Emeka Ezike, Dominic Garrity, and Victor Goncalves

Professor Stephanie Gil

COMPSCI 286

23 March 2022

Programming Assignment 3

Group Information

Group number: 6

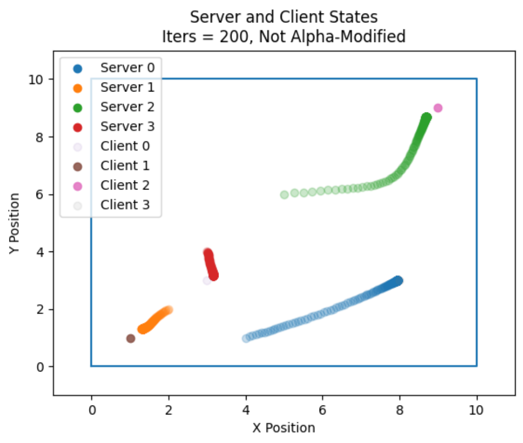
Group members: Dominic Garrity (HUID: 61367713), Victor Goncalves (HUID: 11482961), and Emeka Ezike (HUID: 11464432)

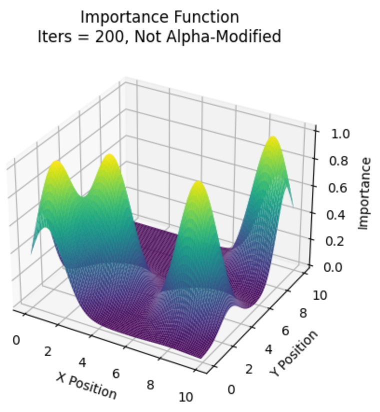
Problem 1

Note: In coverage-spoof.py, we created two variables, "part" and "num_iter," for easier evaluation. You can set "part" to the part of #1 that you wish to evaluate and "num_iters" to the number of iterations for which you wish to run its corresponding simulation. For every part of this problem, we ran its simulation for 200 time steps; used alpha = -10; and let the importance function $\rho_i(q) = exp(-0.5(q - p_i)^T(q - p_i))$*, as Gil and her colleagues did in section 9.3 of "Guaranteeing Spoof-Resilient Multi-Robot Networks." We also plotted the importance function to verify it. Alphas used to modify this importance function were sampled from a normal distribution whose mean corresponded to whether each client was a spoofer and whose standard deviation was 0.10.*

A. No discussion is necessary, but it may be worth noting that we added an “importance” attribute to the Environment class. We use this to define the “value” that is passed to mix_func() in update_gradient().

B. As shown in Figure 1(a), servers in our simulation tended toward both legitimate and spoofed clients. This is because, as shown in Figure 1(b), the environment's importance function was not alpha-modified, so–like Gil and her colleagues demonstrated–there was no defense against spoofing. Clients’ states, regardless of their legitimacy, corresponded to peaks in the importance function and therefore attracted servers that were executing the Schwager coverage algorithm.



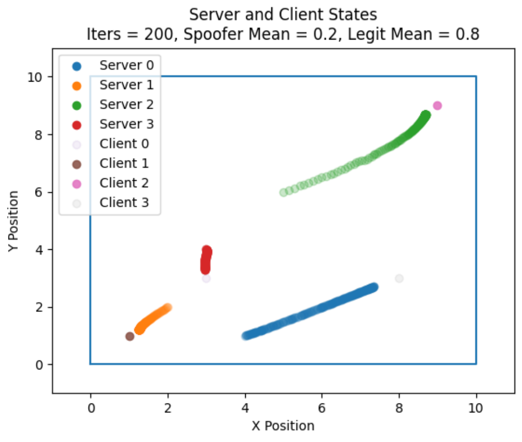


(a) *(b)*

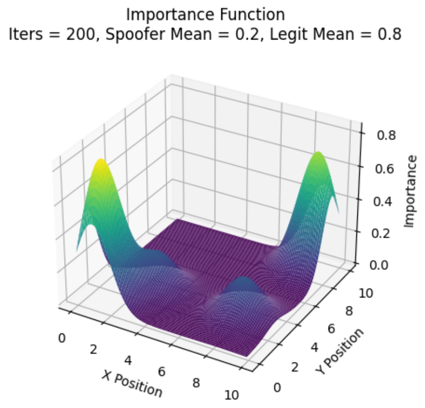
Figure 1: We ran a server-client simulation using an importance function that was not alpha-modified.

