

# System do objaśniania akcji polityk systemu zarządzania zasobami chmurowymi z wykorzystaniem trenowania przez głębokie uczenie ze wzmocnieniem

Andrzej Małota

promotor: dr inż. Włodzimierz Funika

# Plan prezentacji



1. Przedstawienie problemu zarządzania zasobami chmury obliczeniowej
2. Cel pracy
3. Rozwiązanie problemu
  - a. Środowisko symulacyjne
  - b. Agent DQN vs agent PPO
  - c. Graficzna reprezentacja wpływu cech
4. Wyniki
  - a. Agenci reaktywni vs proaktywni
  - b. Podsumowanie polityki agenta
  - c. Śledzenie postępów w treningu
  - d. Selekcja cech
5. Podsumowanie

# Przedstawienie problemu zarządzania zasobami chmury obliczeniowej



- ❖ celem zarządzania zasobami chmury obliczeniowej jest minimalizacja kosztu infrastruktury dla aplikacji użytkownika przy zachowaniu wysokiej jakości usług,
- ❖ rozwiązanie poprzez użycie agentów głębokiego uczenia ze wzmocnieniem,
  - lepsze rezultaty niż standardowe rozwiązania,
  - użycie w produkcji jest uznawane za ryzykowne z powodu braku przejrzystości zasad podejmowania decyzji: nie wiadomo, dlaczego konkretna akcja została wykonana.

## Cel pracy



- ❖ rozszerzenie istniejącego eksperymentalnego systemu autonomicznego zarządzania zasobami obliczeniowymi chmury używającego techniki skalowania poziomego opartego o agentów głębokiego uczenia ze wzmocnieniem, poprzez:
  - implementacje algorytmu dokonującego lokalnej oraz globalnej interpretacji post hoc agenta, poprzez wyliczenie wpływu poszczególnych wejściowych metryk infrastruktury chmurowej na decyzje agenta oraz przedstawienie ich w formie graficznej,
  - przeprowadzenie procesu trenowania agentów oraz ich analiza pod kątem interpretowalności.

# Rozwiązanie problemu (1)

## - środowisko symulacyjne



Agenci byli trenowani w symulatorze chmury obliczeniowej. Metryki infrastruktury chmury to:

- ❖ liczba maszyn wirtualnych w użyciu,
- ❖ średnie oraz 90 percentyl użycia pamięci RAM,
- ❖ średnie oraz 90 percentyl obciążenia CPU,
- ❖ całkowity oraz niedawny czas oczekiwania zadań w kolejce.

Funkcja nagrody = - (koszt działania infrastruktury + kara za opóźnienie wykonania zadań)

Dostępne akcje agenta:

- ❖ brak akcji,
- ❖ dodanie/usunięcie małej/średniej/dużej maszyny wirtualnej.

## Rozwiązanie problemu (2) - interpretacja agenta



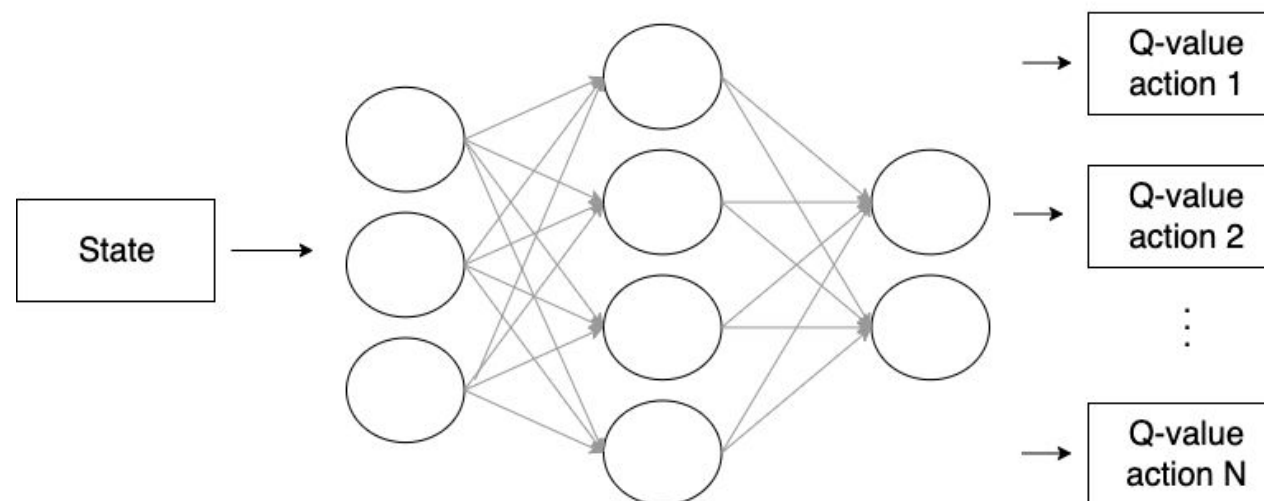
- ❖ aby zrozumieć tok rozumowania agenta głębokiego uczenia można dokonać interpretacji post hoc głębokiej sieci neuronowej na której zbudowana jest jego polityka,
- ❖ do interpretacji sieci neuronowej został użyty algorytm Integrated Gradients (IG),
- ❖ podejście do interpretacji agenta oraz jej graficzna reprezentacja zależy od użytego algorytmu uczenia (DQN/PPO) który definiuje architekturę sieci neuronowej oraz typu pierwszej warstwy sieci (w pełni połączona/konwolucyjna).

# Rozwiązanie problemu (3)

## - agent DQN



- ❖ aby zinterpretować agenta Deep Q-Network (DQN) trzeba policzyć wpływ metryk wejściowych na wyjście z ostatniej warstwy którymi są Q-values,
- ❖ wartość Q-value dla akcji informuje nas o oczekiwanym zwrocie (suma przyszłych nagród uzyskanych przy użyciu optymalnej polityki) z wykonania danej akcji w aktualnym stanie,
- ❖ przy deterministycznym działaniu agenta i przy użyciu chciwej polityki, agent zawsze wykonuje akcje powiązaną z największą wartością Q-value.



