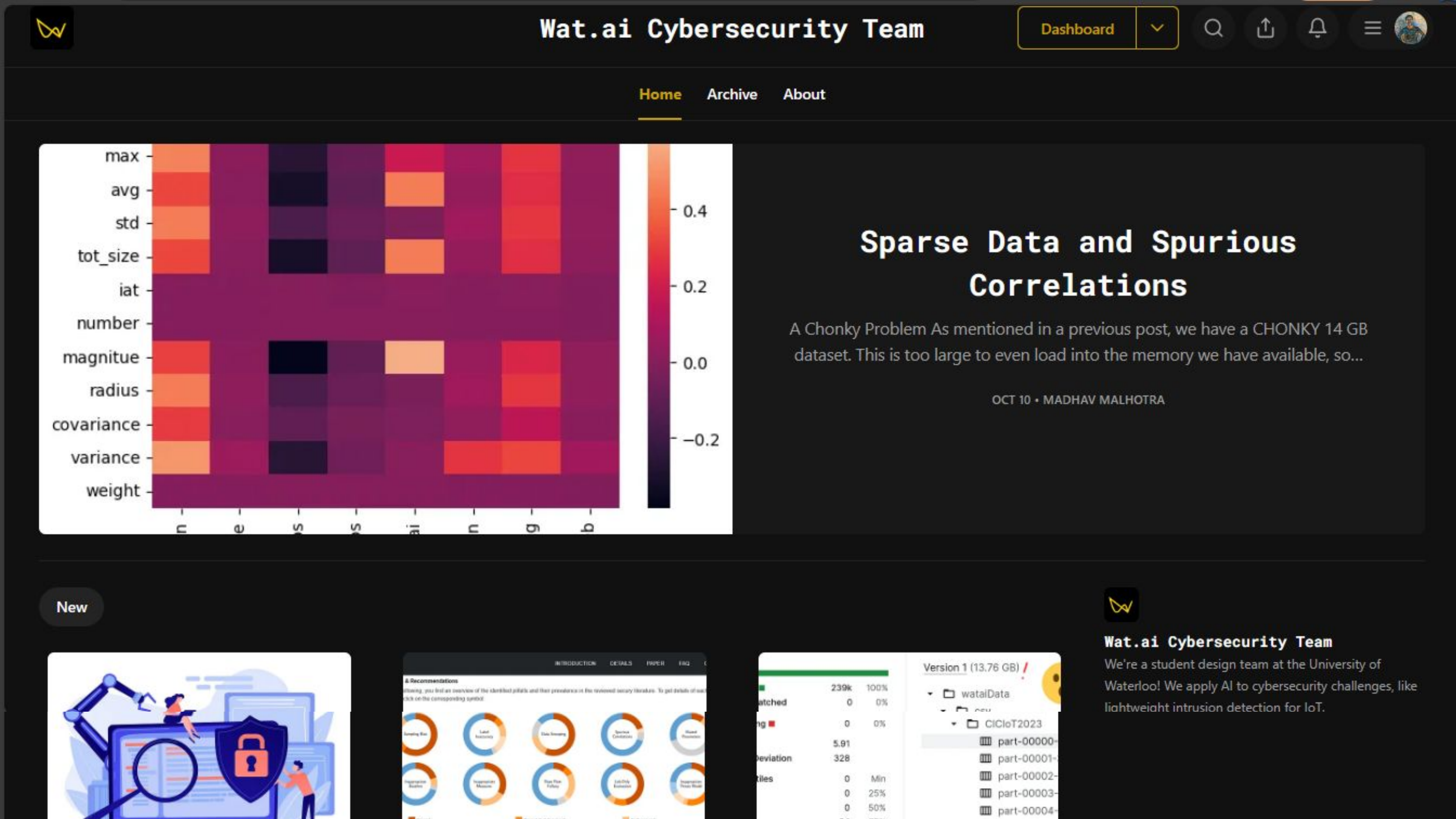
Intro to Bio-inspired AI

Wat.ai Cybersecurity Team



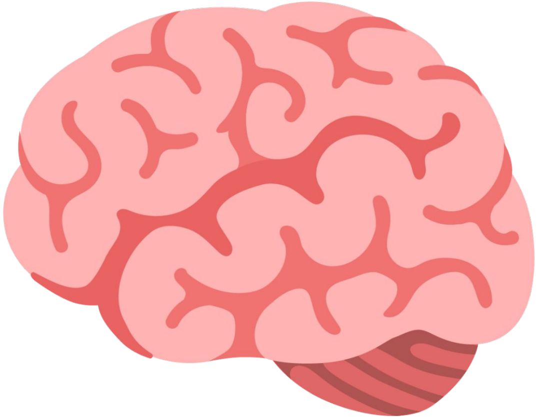
Sparse Data and Spurious Correlations

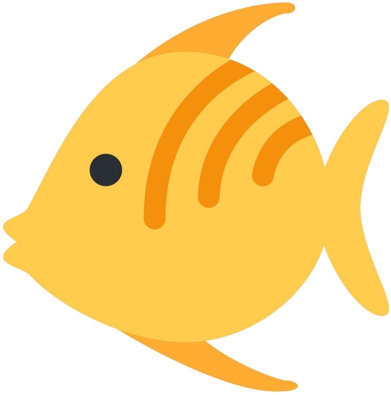
A Chonky Problem As mentioned in a previous post, we have a CHONKY 14 GB dataset. This is too large to even load into the memory we have available, so...

OCT 10 • MADHAV MALHOTRA

Wat.ai Cybersecurity Team

We're a student design team at the University of Waterloo! We apply AI to cybersecurity challenges, like lightweight intrusion detection for IoT.

