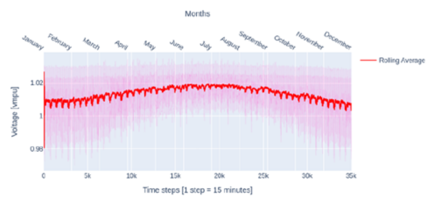


(c) 25000-Data-Point Model

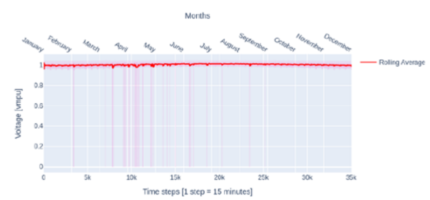
Fig. 5.4: Itachi: Voltage Performance



(a) 22500-Data-Point Model



(b) 30000-Data-Point Model



(c) 35040-Data-Point Model

Fig. 5.15: Kitachi: Voltage Performance

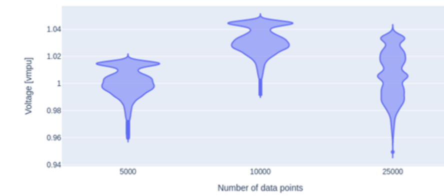


Fig. 5.5: Itachi: Voltage Distribution of the models generated from 5000, 10000 and 25000 data points respectively

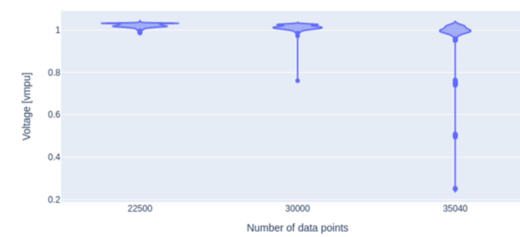
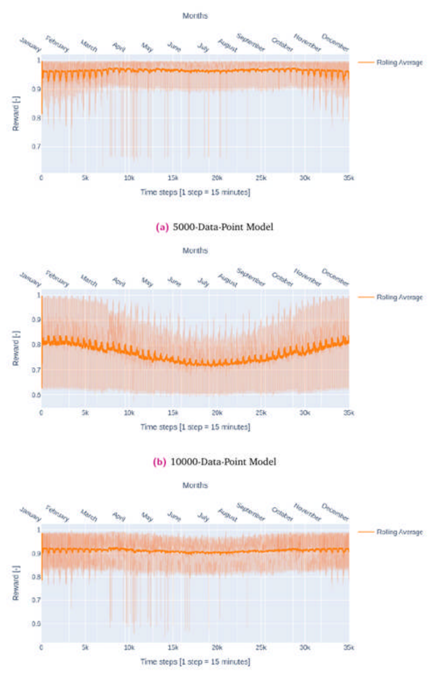
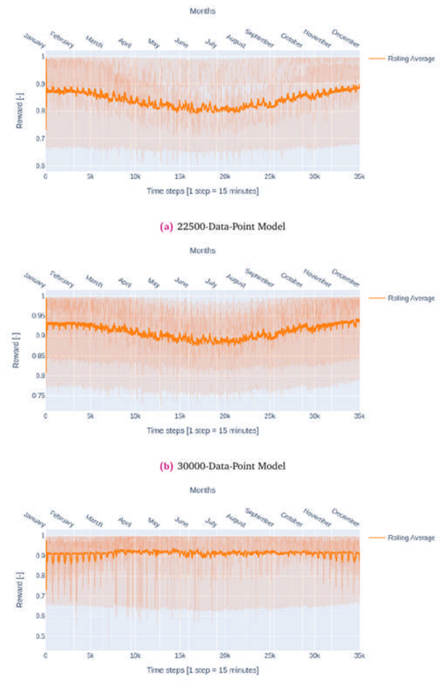


Fig. 5.16: Kitachi: Voltage Distribution of the models generated from 22500, 30000 and 35040 data points respectively

Sample Efficient Model for SUP vs BCO: *Reward*



(a) 5000-Data-Point Model
(b) 10000-Data-Point Model
(c) 25000-Data-Point Model
Fig. 5.6: Itachi: Reward Performance



(a) 22500-Data-Point Model
(b) 30000-Data-Point Model
(c) 35040-Data-Point Model
Fig. 5.17: Kitachi: Reward Performance