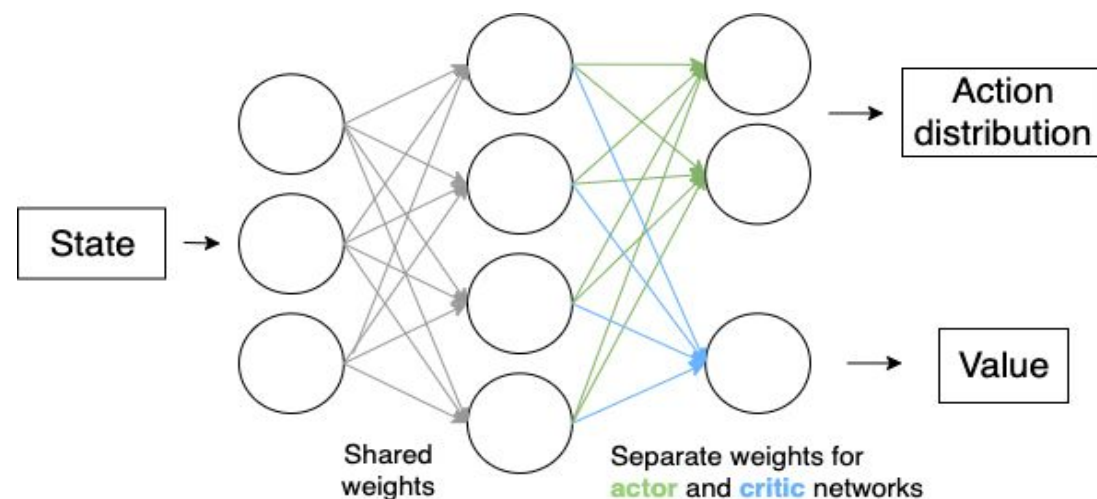
**Rys. 1** Schemat sieci neuronowej agenta DQN.

# Rozwiązanie problemu (4)

## - agent PPO



- ❖ agent trenowany algorytmem Proximal Policy Optimization (PPO) posiada architekturę Aktor-Krytyk, główna sieć neuronowa rozgałęzia się na dwie podsieci: sieć Aktora oraz Krytyka,
- ❖ Krytyk estymuje funkcję wartościującą natomiast Aktor optymalizuje politykę oraz zwraca predykcje akcji,
- ❖ ponieważ w tej pracy interesuje nas interpretacja akcji agenta, trzeba policzyć wpływ metryk wejściowych na wyjście z ostatniej warstwy sieci Aktora.



**Rys. 2** Schemat sieci neuronowej agenta PPO.



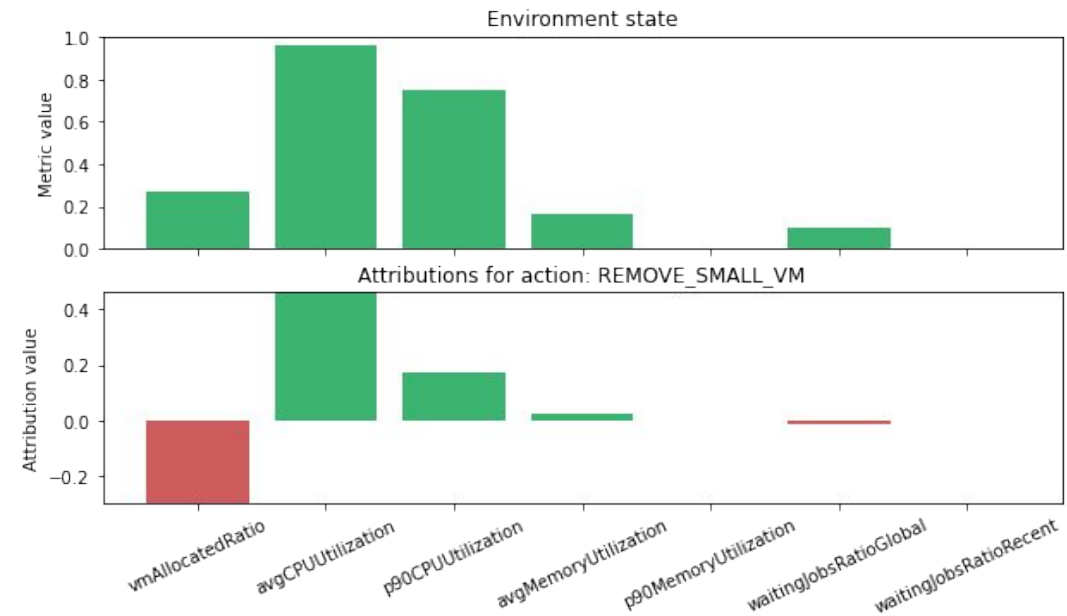
# Rozwiązanie problemu (5)

## - warstwa w pełni połączona



- ❖ w zależności od pierwszej warstwy sieci neuronowej otrzymujemy różne graficzne reprezentacje atrybucji cech,
- ❖ dla w pełni połączonej warstwy wejściowej wpływ metryk przedstawiony jest na wykresie słupkowym (przykład poniżej),
- ❖ dodatnia wartość (na zielono) oznacza pozytywny wpływ cechy na akcje, pozytywny wpływ oznacza że dana cecha przyczyniła się do predykcji danej akcji, ujemna wartość (na czerwono)