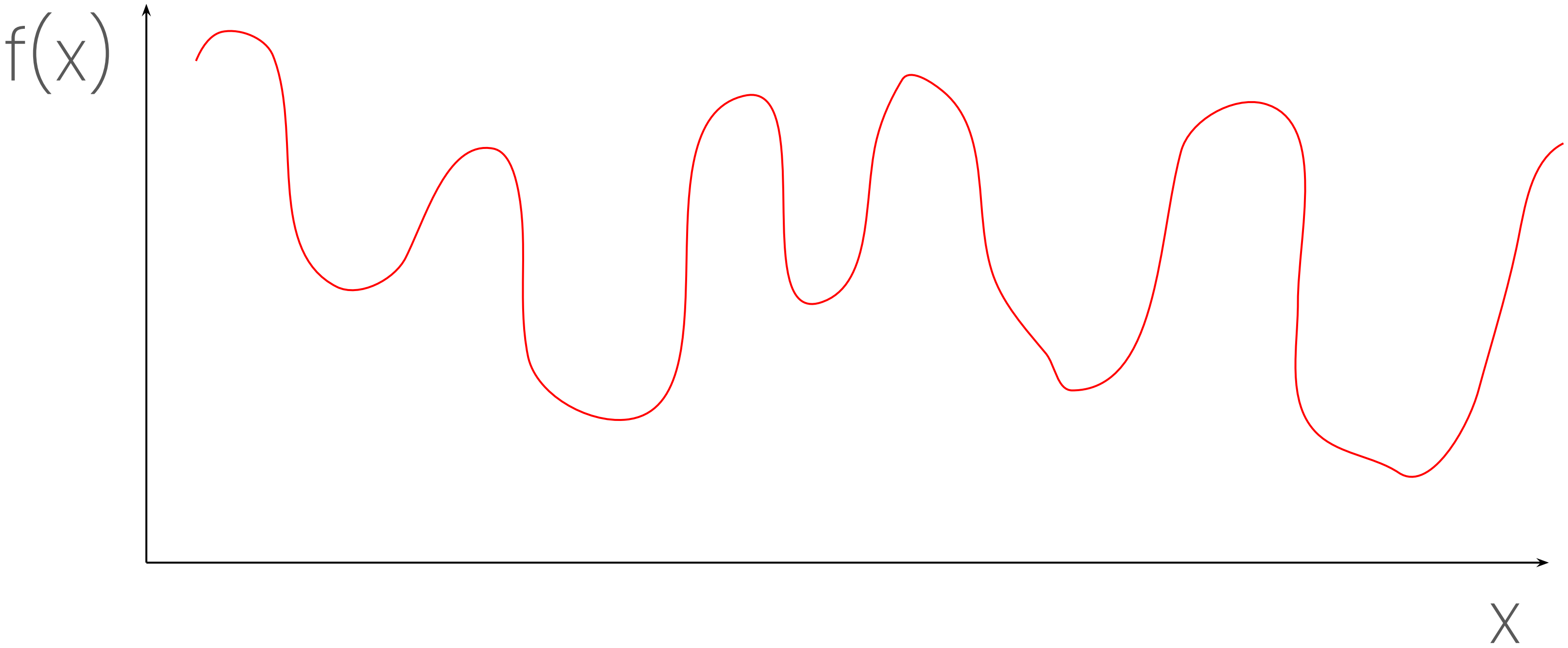




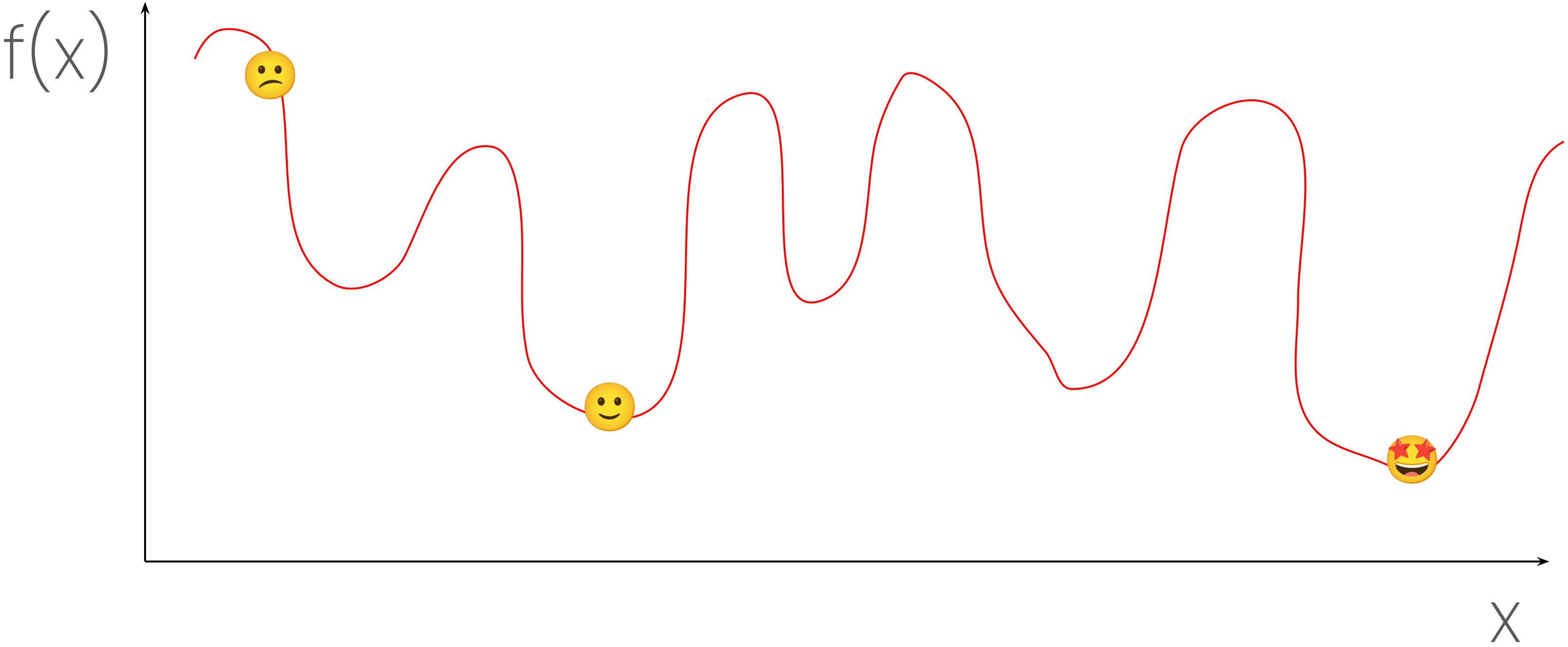
Checkpoint 1 🎉

We finished the about us narcissism

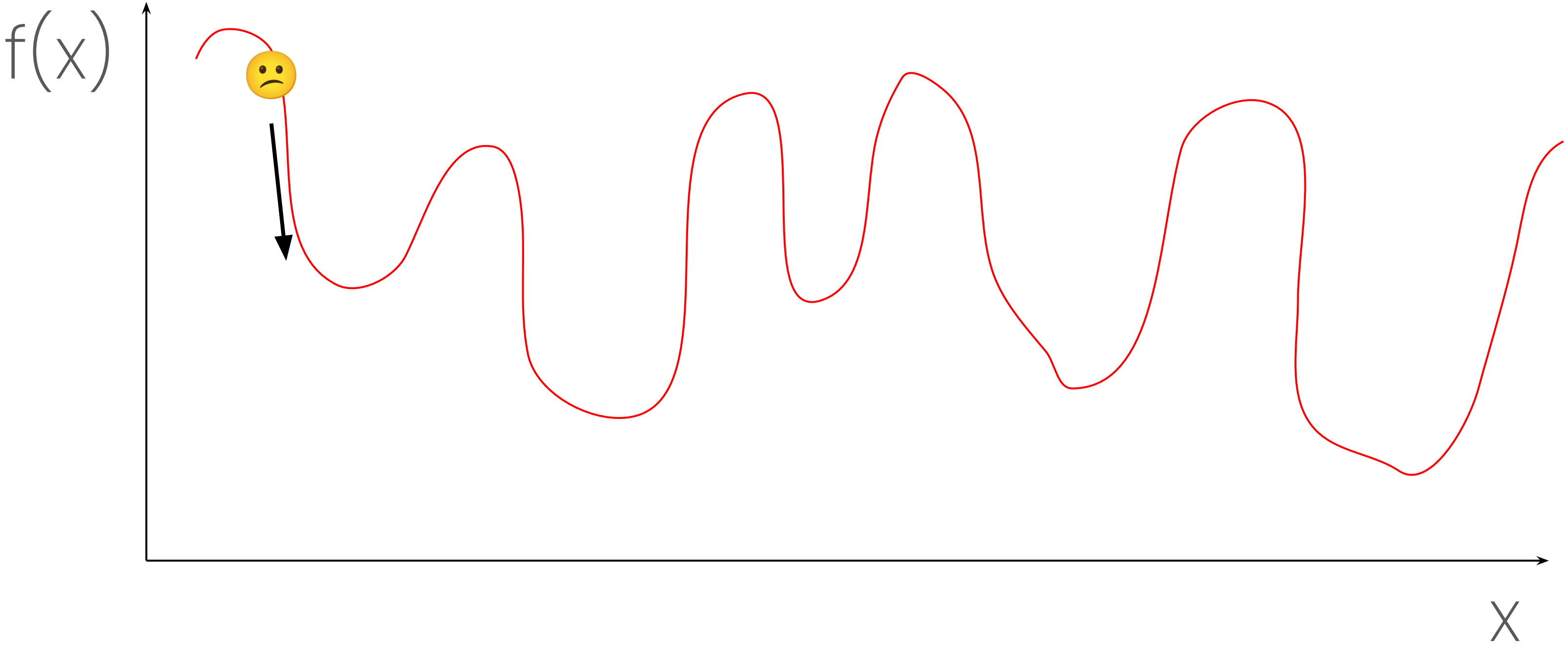
How to minimise the function?



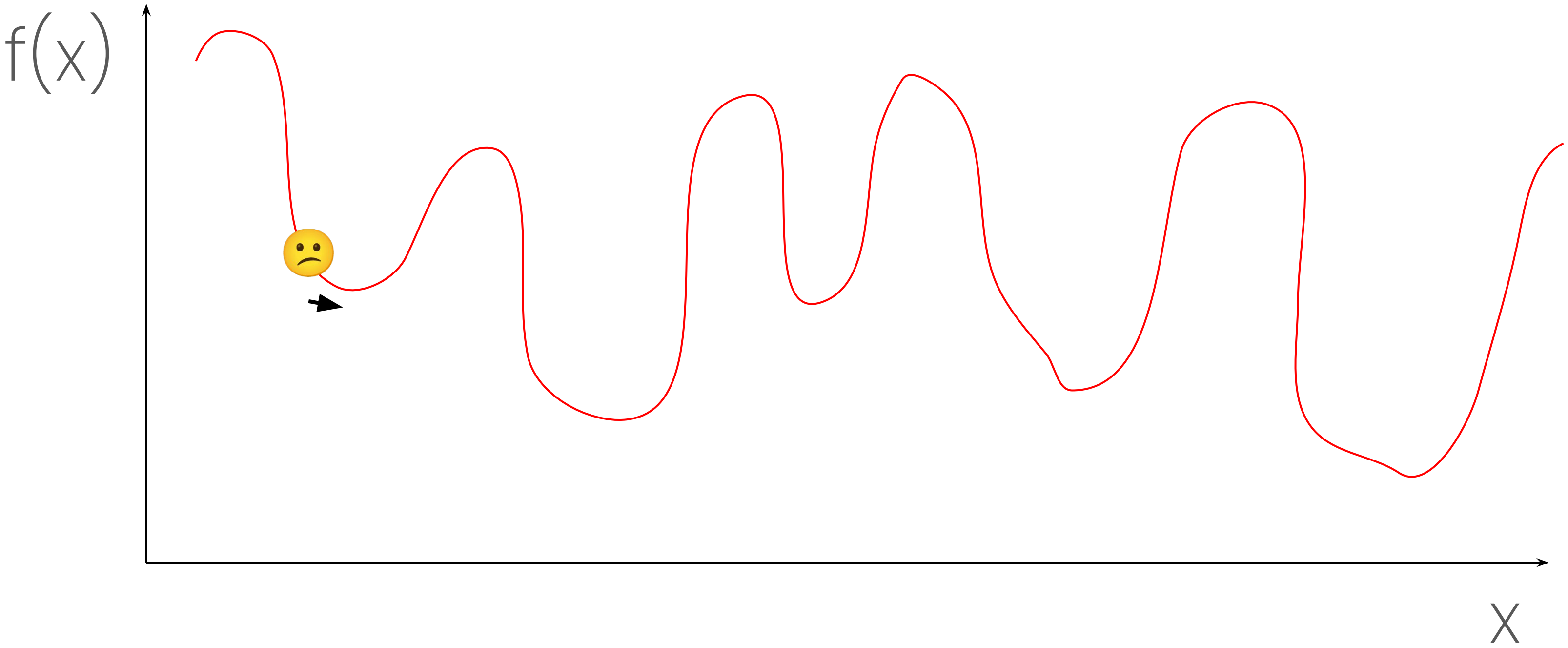
How to minimise the function?



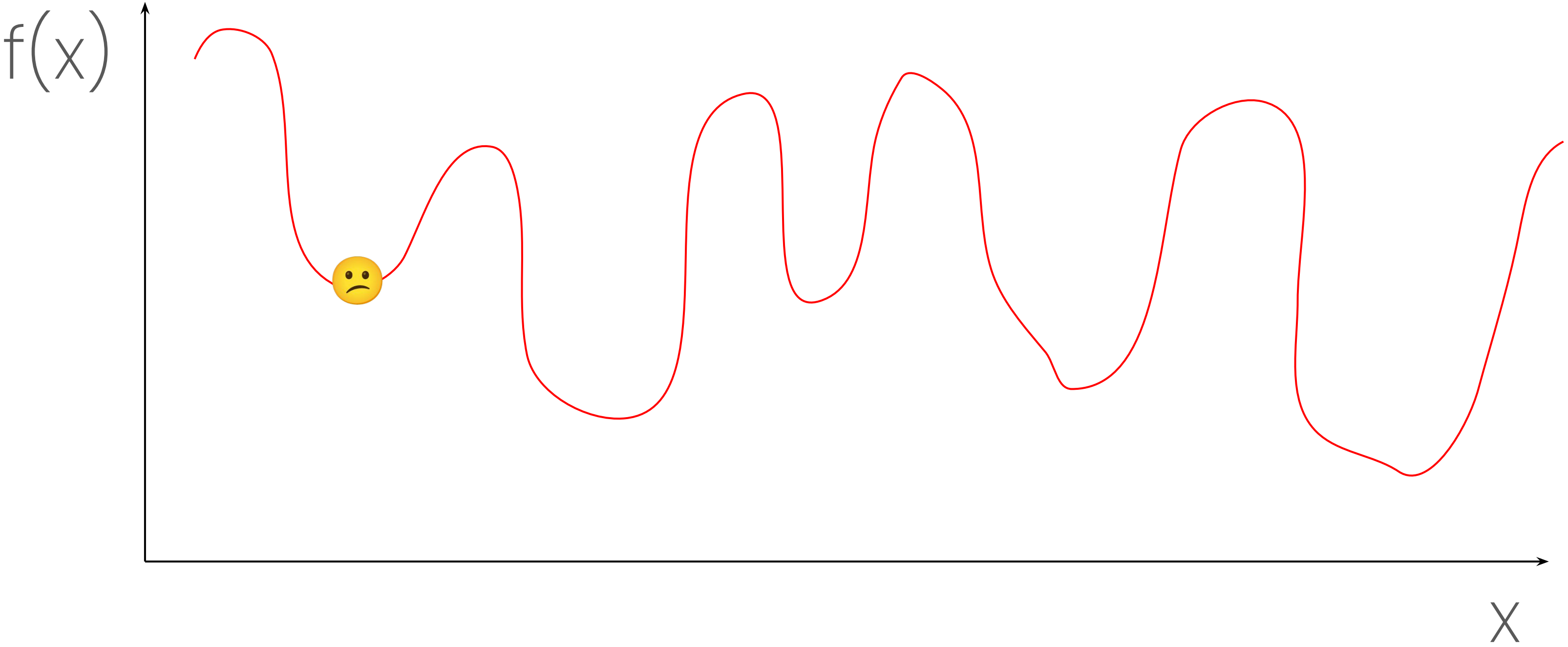
“Gradient-based” (calculus magic)