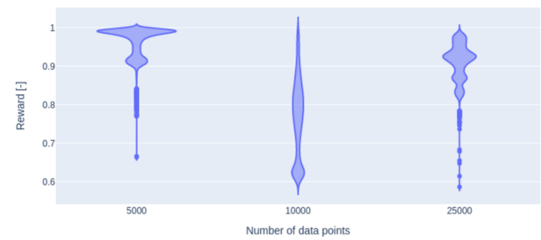


Fig. 5.7: Itachi: Reward Distribution of the models generated from 5000, 10000 and 25000 data points respectively

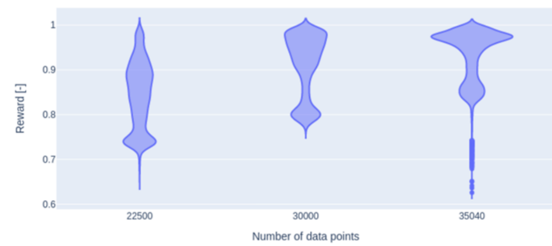


Fig. 5.18: Kitachi: Reward Distribution of the models generated from 22500, 30000 and 35040 data points respectively

Optimizing Hyperparameters: SAC vs BCO

Tab. 7.1: Parameter Combinations

update_after	update_every	batch_size	update_after	update_every	batch_size
250	250	250	250	250	500
500	250	250	500	250	500
1250	250	250	1250	250	500
2000	250	250	2000	250	500
250	500	250	250	500	500
500	500	250	500	500	500
1250	500	250	1250	500	500
2000	500	250	2000	500	500
250	1250	250	250	1250	500
500	1250	250	500	1250	500
1250	1250	250	1250	1250	500
2000	1250	250	2000	1250	500
250	2000	250	250	2000	500
500	2000	250	500	2000	500
1250	2000	250	1250	2000	500
2000	2000	250	2000	2000	500
250	250	1250	250	250	2000
500	250	1250	500	250	2000
1250	250	1250	1250	250	2000
2000	250	1250	2000	250	2000
250	500	1250	250	500	2000
500	500	1250	500	500	2000
1250	500	1250	1250	500	2000
2000	500	1250	2000	500	2000
250	1250	1250	250	1250	2000
500	1250	1250	500	1250	2000
1250	1250	1250	1250	1250	2000
2000	1250	1250	2000	1250	2000
250	2000	1250	250	2000	2000
500	2000	1250	500	2000	2000
1250	2000	1250	1250	2000	2000
2000	2000	1250	2000	2000	2000

Tab. 7.2: Mean, Standard Deviation, and Median Values

Value	Update_after	Update_every	Batch_size
250	x	x	x
500	x	x	x
1250	x	x	x
2000	x	x	x