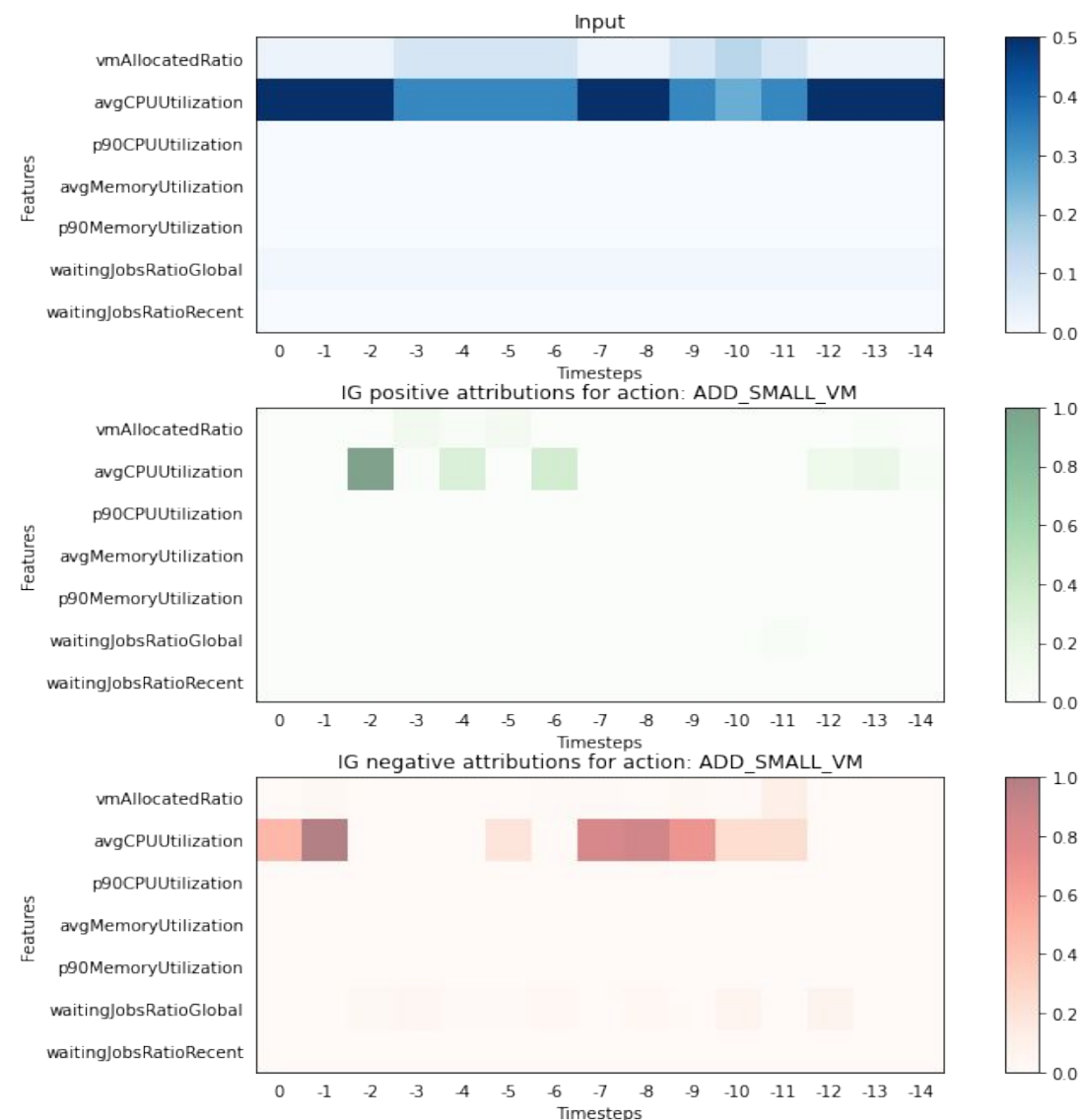
**Rys. 3** Graficzna reprezentacja wpływu metryk wejściowych na decyzje agenta z pierwszą warstwą w pełni połączoną w postaci dwóch wykresów słupkowych: metryk środowiska (górny), atrybucje metryk (dolny). Wspólna oś x - metryki środowiska, Oś y dla górnego wykresu - wartość metryk, dla dolnego wykresu - wartość wpływu.



# Rozwiązanie problemu (6) - warstwa konwolucyjna

- ❖ dla konwolucyjnej warstwy wejściowej wpływ metryk przedstawiony jest jako dwuwymiarowa mapa istotności.

**Rys. 4** Graficzna reprezentacja wpływu metryk wejściowych na decyzje agenta z pierwszą warstwą konwolucyjną w postaci trzech map cieplnych przedstawiających: wartość metryk środowiska (górna), atrybucje dodatnie (środkowa) oraz atrybucje ujemne (dolna). Wspólna oś x - kroki czasowe (0 - aktualny krok czasowy, -14 - czternasty krok w przeszłości), Oś y - metryki środowiska. Im bardziej intensywny kolor tym większa wartość.