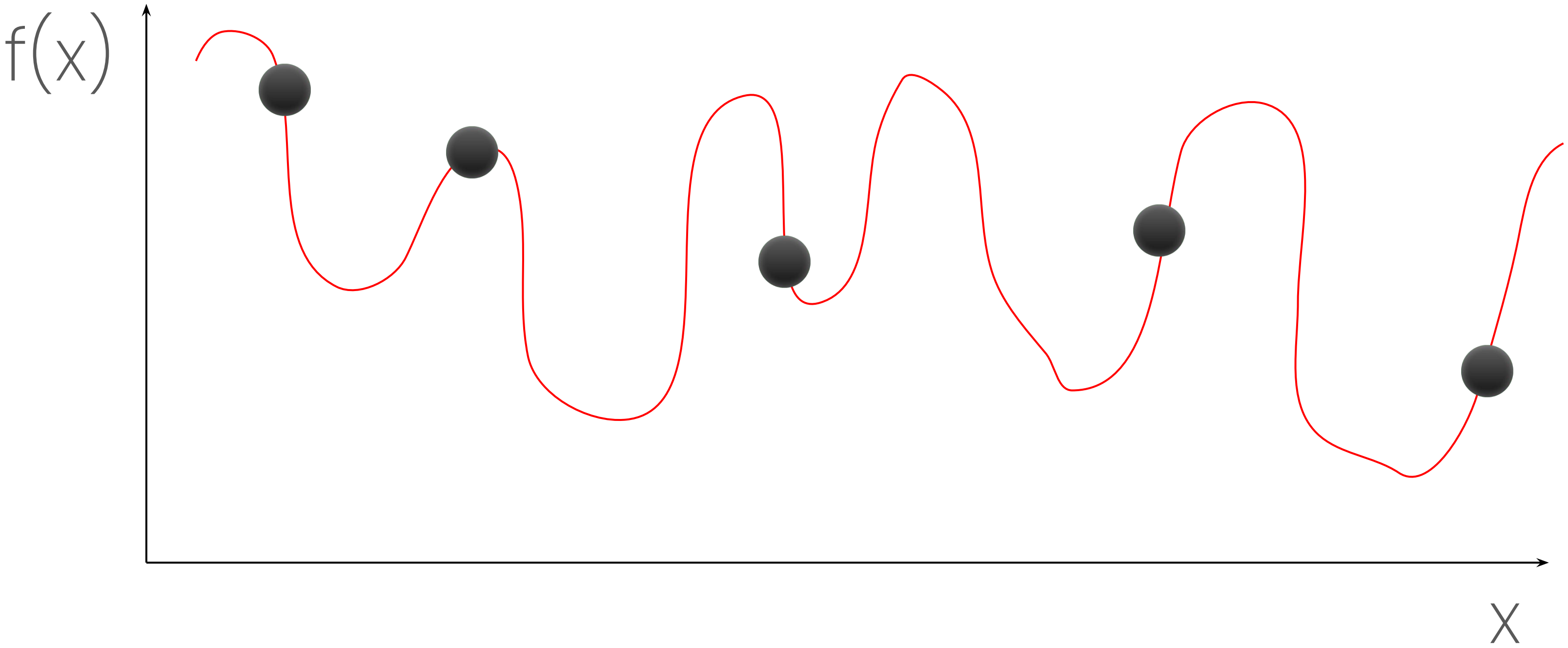
“Gradient-based” (calculus magic)



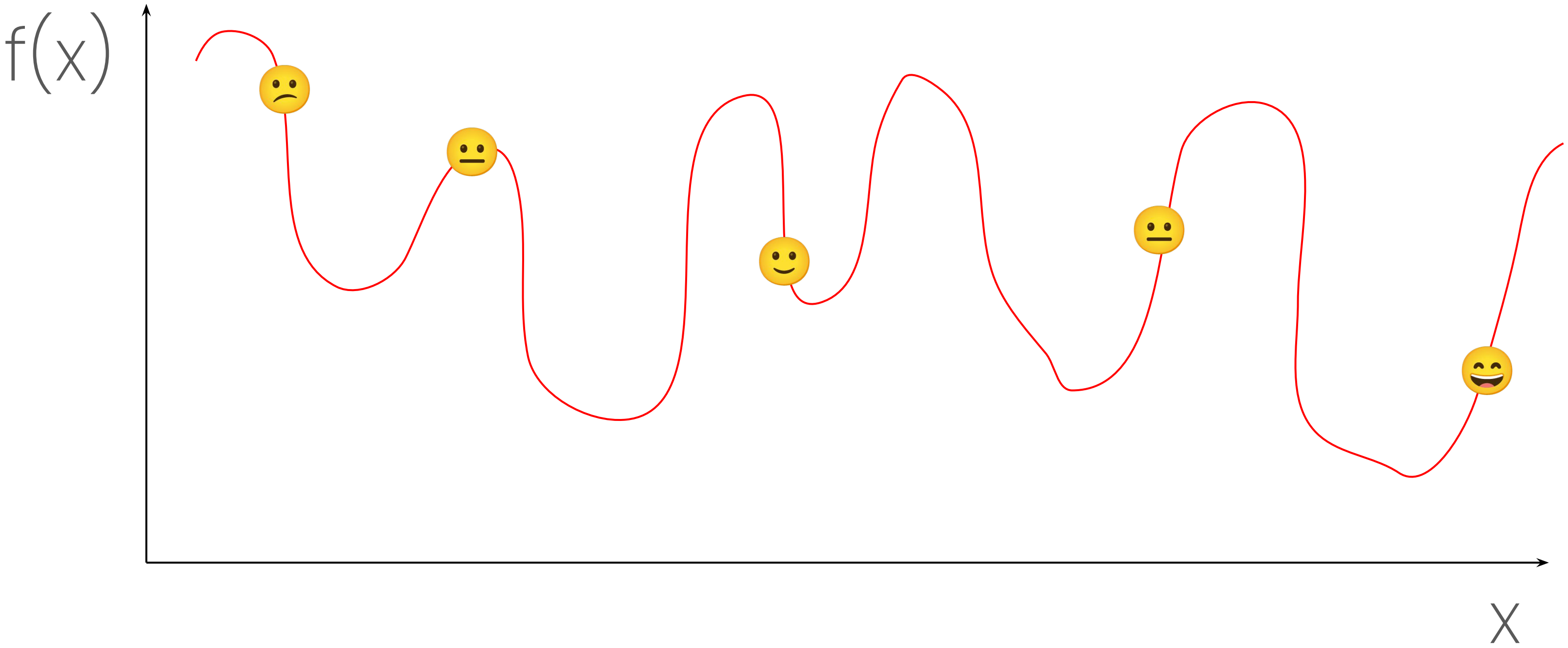
“Gradient-based” (calculus magic)