Optimizing Hyperparameters: SAC vs BCO

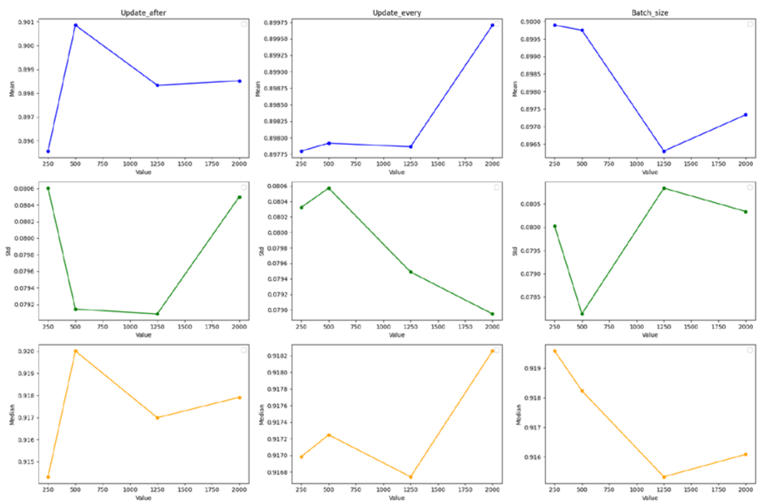


Fig. 5.10: SAC: Efficient Hyperparameters - Reward Analysis

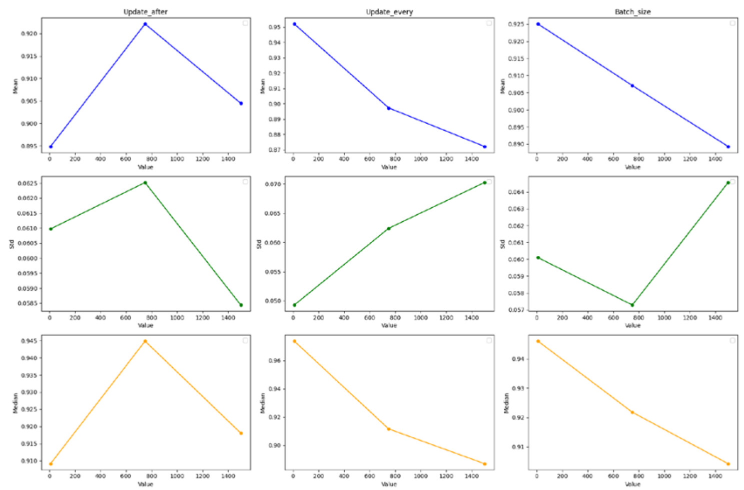
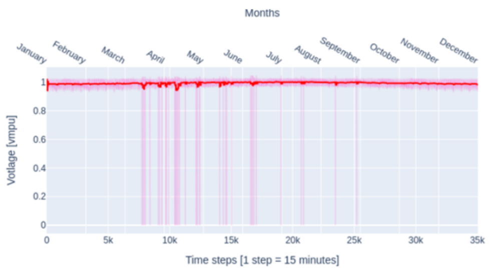
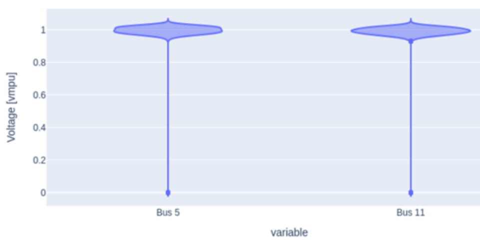


Fig. 5.19: Kitachi: Efficient Hyperparameters - Reward Analysis

Two buses: Itachi vs SAC vs BCO: *Voltage*

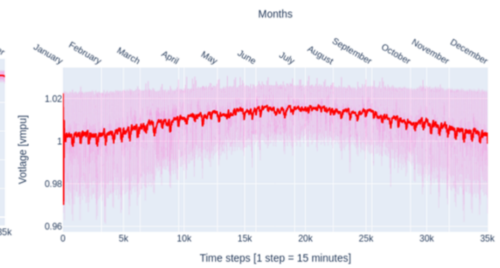


(a) Voltage Performance of Buses 5 and 11

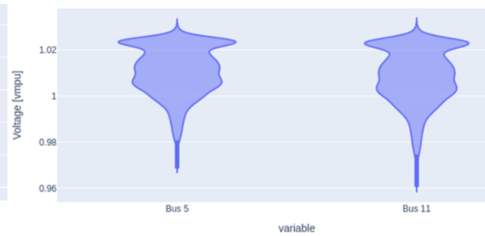


(b) Voltage Distribution of Buses 5 and 11

Fig. 5.8: Itachi: Robustness Analysis - Voltage

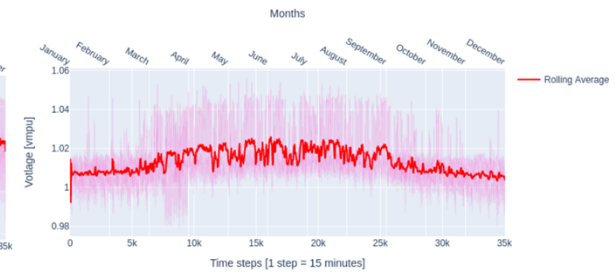


(a) Voltage Performance of Buses 5 and 11

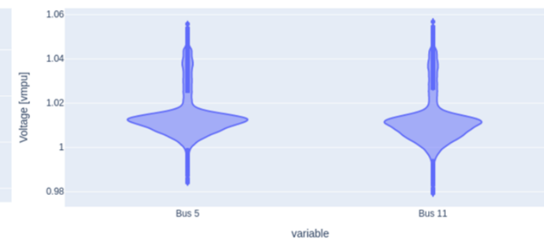


(b) Voltage Distribution of Buses 5 and 11

Fig. 5.13: SAC: Robustness Analysis - Voltage



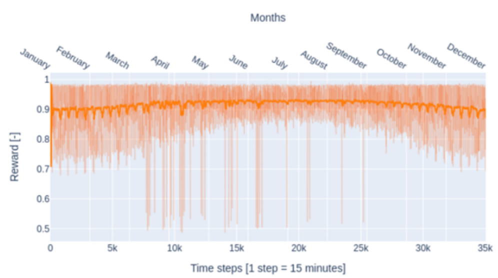
(a) Voltage Performance of Buses 5 and 11



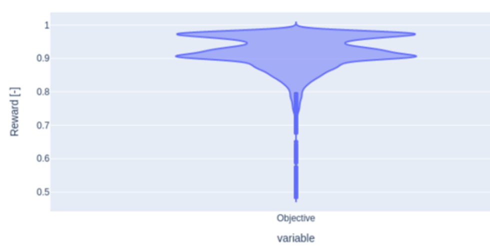
(b) Voltage Distribution of Buses 5 and 11