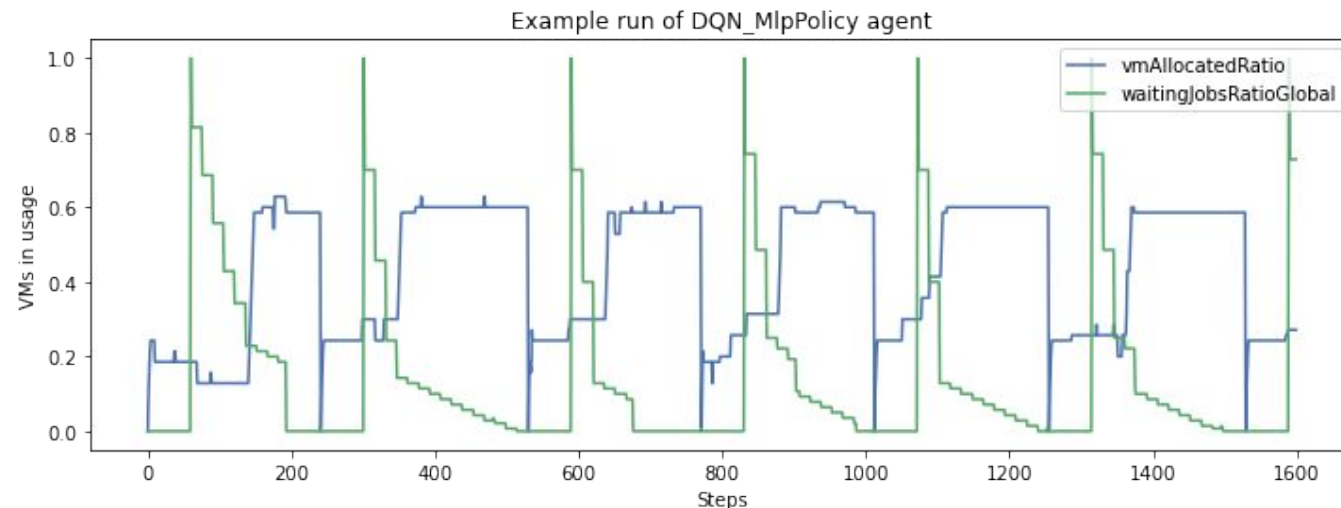


# Wyniki analizy (1)

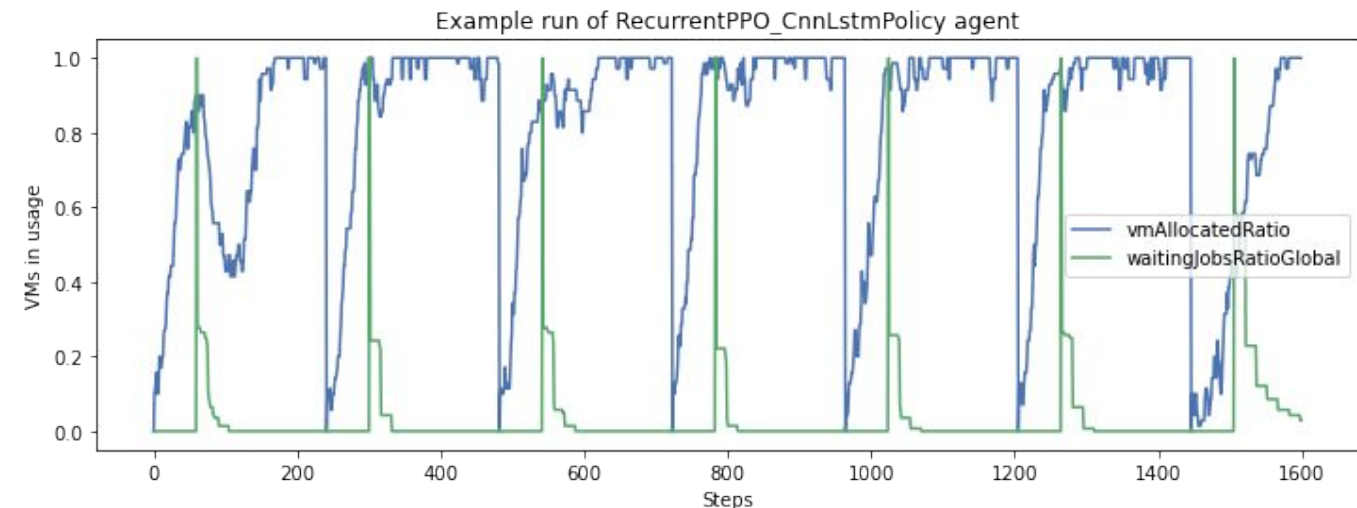
## - agenci reaktywni vs proaktywni



- ❖ dzięki analizie reakcji agentów na gwałtowne skoki obciążenia infrastruktury można wyróżnić dwa typy agentów: reaktywnych i proaktywnych,
- ❖ agenci reaktywni z opóźnieniem reagują na zmiany obciążenia,
- ❖ agenci proaktywni są w stanie przewidzieć zmiany obciążeniu wywołane przez aplikację kliencką i zapewnić niezbędne zasoby obliczeniowe na czas.



**Rys. 5** Przykładowy przebieg z działania reaktywnego agenta DQN z architekturą MLP.



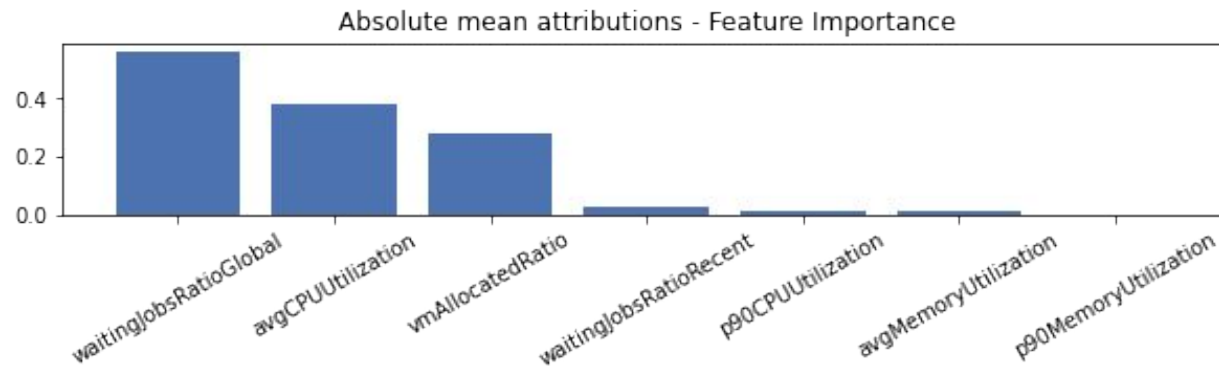
**Rys. 6** Przykładowy przebieg z działania proaktywnego agenta PPO z architekturą CNN-LSTM.

## Wyniki analizy (2)

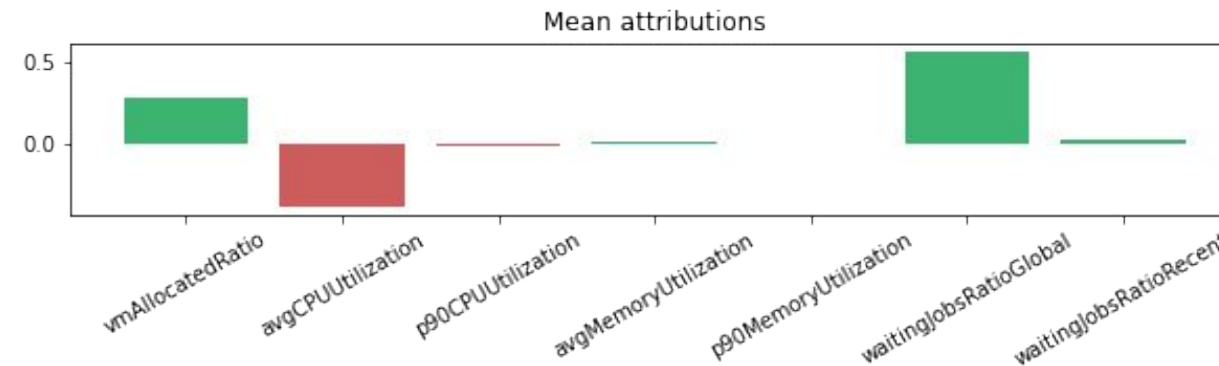
### - podsumowanie polityki agenta



- ❖ aby znaleźć generalne wzorce zachowań agenta można podsumować politykę agenta przy pomocy globalnej interpretacji, obliczonej na dwa sposoby:
  - średnia absolutna atrybucja cech (ważność/istotność cech),
  - średnia atrybucja cech.