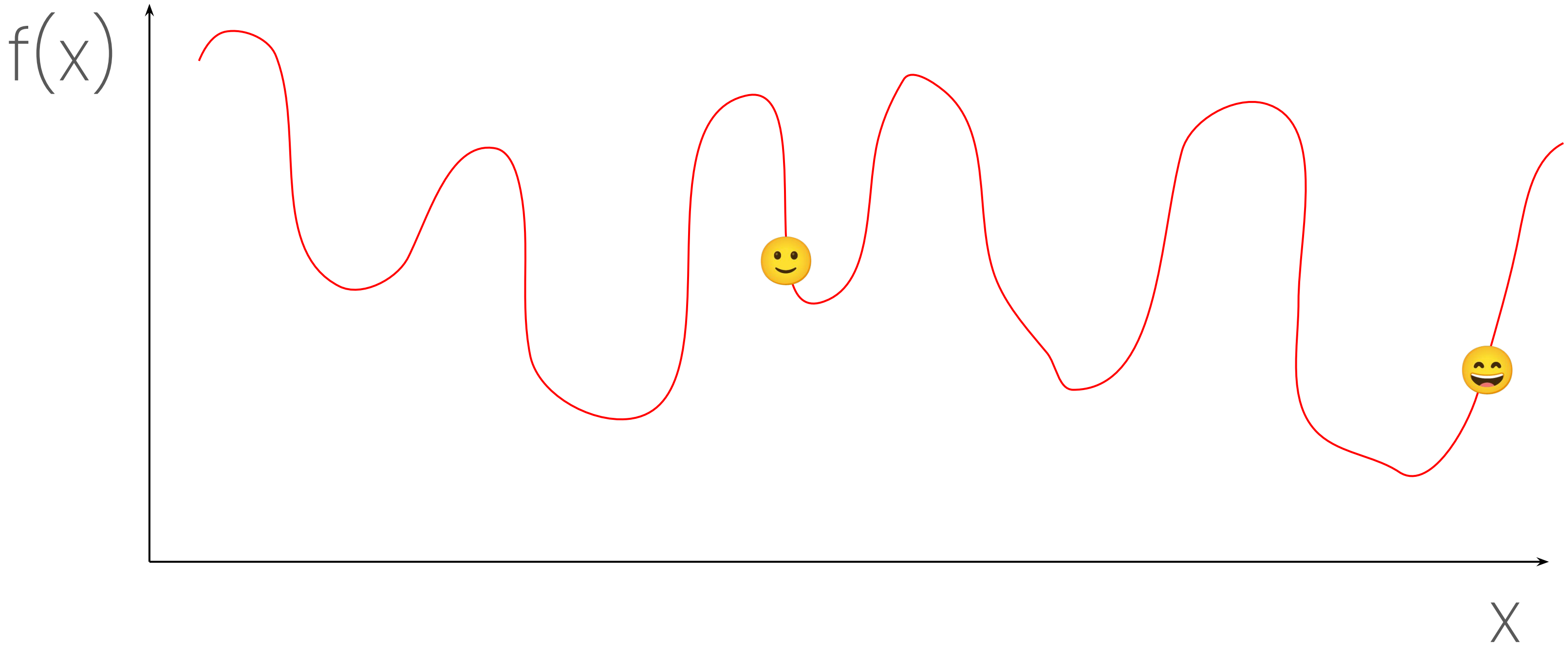
Evolution-based



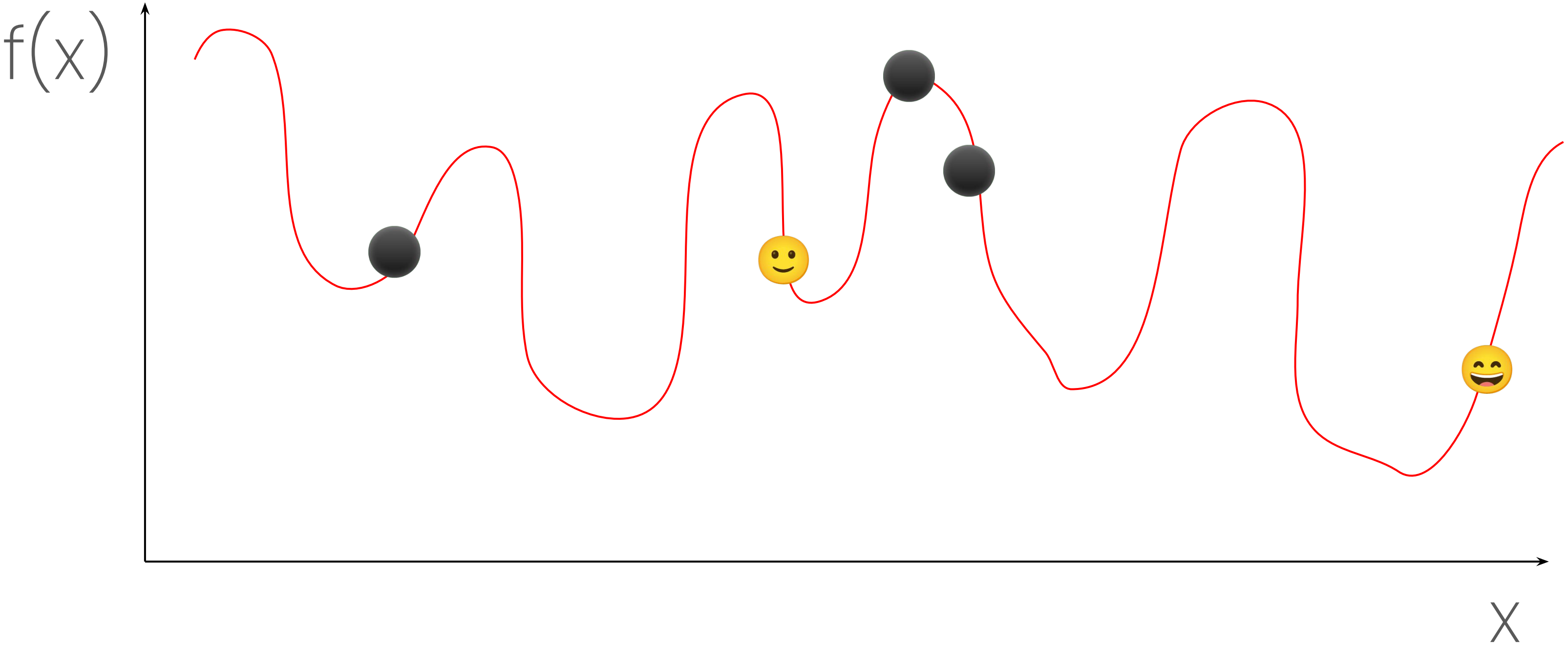
Evolution-based



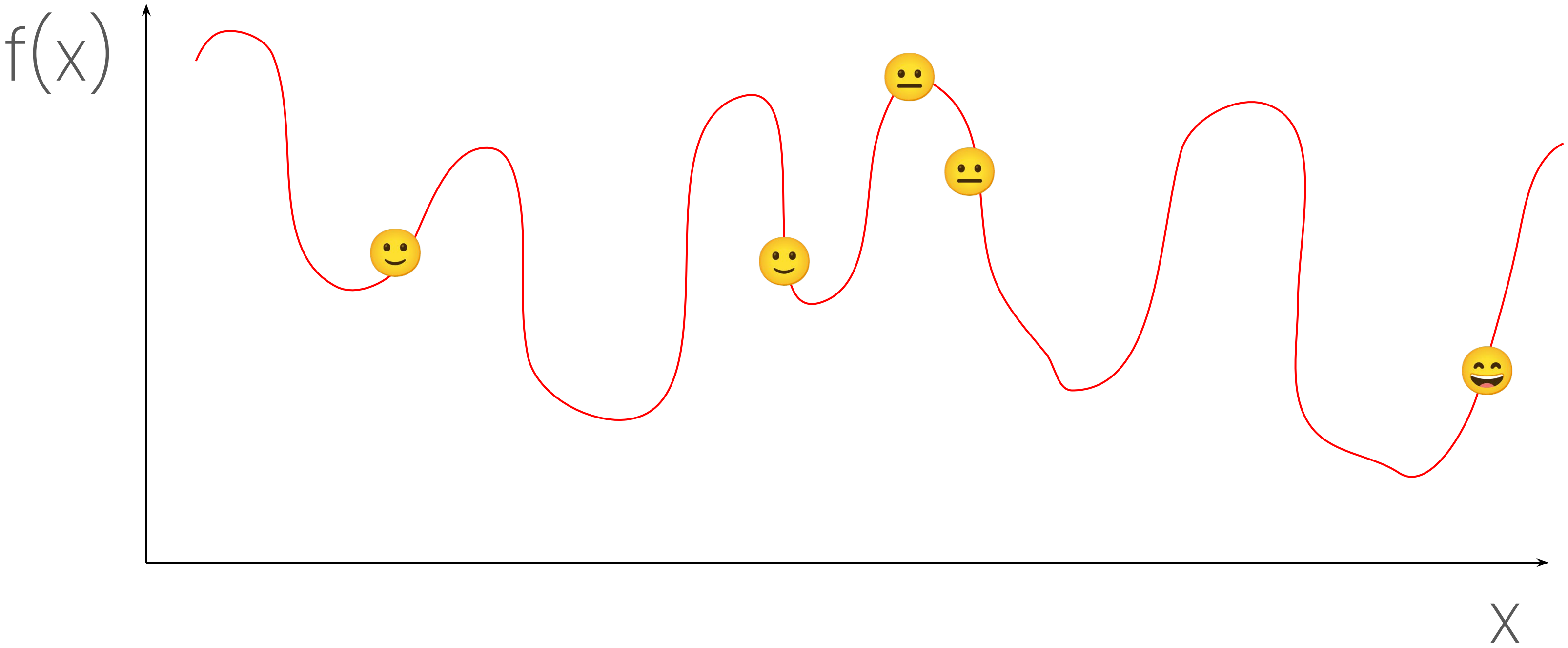
Evolution-based



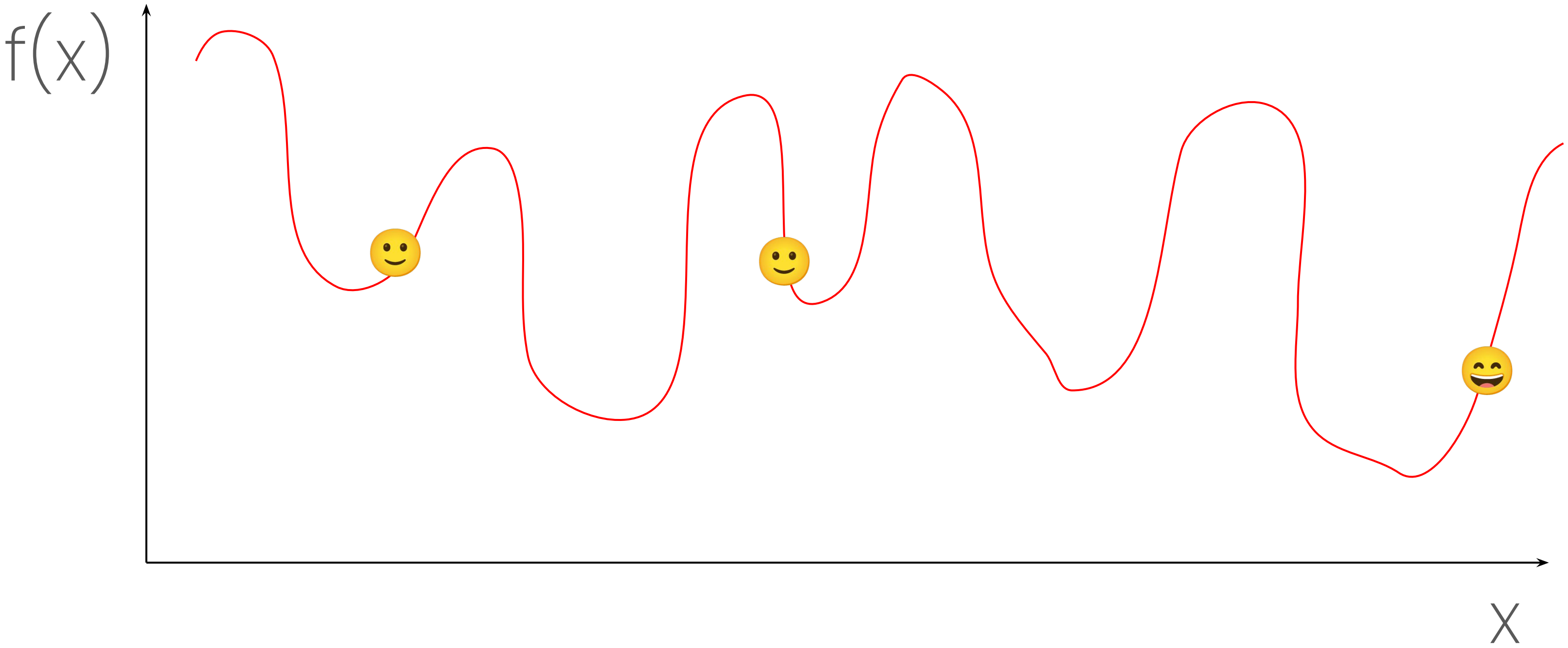
Evolution-based



Evolution-based



Evolution-based

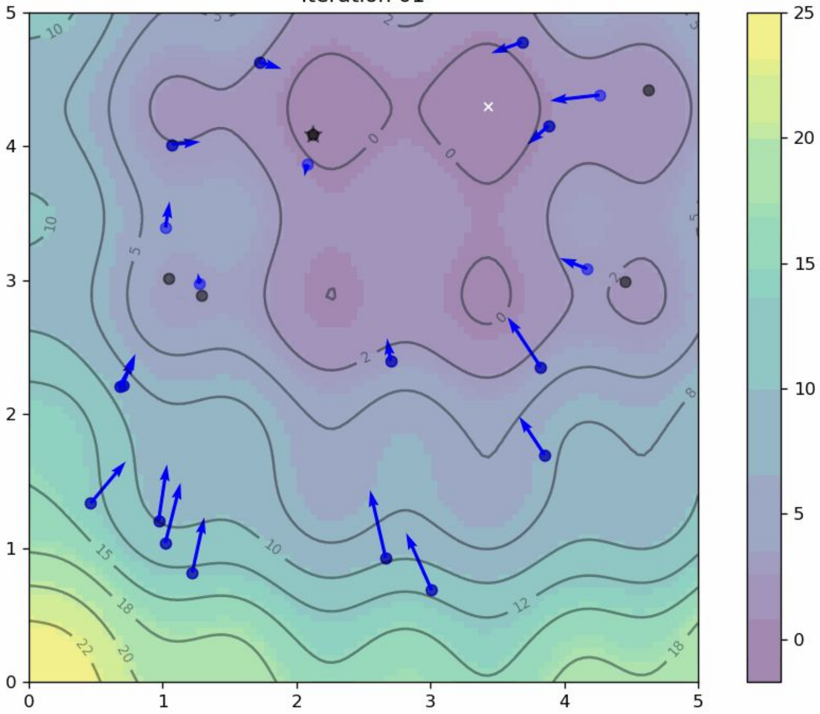


Checkpoint 2 🎉

So is this CALC 101 or what?