Fig. 5.22: Kitachi: Robustness Analysis - Voltage

Two buses: Itachi vs SAC vs BCO: *Reward*

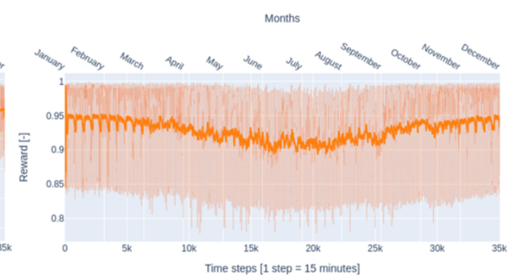


(a) Reward Performance of Buses 5 and 11

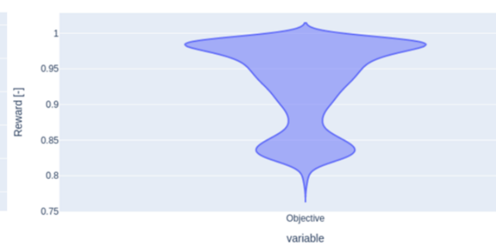


(b) Reward Distribution of Buses 5 and 11

Fig. 5.9: Itachi: Robustness Analysis - Reward



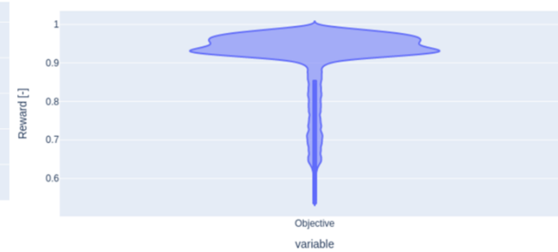
(a) Reward Performance of Buses 5 and 11



(b) Reward Distribution of Buses 5 and 11

Fig. 5.14: SAC: Robustness Analysis - Reward

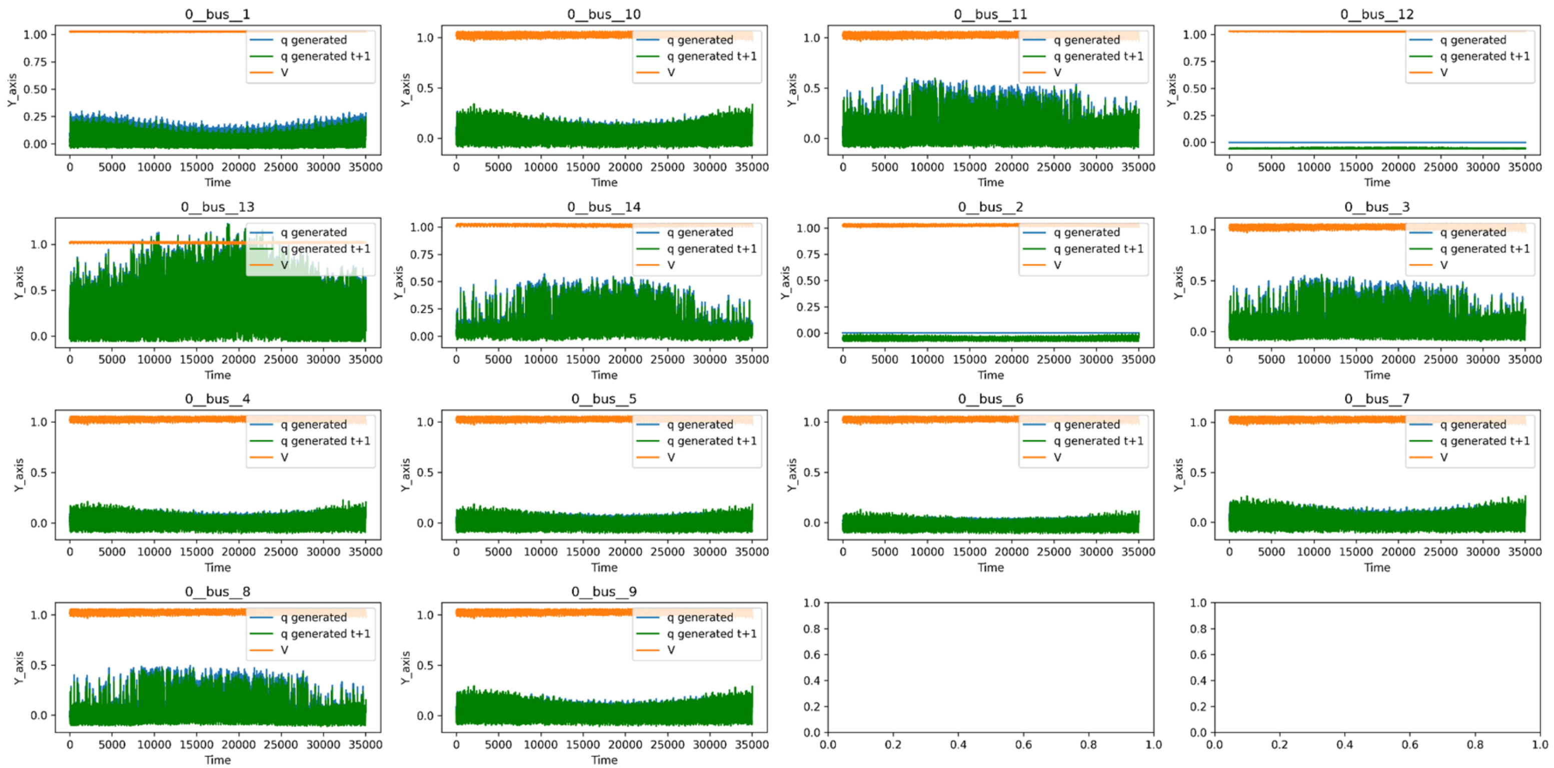
(a) Reward Performance of Buses 5 and 11