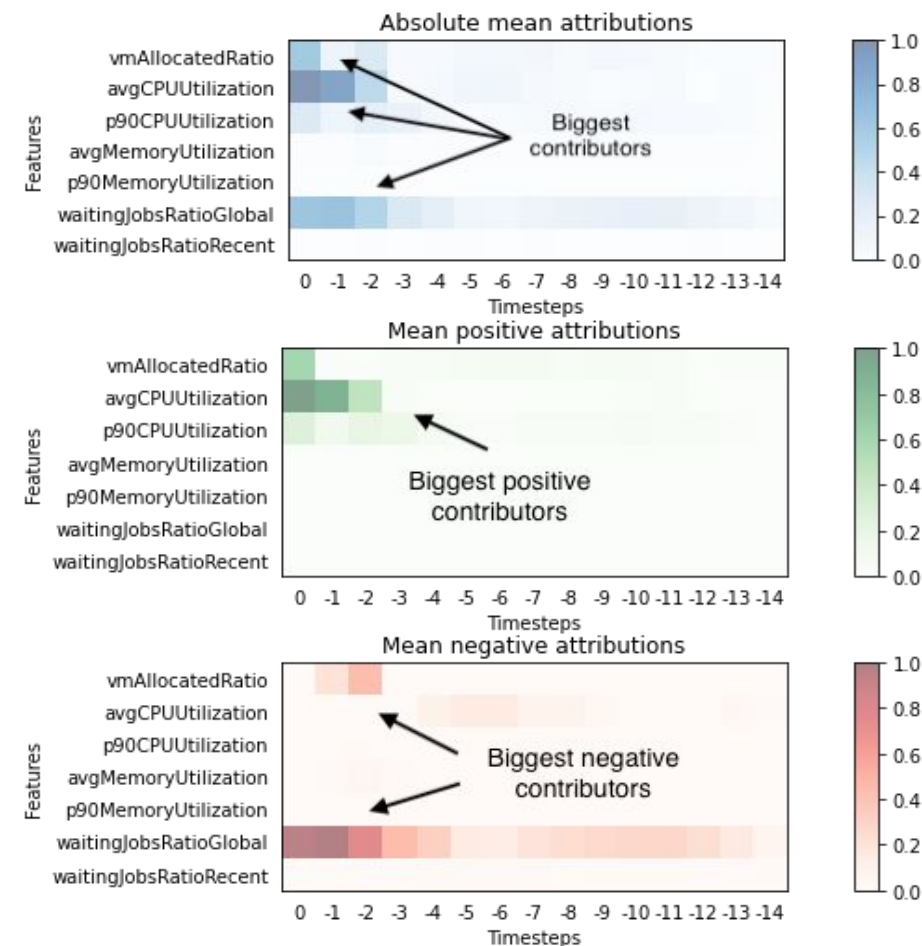
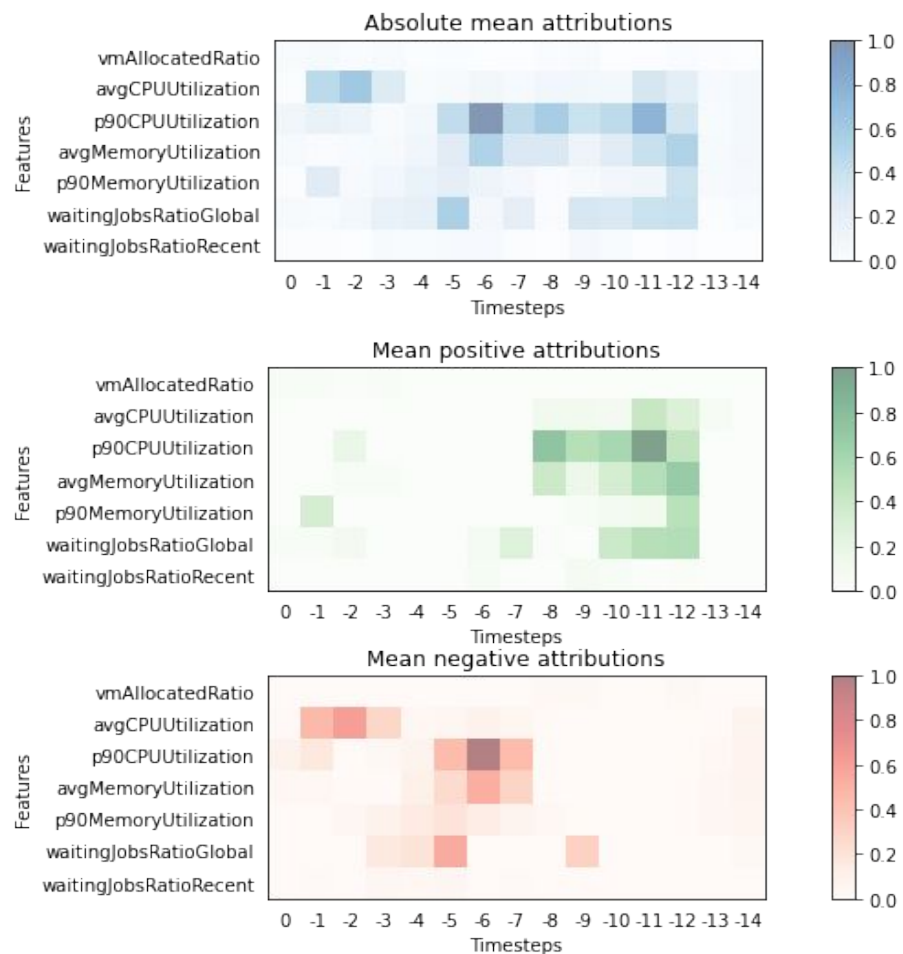
**Rys. 7** Ważność cech agenta PPO z architekturą MLP.



**Rys. 8** Średnia atrybucja cech agenta PPO z architekturą MLP.

# Wyniki analizy (3)

## - podsumowanie polityki agenta



**Rys. 9** Ważność cech agenta PPO z architekturą CNN-LSTM.

**Rys. 10** Ważność cech agenta DQN z architekturą CNN.

## Wyniki analizy (3)

### - śledzenie postępów w treningu



- ❖ śledzenie procesu uczenia się agentów polega na obserwacji zmian atrybucji cech pomiędzy kolejnymi iteracjami agenta,
- ❖ śledzenie zostało zrobione na dwa sposoby:
  - przy użyciu interpretacji globalnej
    - śledzenie zmian ważności cech aby zaobserwować zmiany w generalnym działaniu agenta,
  - przy użyciu interpretacji lokalnej
    - śledzenie zmian ważności cech dla konkretnej obserwacji wejściowej w celu analizy jak kolejne iteracje agentów sobie z nią radzą.