Monterey Bay
Aquarium

Iteration 01



Number particles: $\{1,\ 2,\ .\ .\ .\ .\ ,i,\ .\ .\ .\ .\ ,\ P\}$

Number particles: $\{1,\ 2,\ .\ .\ .\ .\ ,i,\ .\ .\ .\ .\ ,\ P\}$

Objective function: $f(x,\ y)$

Position

$$X^{i}(t) = (x^{i}(t),\ y^{i}(t))$$

Position

$$X^{i}(t) = (x^{i}(t),\ y^{i}(t))$$

Velocity

$$V^{i}(t) = (v^{i}_{x}(t),\ v^{i}_{y}(t))$$

$$X^{i}(t + 1) = X^{i}(t) + V^{i}(t + 1)$$

$$V^i(t+1) =$$

$$wV^i(t)$$

Inertia

$$V^i(t + 1) =$$

$$wV^i(t)$$

Inertia

$$+ c_1r_1(pbest^i - X^i(t))$$

Cognitive

$$V^{i}(t + 1) =$$

$$wV^{i}(t)$$

Inertia

$$+ c_{1}r_{1}(pbest^{i} - X^{i}(t))$$

Cognitive

$$+ c_{2}r_{2}(gbest - X^{i}(t))$$

Social

Position

$$X^i(t + 1) = X^i(t) + V^i(t + 1)$$

Velocity

$$V^i(t + 1) =$$

$$wV^i(t)$$

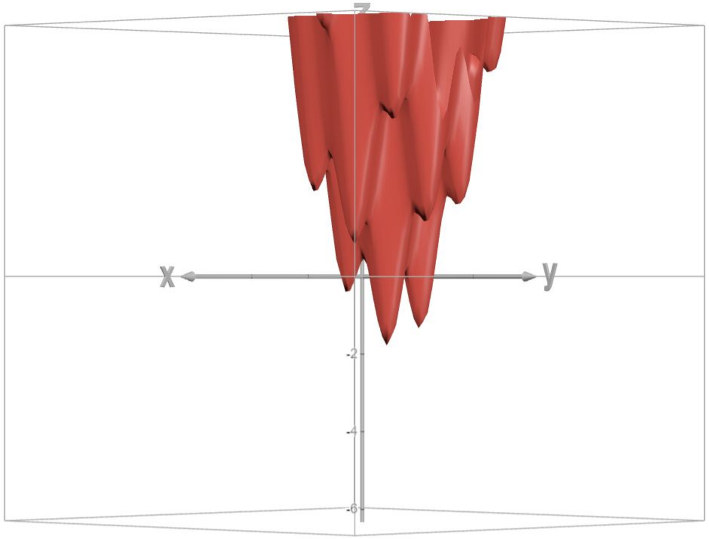
Inertia

$$+ c_1 r_1 (pbest^i - X^i(t))$$

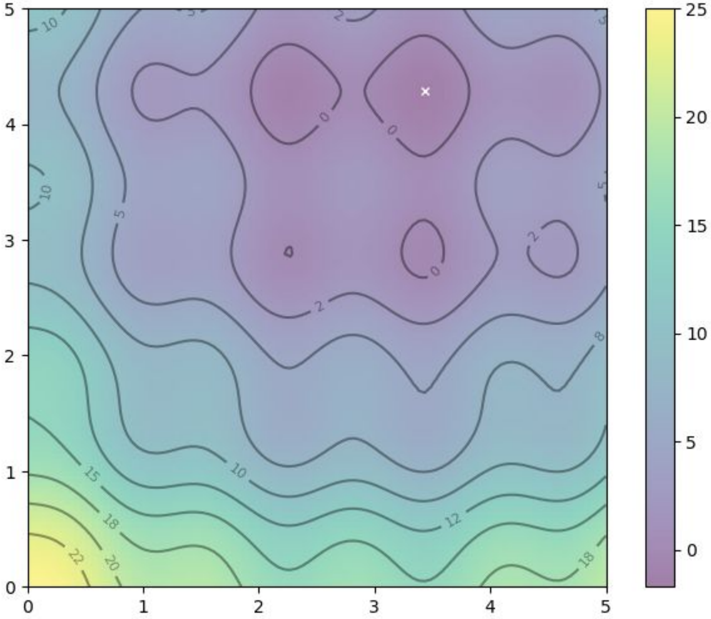
Cognitive

$$+ c_2 r_2 (gbest - X^i(t))$$

Social

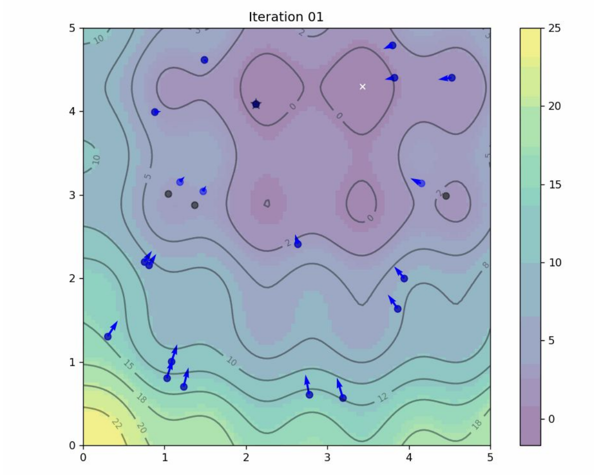


z
x
y
-2
-4
-6

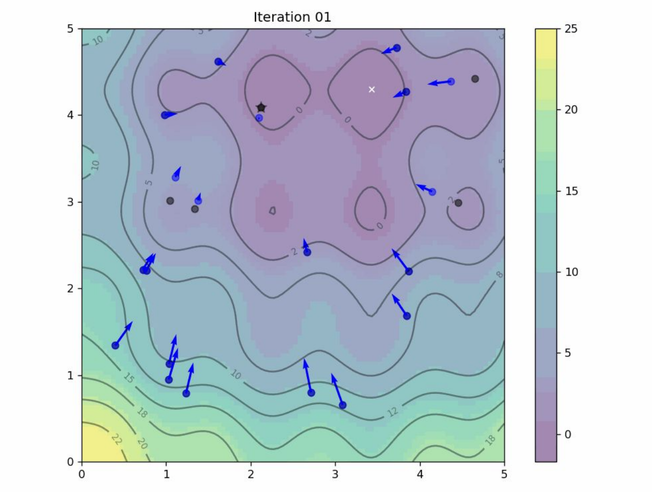


10
2
5
0
0
10
5
2
0
2
0
2
8
5
15
10
18
12
22
20
18
0
1
2
3
4
5
25
20
15
10
5
0

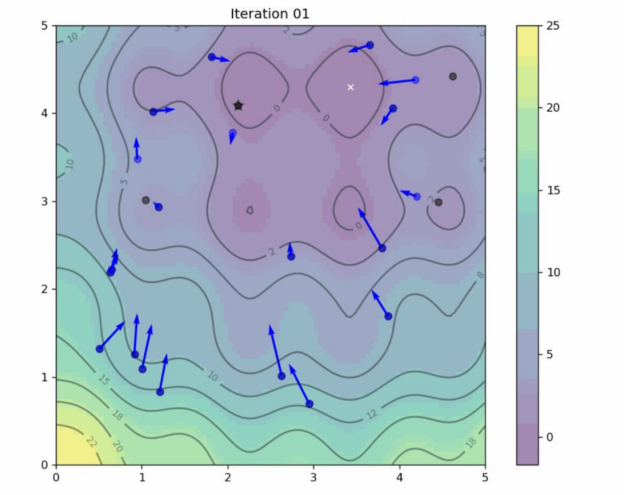
c1=c2=0.1



W=0

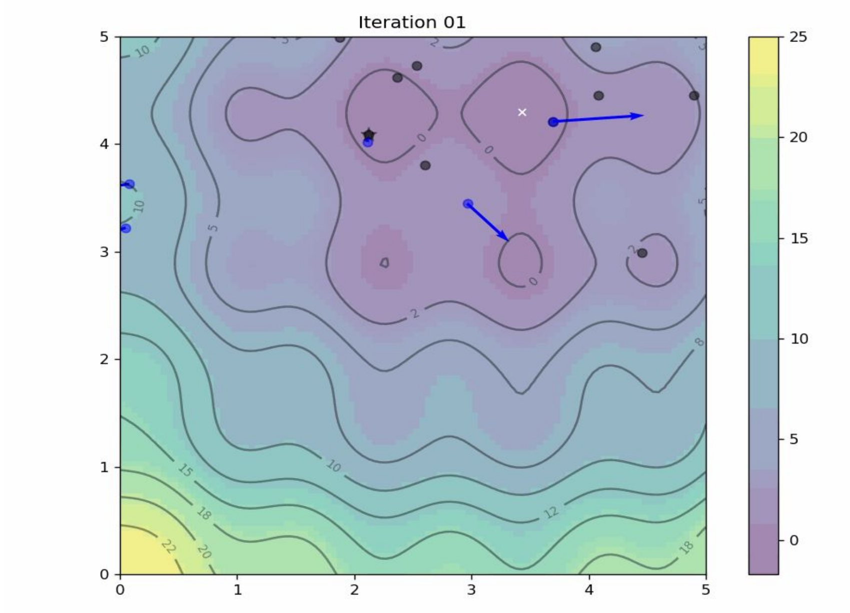


W=0.5

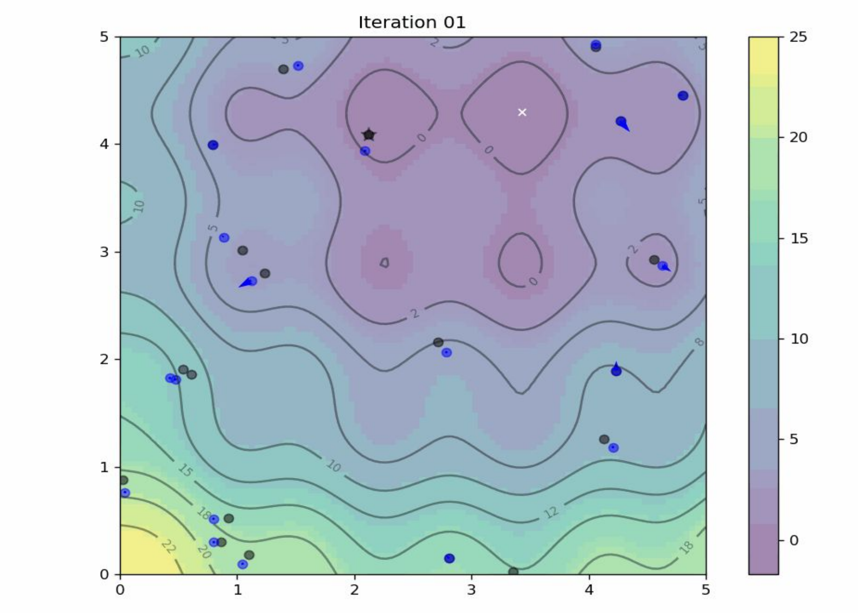


W=1

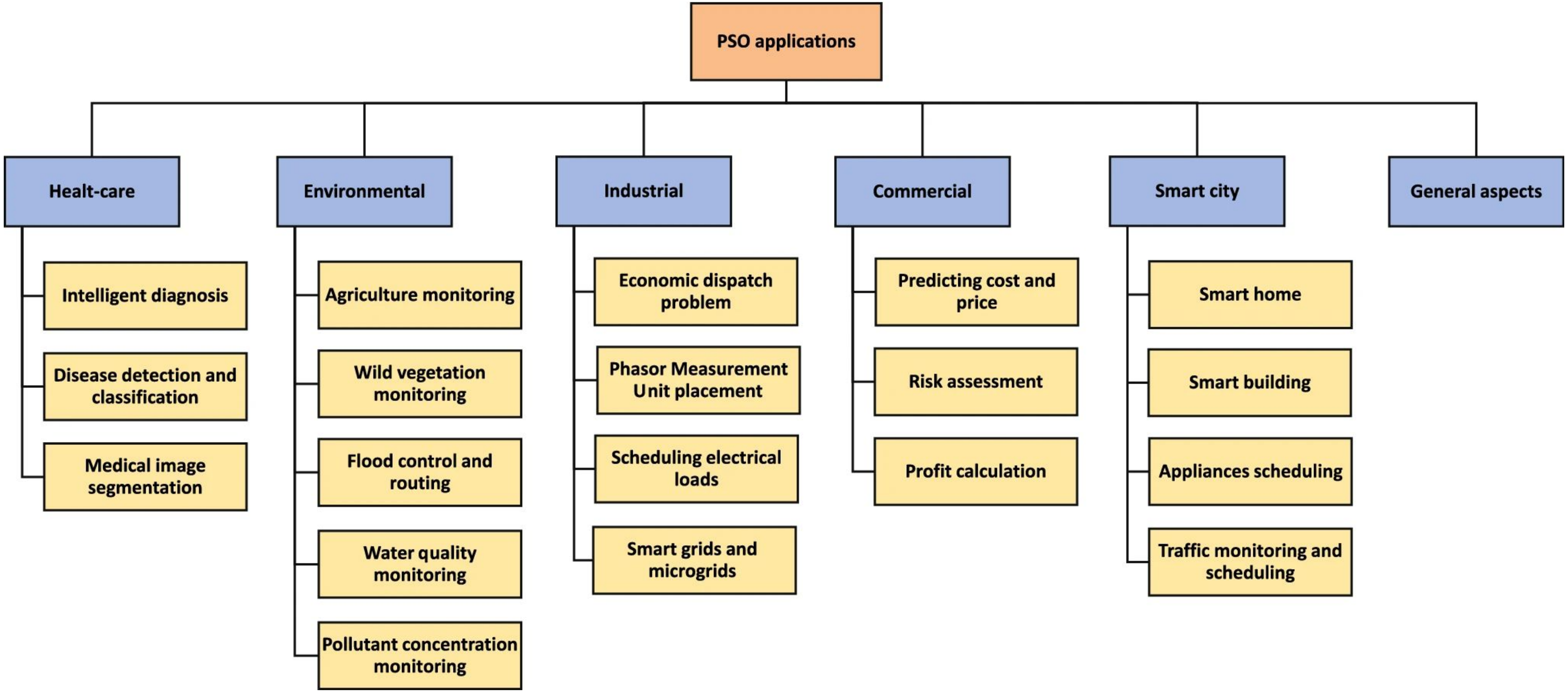