(b) Reward Distribution of Buses 5 and 11

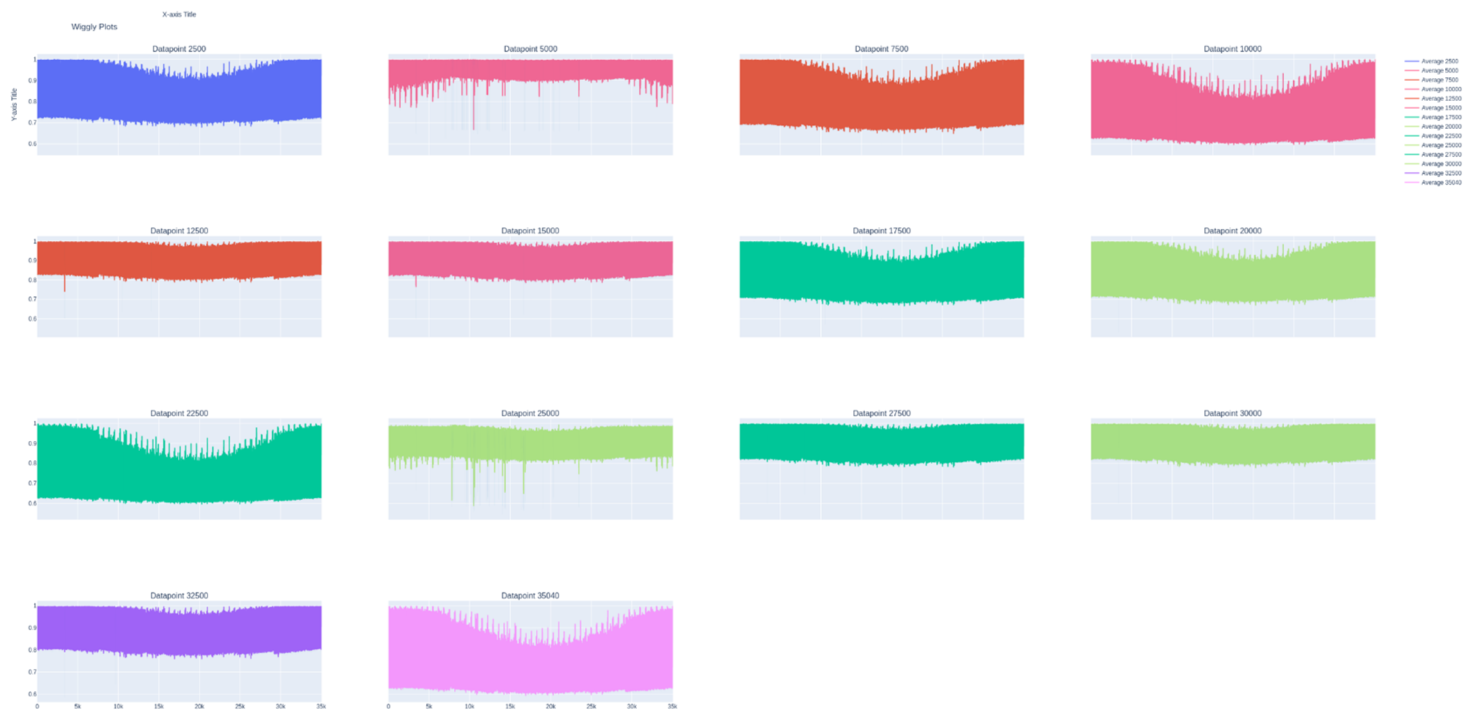
Fig. 5.23: Kitachi: Robustness Analysis - Reward