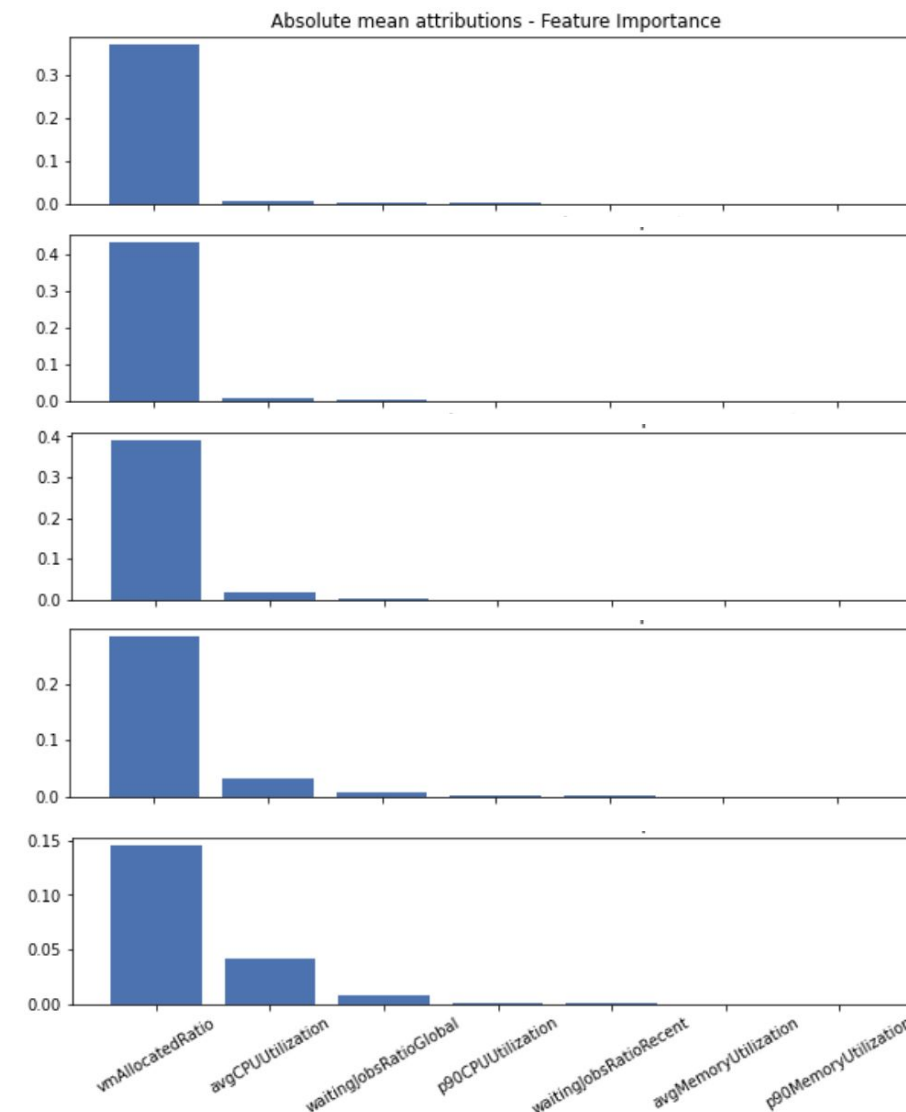


# Wyniki analizy (4)

## - śledzenie postępów w treningu



**Rys. 11** Grupa wykresów przedstawiających zmiany ważności cech agenta DQN z architekturą MLP podczas jego treningu. Wykresy ułożone są od góry do dołu względem długości treningu agentów, od agenta z losowymi parametrami do agenta po treningu.