w=0.8



c1=0, c2=2



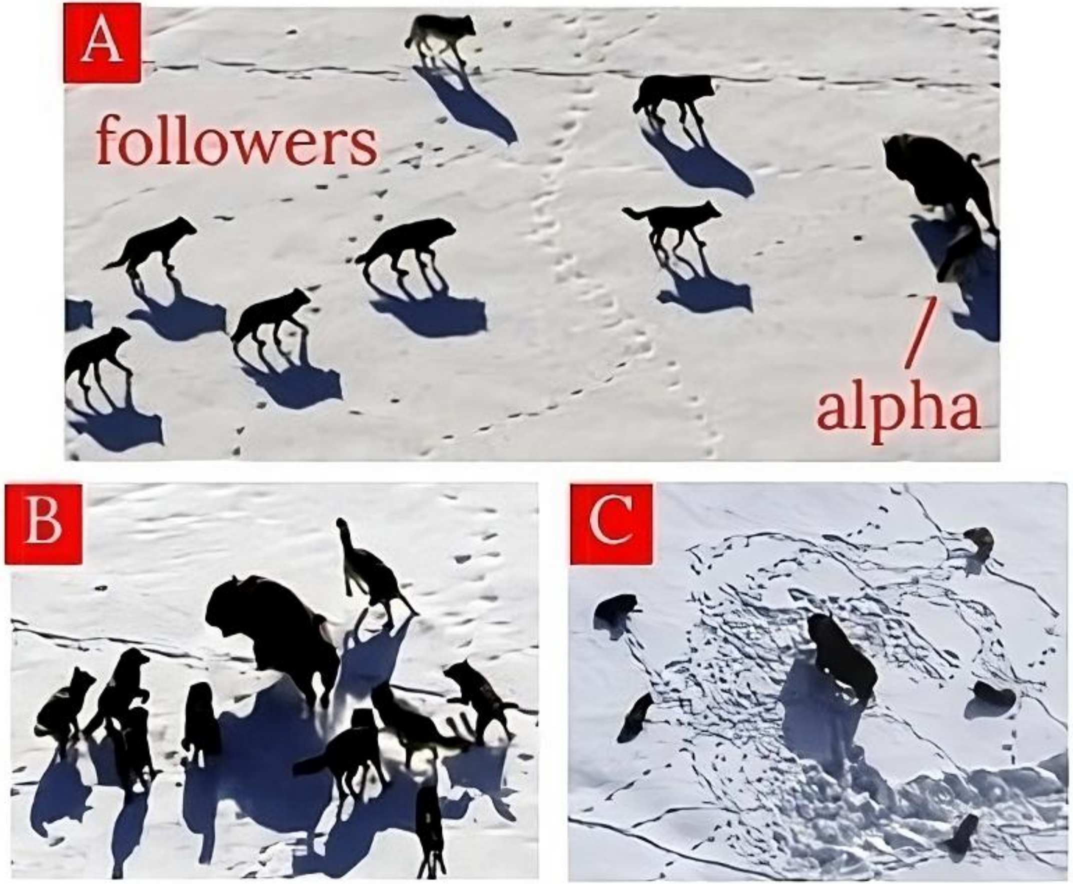
c1=2, c2=0

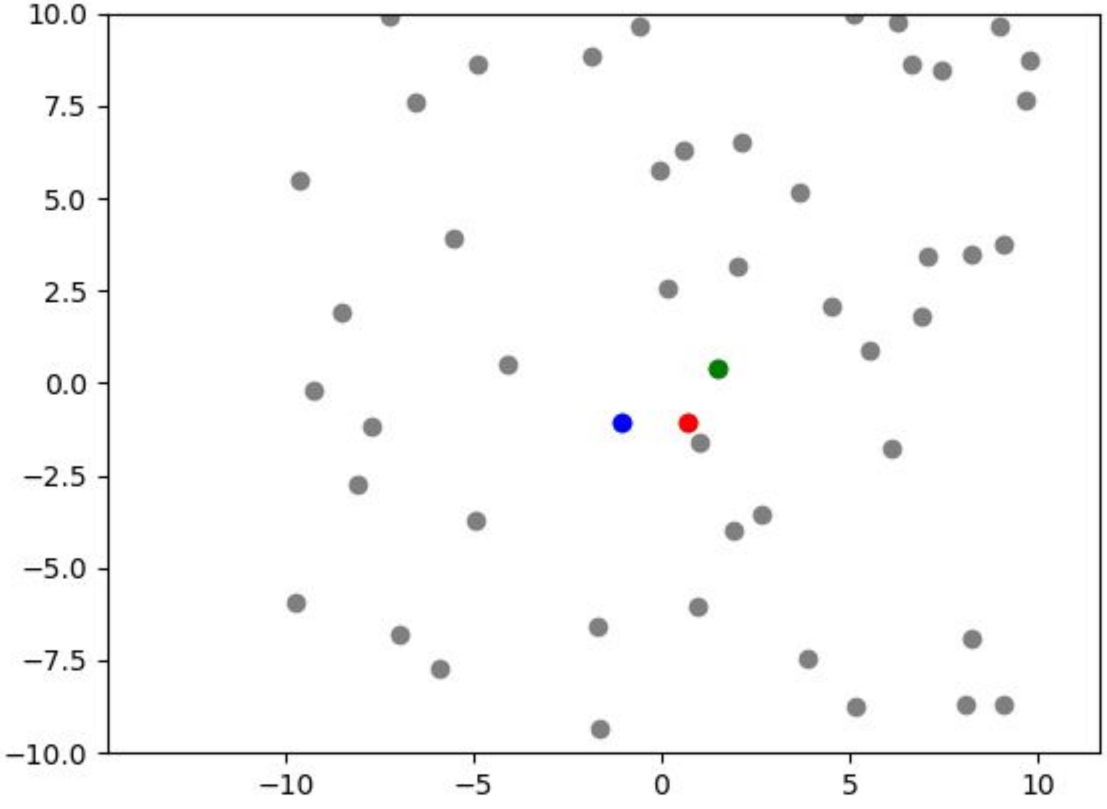

PSO applications
Healt-care
Intelligent diagnosis
Disease detection and classification
Medical image segmentation
Environmental
Agriculture monitoring
Wild vegetation monitoring
Flood control and routing
Water quality monitoring
Pollutant concentration monitoring
Industrial
Economic dispatch problem
Phasor Measurement Unit placement
Scheduling electrical loads
Smart grids and microgrids
Commercial
Predicting cost and price
Risk assessment
Profit calculation
Smart city
Smart home
Smart building
Appliances scheduling
Traffic monitoring and scheduling
General aspects

https://link.springer.com/article/10.1007/s11831-021-09694-4

Checkpoint 3 🎉

No more swimming with the fishies


A
followers
alpha
B
C



Animation by Quasar, *Stack Overflow*, 2021.

Position

Leaders

Motion

Position

Number wolves: $\{1, 2, \ldots, i, \ldots, W\}$

Position

Number wolves: $\{1, 2, \ldots, i, \ldots, W\}$

Position: $\vec{x}_i = [5, 10, 1]$

Position

Number wolves: $\{1, 2, \ldots, i, \ldots, W\}$

Position: $\vec{x}_i = [5, 10, 1]$

Specifically: $\vec{x}_i = [0, 0, 1, 0, 0, 0, 1, 1, 0, 1]$

Leaders

Selector: $D = \{l, w, h\}$. $\vec{x} = [1\ 0\ 1]$.