All Buses Performance with Q-Controller equation

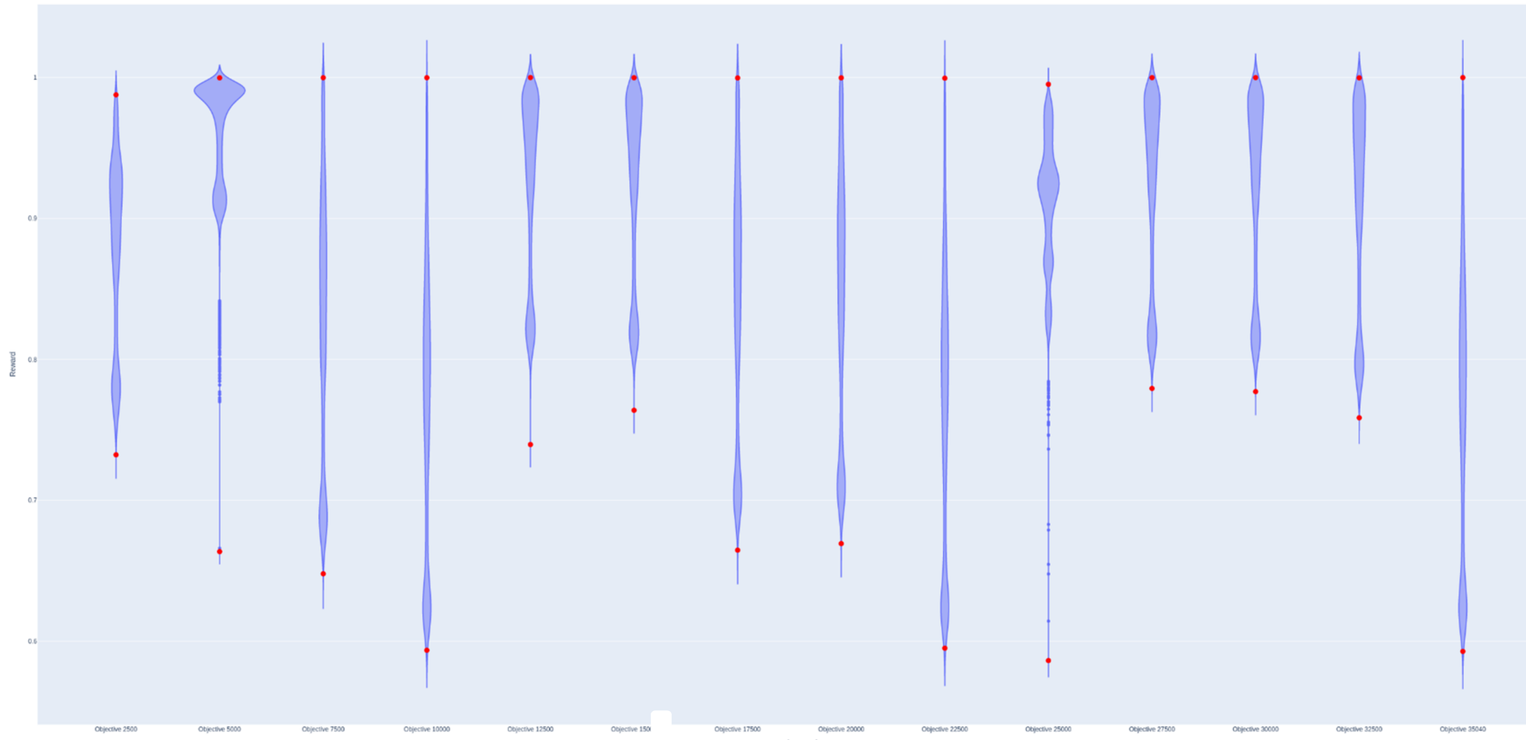


Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