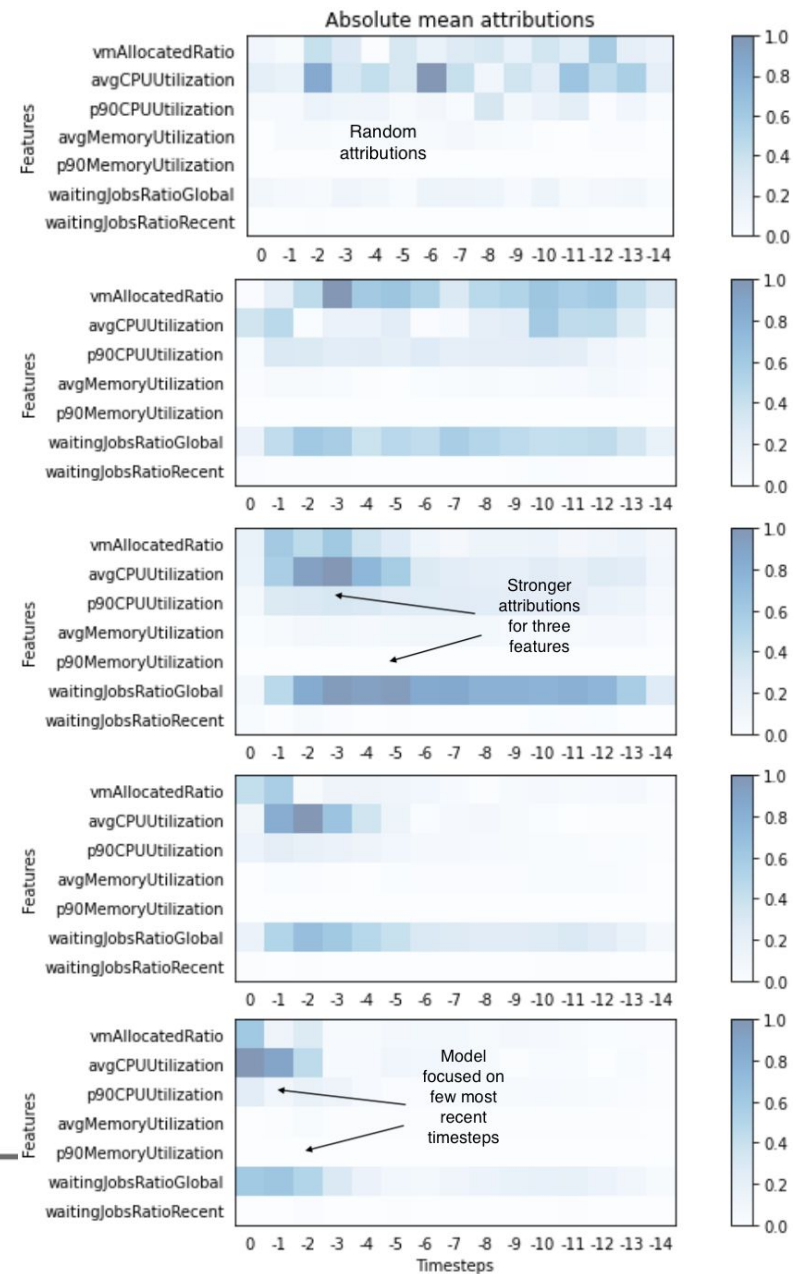


# Wyniki analizy (5)

## - śledzenie postępów w treningu



**Rys. 12** Grupa wykresów przedstawiających zmiany ważności cech agenta DQN z architekturą CNN podczas jego treningu.



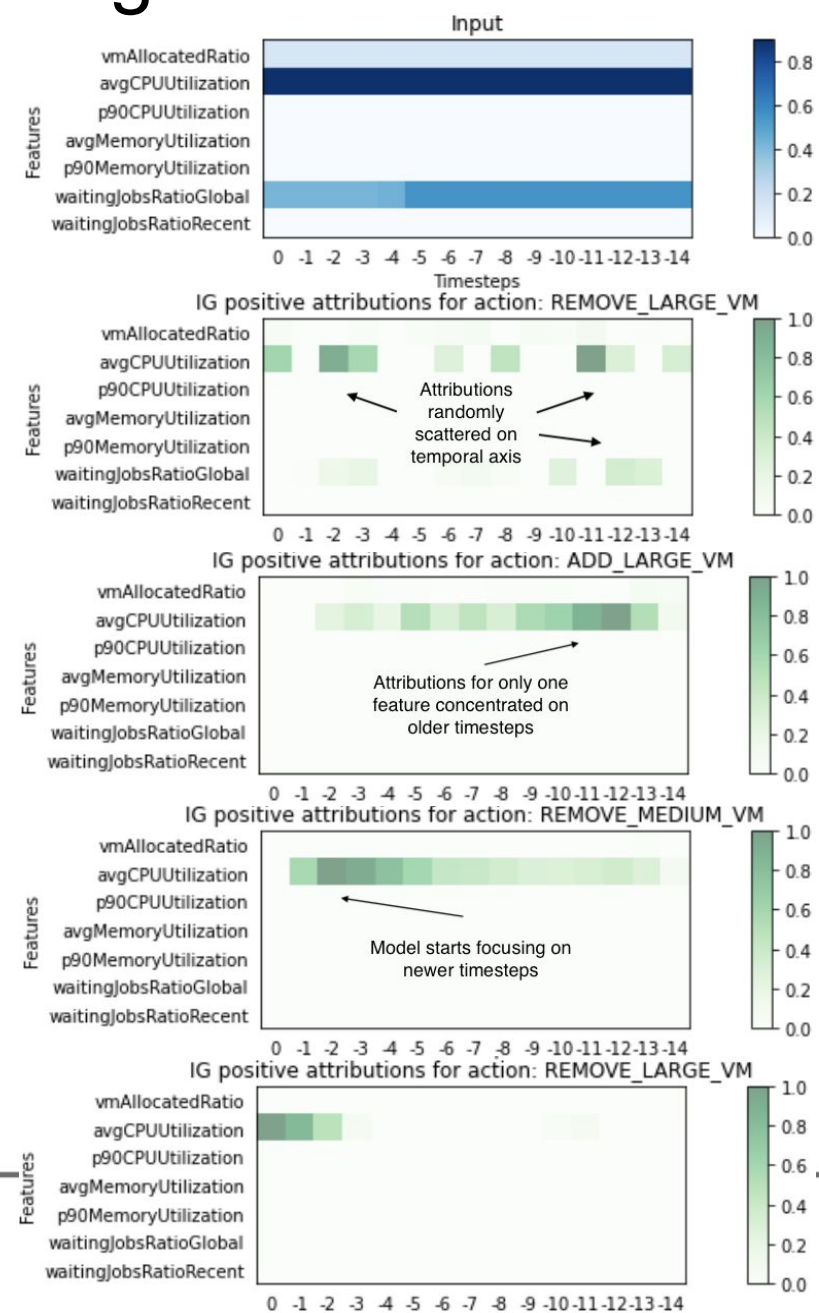


# Wyniki analizy (6)

## - śledzenie postępów w treningu



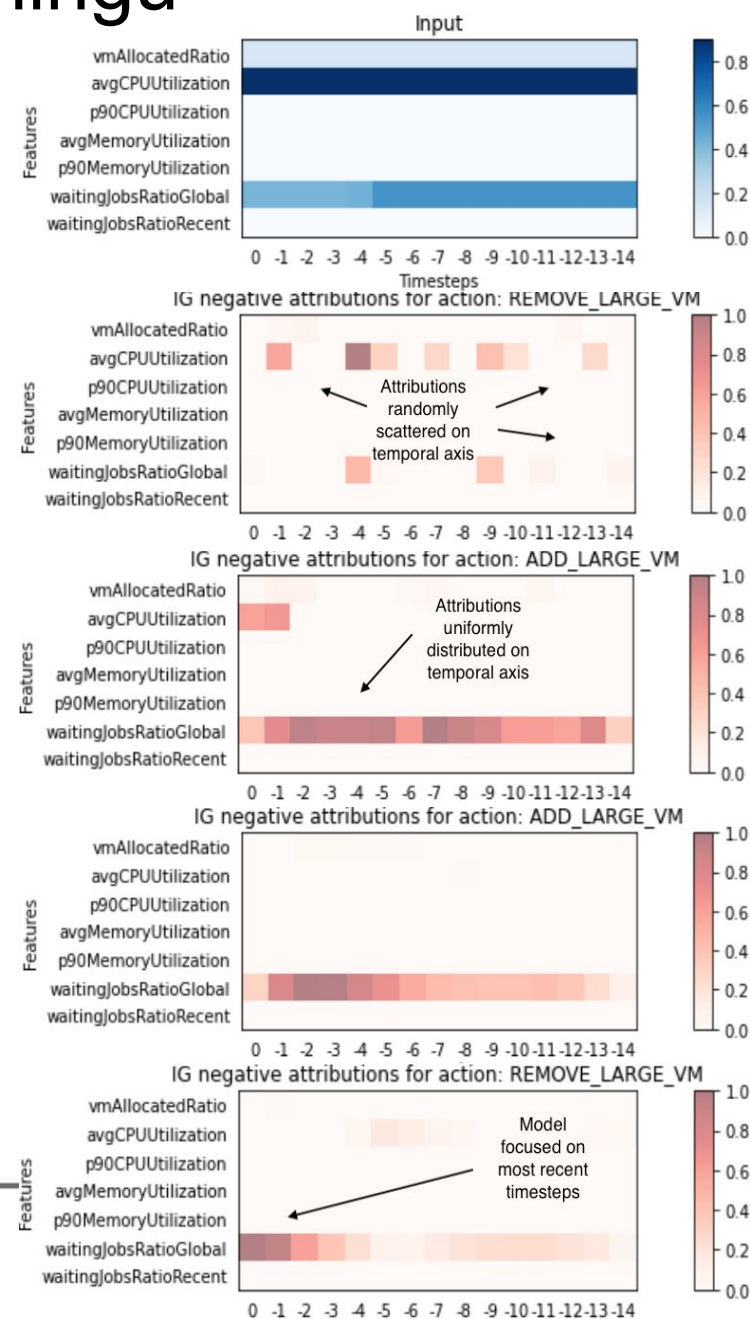
**Rys. 13** Grupa wykresów przedstawiających zmiany pozytywnych atrybucji cech agenta DQN z architekturą CNN dla tej samej obserwacji podczas jego treningu.