Leaders

Selector: $D = \{l, w, h\}$. $\vec{x} = [1\ 0\ 1]$. $S(\vec{x}) = \{l, h\}$

Leaders

Selector:

$$D = \overset{N_{tot}}{\{l, w, h\}}.\ \vec{x} = [1\ 0\ 1].\ S(\vec{x}) = \overset{N_{feat}}{\{l, h\}}$$

Leaders

Selector: $D = \overbrace{\{l, w, h\}}^{N_{tot}}.\ \vec{x} = [1\ 0\ 1].\ S(\vec{x}) = \overbrace{\{l, h\}}^{N_{feat}}$

Error: $E(S(\vec{x})) = E(\{l, h\}) = 0.23$

Leaders

Selector:

$$D = \overset{N_{tot}}{\{l, w, h\}}.\ \vec{x} = [1\ 0\ 1].\ S(\vec{x}) = \overset{N_{feat}}{\{l, h\}}$$

Error:

$$E(S(\vec{x})) = E(\{l, h\}) = 0.23$$

Output:

$$f(\vec{x}) = k \cdot E(S(\vec{x}))$$

Leaders

Selector:

$$D = \overset{N_{tot}}{\{l, w, h\}}.\ \vec{x} = [1\ 0\ 1].\ S(\vec{x}) = \overset{N_{feat}}{\{l, h\}}$$

Error:

$$E(S(\vec{x})) = E(\{l, h\}) = 0.23$$

Output:

$$f(\vec{x}) = k \cdot E(S(\vec{x})) + (1 - k) \cdot \frac{N_{feat}}{N_{tot}}$$

Motion

Following: $\vec{x}_{\alpha,i}(t+1) = \vec{x}_{\alpha}(t)$

Motion

Following:

$$\vec{x}_{\alpha,i}(t+1) = \vec{x}_{\alpha}(t) - c$$

Motion

Following:

$$\vec{x}_{\alpha,i}(t+1) = \vec{x}_{\alpha}(t) - c$$

Enclosing:

$$c(t) = 2\left(1 - \frac{t}{M}\right)$$

Motion

Following:

$$\vec{x}_{\alpha,i}(t+1) = \vec{x}_{\alpha}(t) - c(2\vec{r}_1 - 1)$$

Enclosing:

$$c(t) = 2\left(1 - \frac{t}{M}\right)$$

Motion

Following:

$$\vec{x}_{\alpha,i}(t+1) = \vec{x}_{\alpha}(t) - c(2\vec{r}_1 - 1)\left|2\vec{r}_2\vec{x}_{\alpha}(t) - \vec{x}_i(t)\right|$$

Enclosing:

$$c(t) = 2\left(1 - \frac{t}{M}\right)$$

Motion

Following:

$$\vec{x}_{\alpha,i}(t+1) = \overset{\text{Where to go}}{\vec{x}_\alpha(t)} - \overset{\text{How much error}}{c(2\vec{r}_1 - 1)}\,\overset{\text{How far to go}}{\left|2\vec{r}_2\vec{x}_\alpha(t) - \vec{x}_i(t)\right|}$$

Enclosing:

$$c(t) = 2\left(1 - \frac{t}{M}\right)$$

Motion

Following:

$$\vec{x}_{\alpha,i}(t+1) = \vec{x}_{\alpha}(t) - c(2\vec{r}_1 - 1)\left|2\vec{r}_2\vec{x}_{\alpha}(t) - \vec{x}_i(t)\right|$$

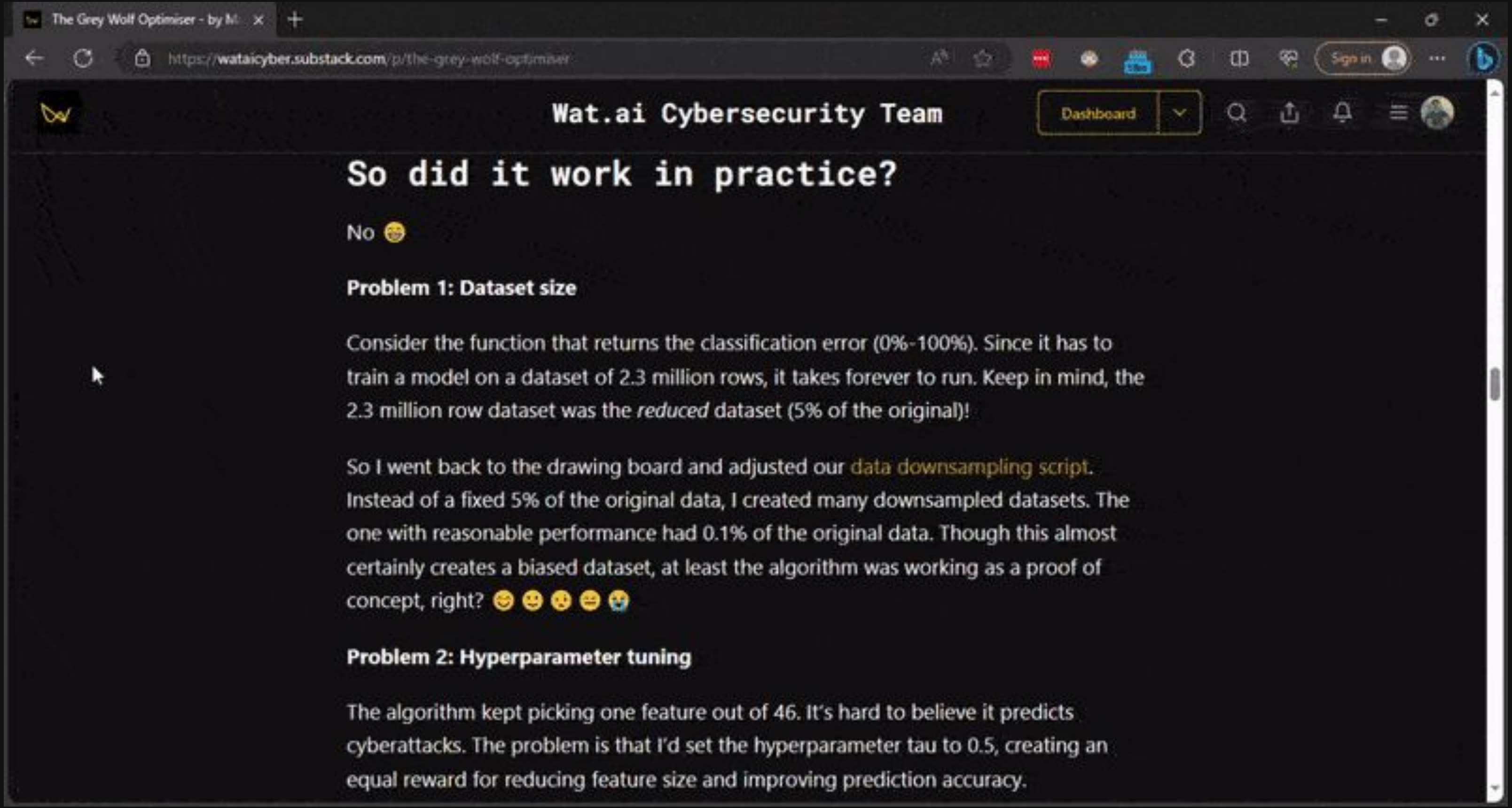
Enclosing:

$$c(t) = 2\left(1 - \frac{t}{M}\right)$$

Centering:

$$\vec{x}_i(t+1) = \frac{1}{3}\left(\vec{x}_{\alpha,i} + \vec{x}_{\beta,i} + \vec{x}_{\delta,i}\right)$$

No


The Grey Wolf Optimiser - by M
https://wataicyber.substack.com/p/the-grey-wolf-optimiser
Sign in
Wat.ai Cybersecurity Team
Dashboard

So did it work in practice?

No 😂

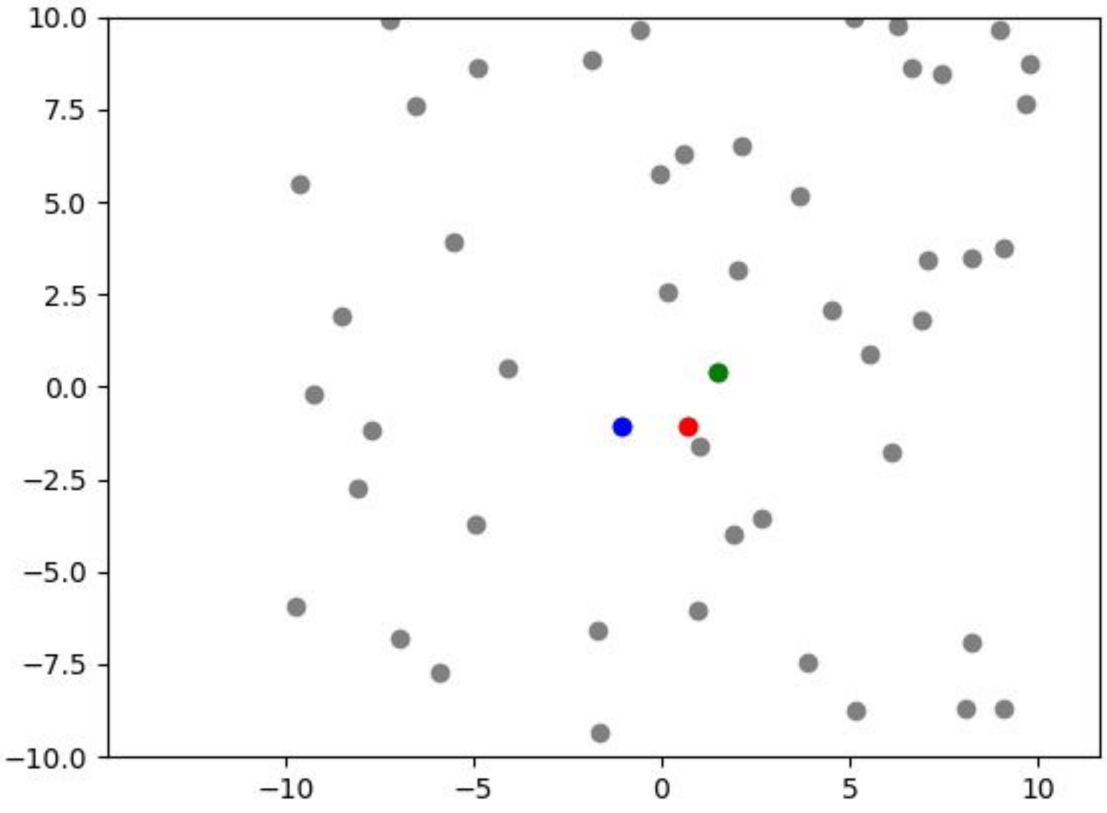
Problem 1: Dataset size

Consider the function that returns the classification error (0%-100%). Since it has to train a model on a dataset of 2.3 million rows, it takes forever to run. Keep in mind, the 2.3 million row dataset was the *reduced* dataset (5% of the original)!