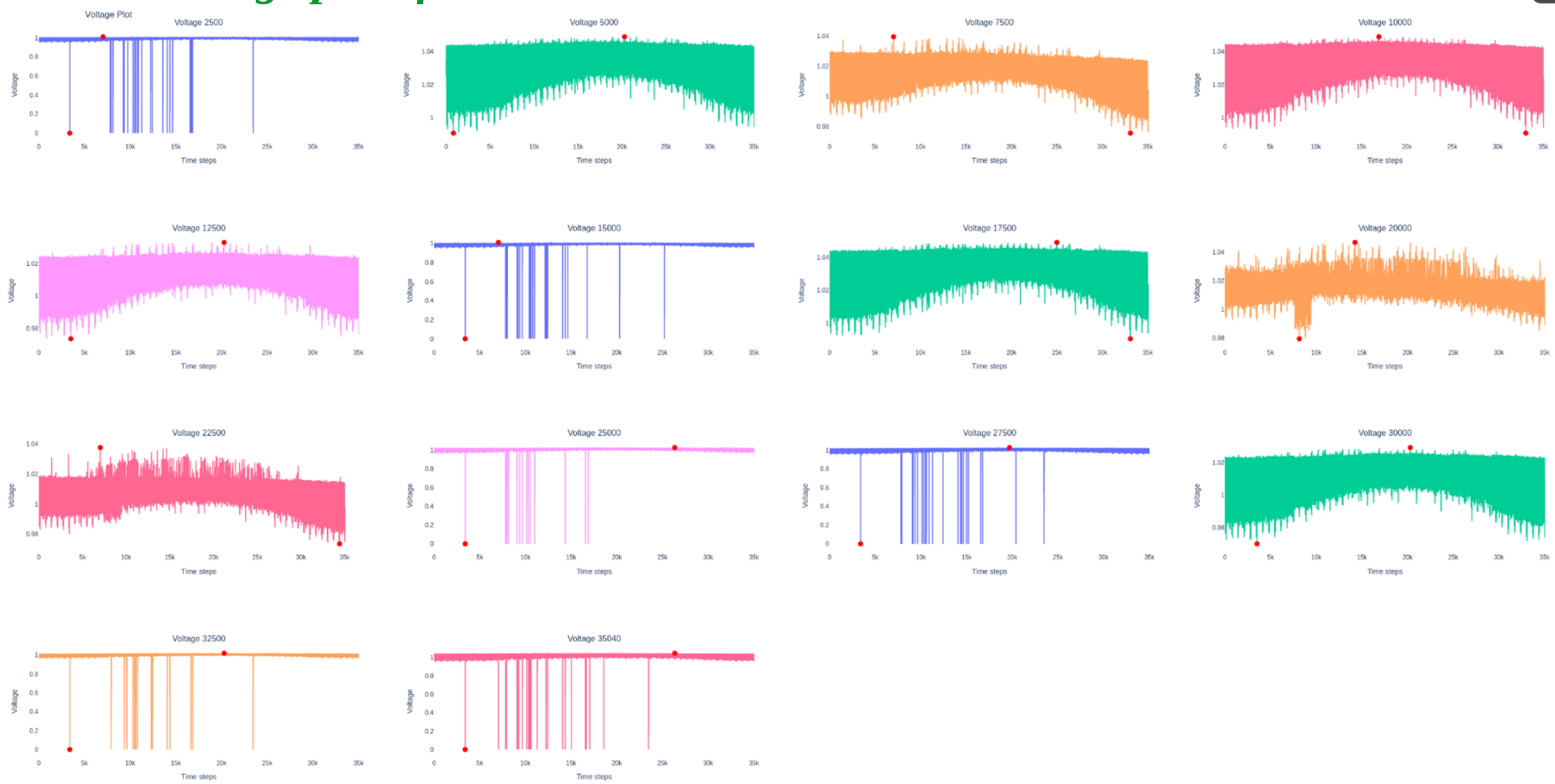
SUP: *Voltage plots for all dataset combinations*