C. These values may be good because they result in higher importance around each legitimate client than around each spoofed client. More specifically, an $E(\alpha_i) = 0.2$ for spoofed clients is advantageous because α_i is a confidence metric for the i^{th} client that should be close to 0 when that client is spoofed. Likewise, an $E(\alpha_i) = 0.8$ for legitimate clients is advantageous because α_i is a confidence metric for the i^{th} client that should be close to 1 when that client is legitimate. Furthermore, these expected values sum to 1 and this could be used as a property of the importance function to facilitate analysis.

D. As described in part C and shown in Figure 2(b), importance is higher around legitimate clients than around spoofed clients and, consequently, servers tend more toward legitimate clients than they did in part B. The servers do not behave optimally, however; although Server 2 was less influenced by the spoofed Client 3, Server 0 still tended directly toward it and Server 3 still tended directly toward Client 0. This may be because the confidence metrics of spoofed clients, expected to be 0.2, are not close enough to 0 and those of legitimate clients, expected to be 0.8, are not close enough to 1.



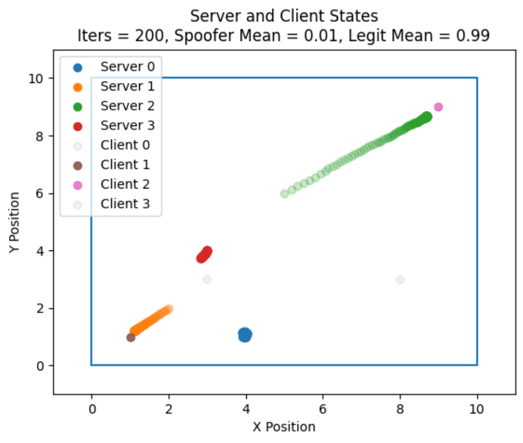
(a)

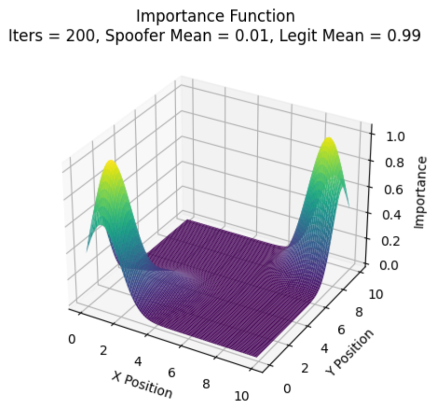


(b)

Figure 2: We ran a server-client simulation using an alpha-modified importance function. Alpha values were sampled from normal distributions whose parameters are listed in the plot subtitles.

E(i)(1). As shown in Figure 3, spoofer clients sample from a (normal) distribution with $E(\alpha_i) = 0.01$, while legitimate ones sample from a (normal) distribution with $E(\alpha_i) = 0.99$. As a result, Servers 1 and 2 tend toward the legitimate Clients 1 and 2. Meanwhile, Servers 0 and 3 remain relatively stationary, since the areas that surround them have nearly uniform importance. (If the importance function were to have greater spread, we would expect Servers 0 and 3 to similarly tend toward Clients 1 and 2.)





(a) *(b)*

Figure 3: We ran another client-server simulation using an alpha-modified importance function. In each simulation, alpha values were sampled from normal distributions whose parameters are listed in the plot subtitles.