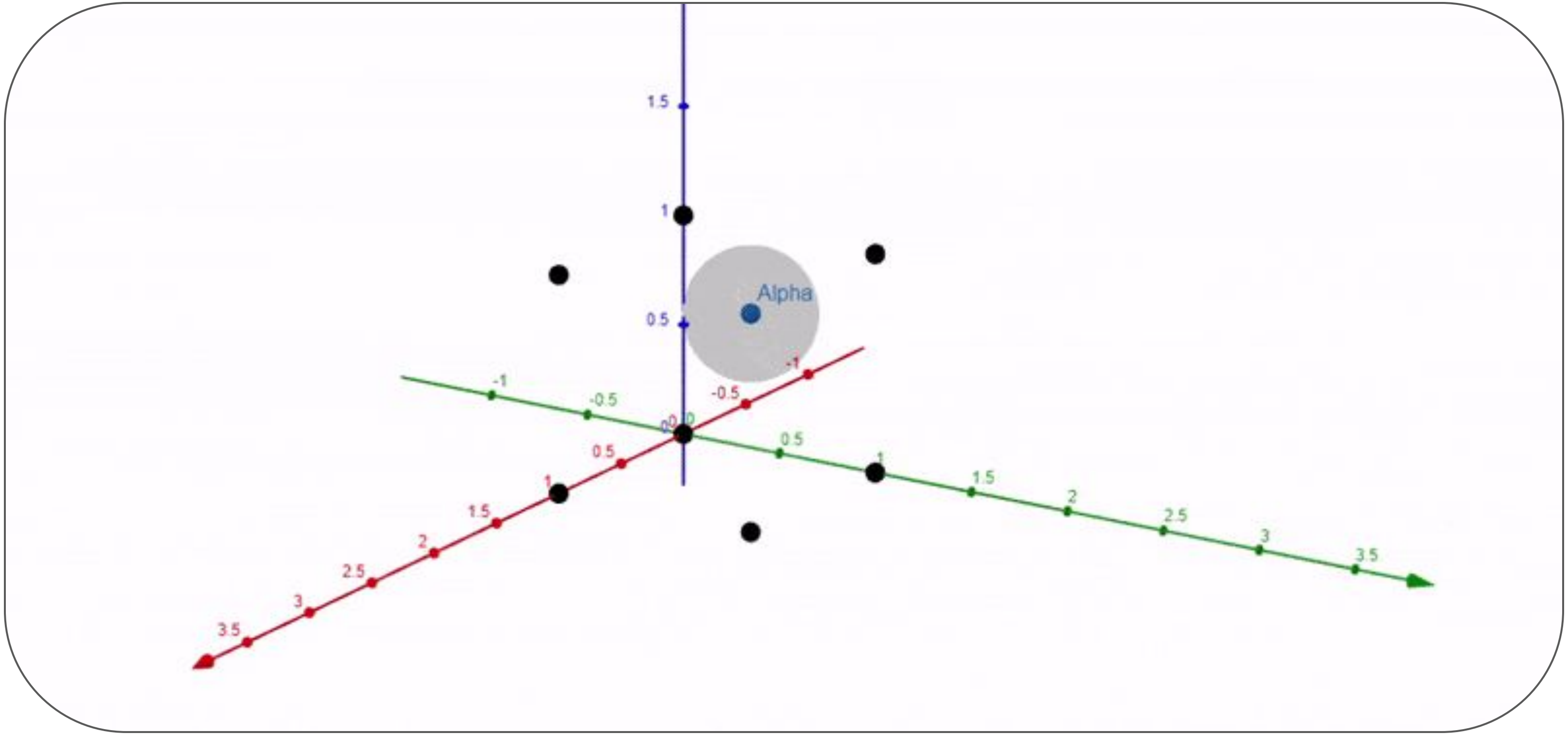
So I went back to the drawing board and adjusted our data downsampling script. Instead of a fixed 5% of the original data, I created many downsampled datasets. The one with reasonable performance had 0.1% of the original data. Though this almost certainly creates a biased dataset, at least the algorithm was working as a proof of concept, right? 😊 🙂 😟 😑 😭

Problem 2: Hyperparameter tuning

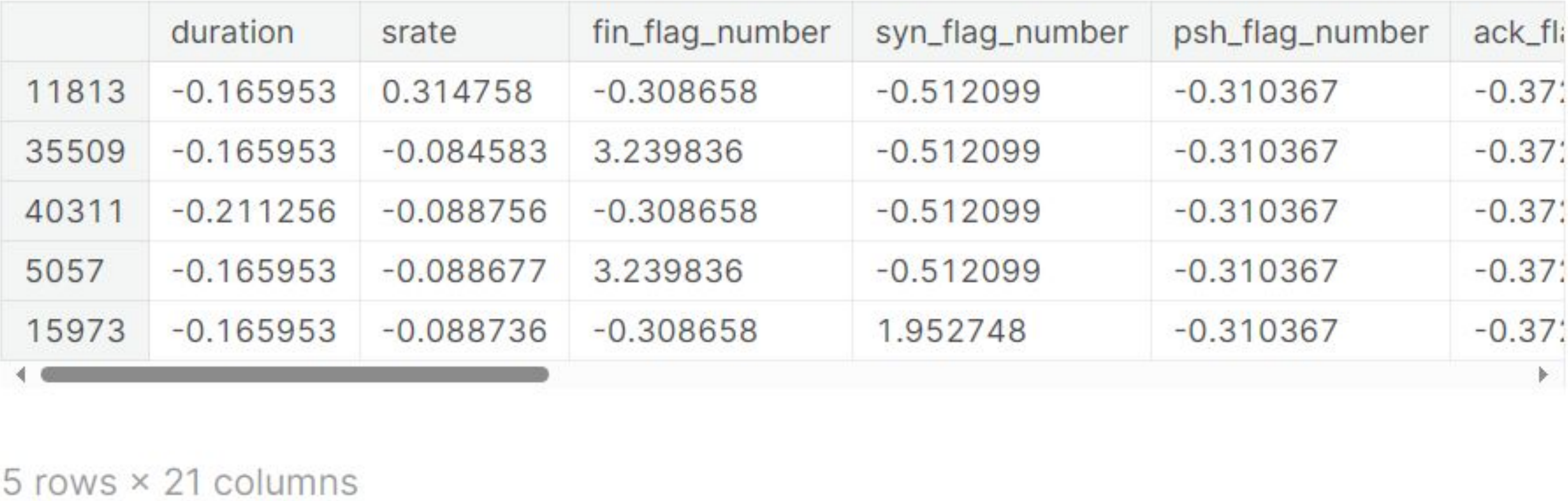
The algorithm kept picking one feature out of 46. It's hard to believe it predicts cyberattacks. The problem is that I'd set the hyperparameter tau to 0.5, creating an equal reward for reducing feature size and improving prediction accuracy.



Animation by Quasar, *Stack Overflow*, 2021.



1.5
1
0.5
Alpha
-1
-0.5
0
0.5
1
1.5
2
2.5
3
3.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3
3.5



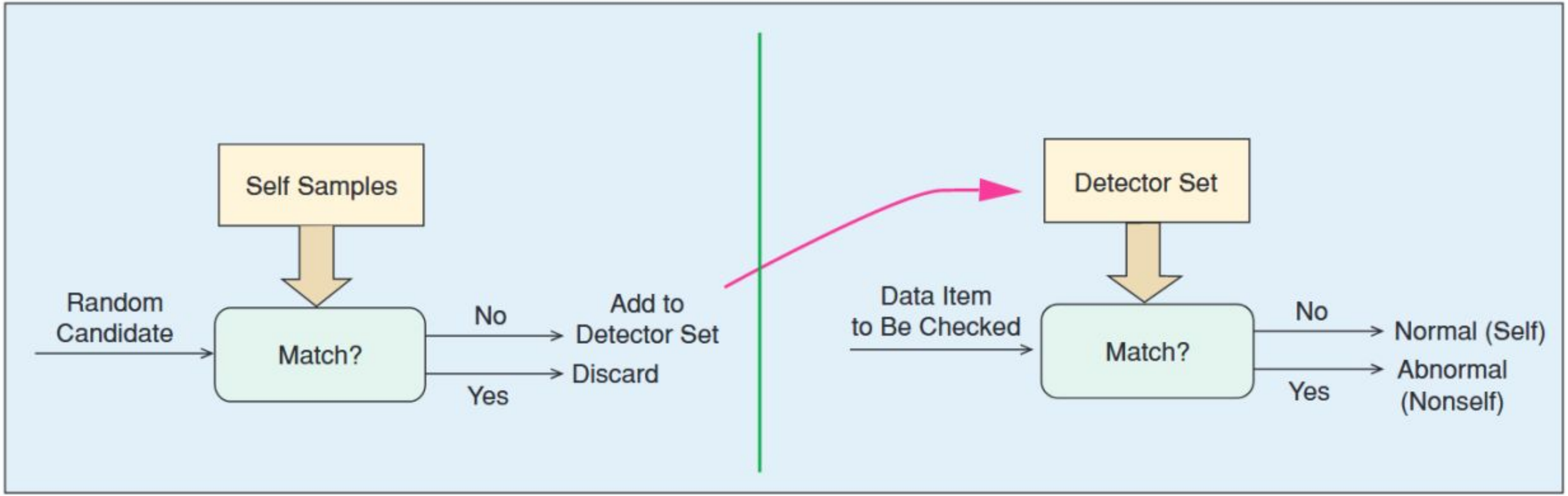
	duration	srate	fin_flag_number	syn_flag_number	psh_flag_number	ack_fl
11813	-0.165953	0.314758	-0.308658	-0.512099	-0.310367	-0.37
35509	-0.165953	-0.084583	3.239836	-0.512099	-0.310367	-0.37
40311	-0.211256	-0.088756	-0.308658	-0.512099	-0.310367	-0.37
5057	-0.165953	-0.088677	3.239836	-0.512099	-0.310367	-0.37
15973	-0.165953	-0.088736	-0.308658	1.952748	-0.310367	-0.37

5 rows × 21 columns

Checkpoint 4 🎉

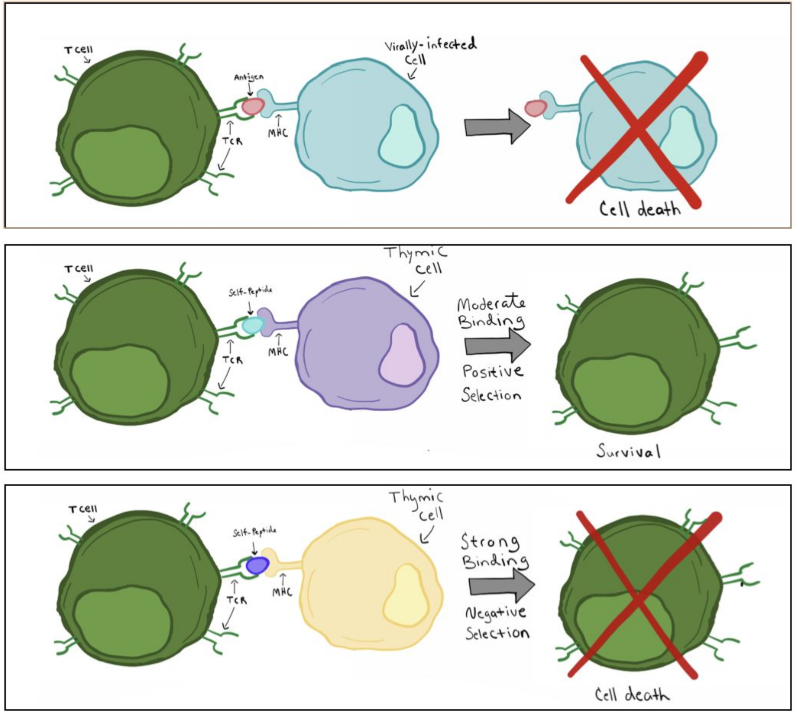
NO MORE CHECKPOINTS!!! 😤

Myesha (NSA)


Self Samples
Random Candidate
Match?
No
Add to Detector Set
Yes
Discard
Detector Set
Data Item to Be Checked
Match?
No
Normal (Self)
Yes
Abnormal (Nonself)

Myesha (NSA)

Myesha (NSA)


T cell
Antigen
TCR
MHC
Virally-infected Cell
Cell death
T cell
Self-Peptide
TCR
MHC
Thymic cell
Moderate Binding
Positive Selection
Survival
T cell
Self-Peptide
TCR
MHC
Thymic cell
Strong Binding
Negative Selection
Cell death

Myesha (NSA)

Math slides

Myesha (NSA)

Demo
(code screenshot,
graphs)

Myesha (NSA)

What are some applications of NSA?

Checkpoint 3 🎉

Begone with itsy-bitsy microbe stuff