SUP: *Voltage plots for all dataset combinations*

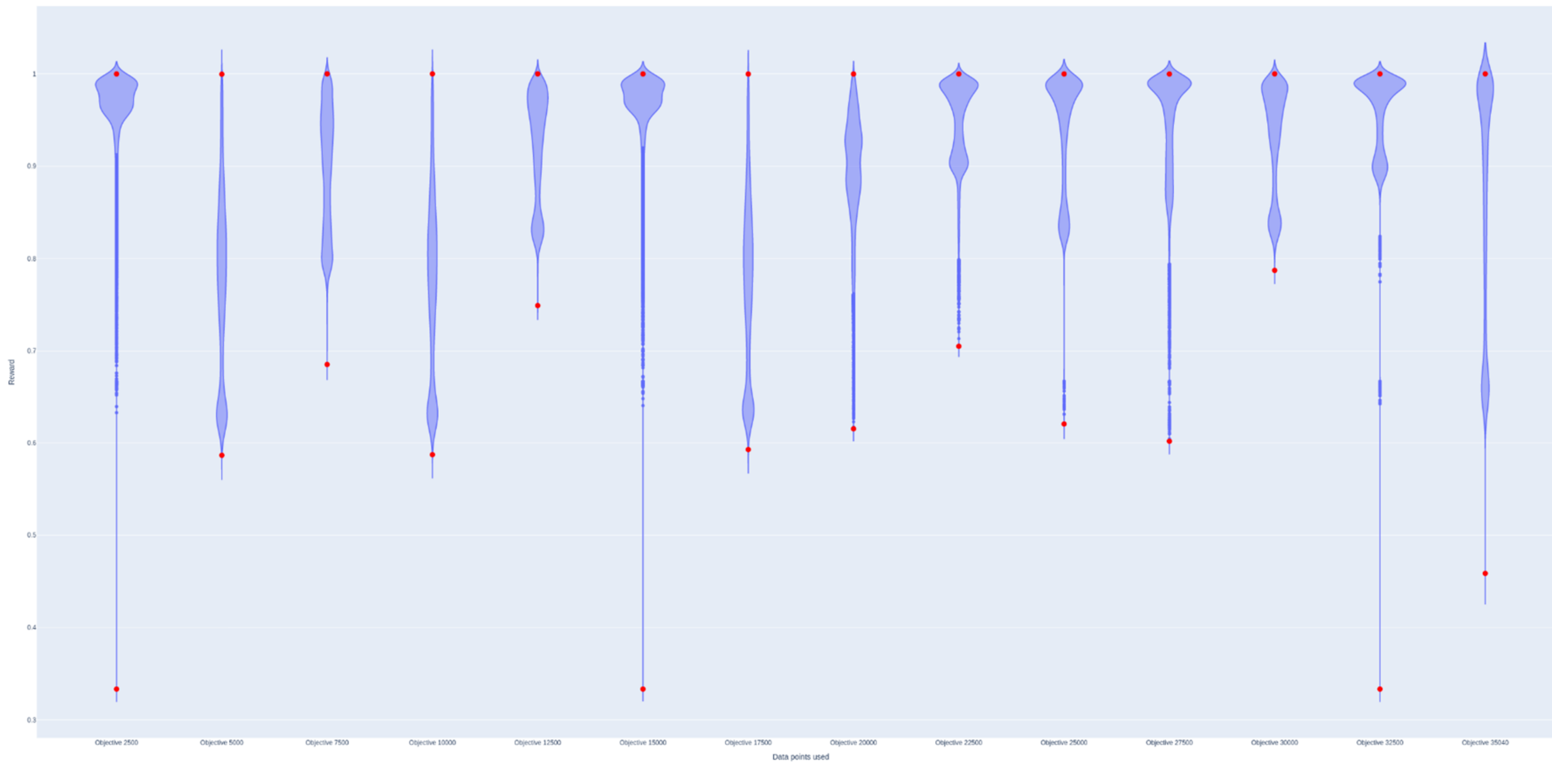


BCO: *Voltage plots for all dataset combinations*



Introduction | Algorithms | System Description | Methodology | Evaluation | Conclusion

BCO: *Voltage plots for all dataset combinations*



Future Work

1. Developing a more advanced neural network model may enhance performance. The current study utilizes randomized search for model hyperparameter optimization, but other methods such as grid search and genetic algorithms could be explored.
   - The neural network architecture in this study is classical, with an input layer, output layer, and a few hidden layers. Investigating more complex architectures may be beneficial, especially when examining the agent's control of the entire grid through 14 buses. Given that each bus may exhibit distinct behaviors, a more intricate architecture could better capture these patterns.
2. Modifying the objective function to penalize undesirable voltage variations and violations can ensure stricter compliance with grid code standards.
3. Increasing the number of repetitions for each experiment can provide deeper insights into performance variations and strengthen confidence in the results.

Challenges

1. Challenges encountered during the thesis included working with a code base that was still in development.
2. Each simulation lasted for a lengthy period of 2 hours and 40 minutes, posing a bottleneck for executing a large number of cases. Although resources like DGX, a high-performance computing system developed by NVIDIA, were available, it was deemed unreliable at the time. Consequently, the decision was made to conduct simulations solely on the local laptop, although at the expense of longer simulation times.
3. This limitation resulted in SAC hyperparameter optimization being based on a single run, leading to reduced confidence in the results.