# Wyniki analizy (7)

## - śledzenie postępów w treningu

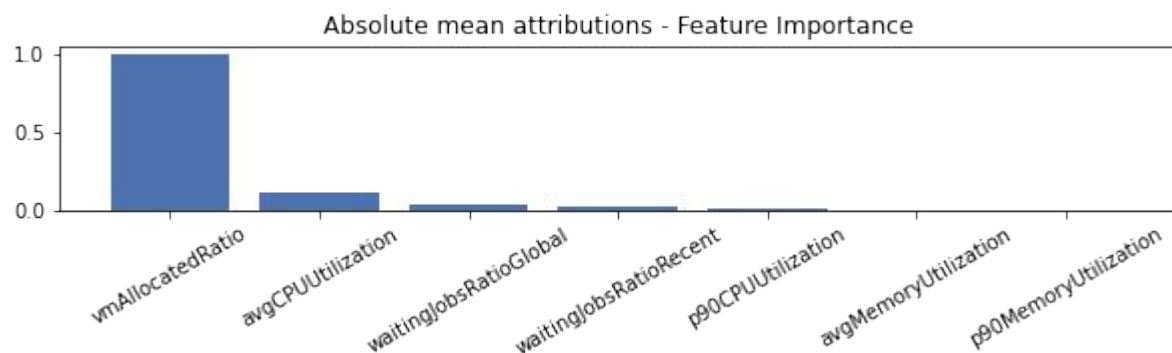
**Rys. 14** Grupa wykresów przedstawiających zmiany negatywnych atrybucji cech agenta DQN z architekturą CNN dla tej samej obserwacji podczas jego treningu.



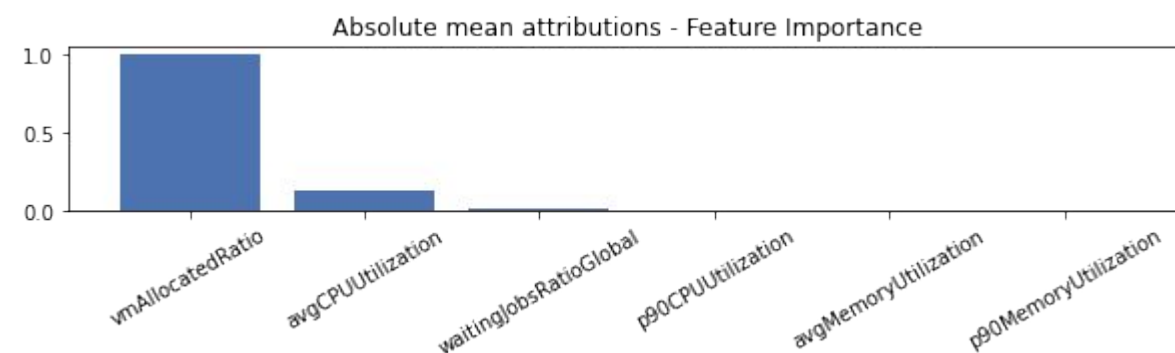
## Wyniki analizy (8) - selekcja cech



- ❖ ostatnim krokiem analizy była selekcja cech na podstawie globalnej interpretacji agenta PPO z architekturą rekurencyjną,
- ❖ ze środowiska została usunięta cecha “waitingJobsRatioRecent” która miała prawie zerowy wpływ na decyzje agenta - jej brak nie spowodował spadku wydajności agenta.



**Rys. 15** Ważność cech oryginalnego agenta PPO z architekturą rekurencyjną.



**Rys. 16** Ważność cech agenta PPO z architekturą rekurencyjną po usunięciu metryki “waitingJobsRatioRecent”.

# Podsumowanie (1)

## - wnioski



- ❖ agenci głębokiego uczenia ze wzmocnieniem są w stanie optymalnie zarządzać zasobami obliczeniowymi chmury oraz być zrozumiałymi przez człowieka bez eksperckiej wiedzy w zakresie uczenia maszynowego oraz działania chmury obliczeniowej,
- ❖ w zależności od ustawień środowiska symulacyjnego oraz typu pierwszej warstwy sieci neuronowej otrzymujemy dwie graficzne reprezentacje wpływu metryk wejściowych na decyzje:
  - dla w pełni połączonej warstwy otrzymujemy jednowymiarowy wektor wpływów bieżącej wartości metryk -> prostsze do interpretacji, mniej informacji,
  - dla konwolucyjnej warstwy otrzymujemy dwuwymiarową mapę istotności wpływu metryk z kilkunastu ostatnich kroków czasowych -> bardziej skomplikowane w interpretacji ale dostarcza więcej informacji,
- ❖ globalna interpretacja agenta może być użyta do znalezienia generalnych wzorców jego zachowań,
- ❖ lokalna interpretacja agenta może być użyta do debuggowania niepoprawnych decyzji,
- ❖ różne architektury sieci neuronowej oraz typy agenta osiągają podobną wydajność w zarządzaniu zasobami używając różnych wzorców zachowań.

## Podsumowanie (2)

### - dalsze prace



- ❖ użycie komponentu który interpretuje agentów w systemie działającym w czasie rzeczywistym do asystowania ludzkiemu operatorowi w podejmowaniu decyzji,
- ❖ dodanie znacznie większej liczby metryk opisujących stan chmury obliczeniowej i dokonanie selekcji cech przy użyciu analizy istotności cech,
- ❖ analiza porównawcza różnych technik interpretacji agenta,
  - inne algorytmy interpretujące sieć neuronową agenta,
  - użycie architektur sieci neuronowej które są samowyjaśnialne, np. mechanizm uwagi



# Dziękuję za uwagę!