E(i)(2). As shown in Figure 4, spoofer clients sample from a (normal) distribution with $E(\alpha_i) = 0.99$, while legitimate ones sample from a (normal) distribution with $E(\alpha_i) = 0.01$. As a result, all of the servers tend toward the spoofed Clients 0 and 3. Interestingly, Server 0 moves upward for some amount of time, likely because it encountered the increase in importance shown near position (5, 1) in Figure 4(b). While the scenario studied in Part E(i)(1) is nearly optimal, this scenario is the opposite.

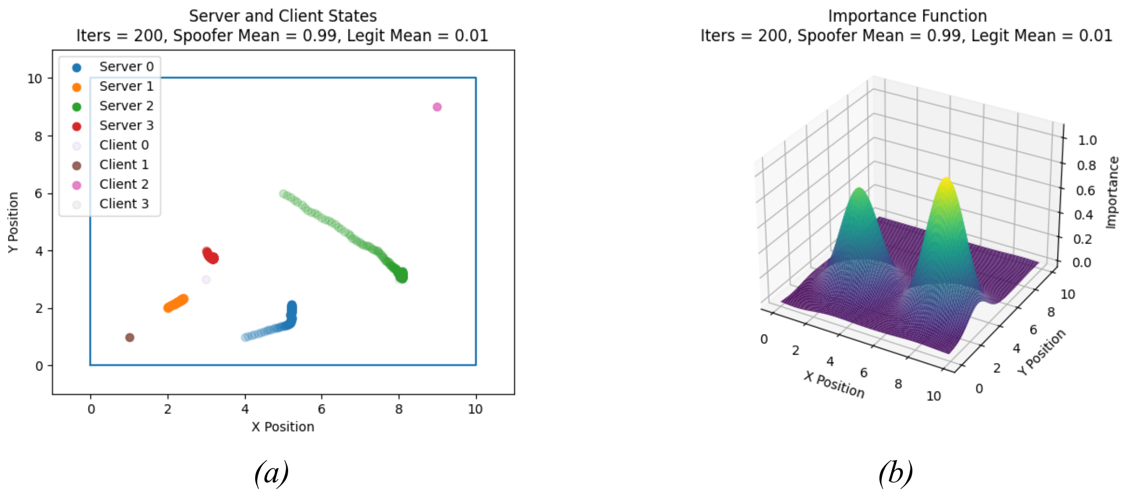
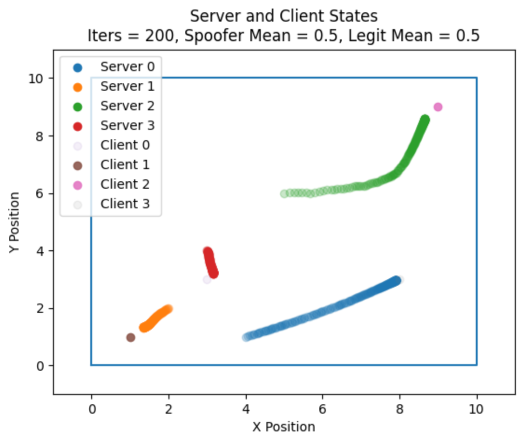
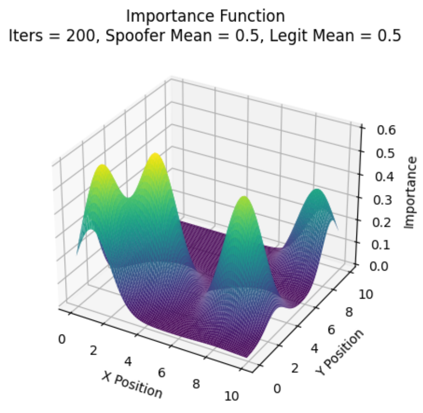


Figure 4: We ran another client-server simulation using an alpha-modified importance function. In this simulation, alpha values were sampled from normal distributions whose parameters are listed in the plot subtitles.

E(ii). As advised and shown in Figure 5, both spoofer and legitimate clients sample from a (normal) distribution with $E(\alpha_i) = 0.50$. For this reason, servers behave like they did in part B, tending toward both legitimate and spoofed clients. This implies that using the same $E(\alpha_i)$ for both types of clients is virtually equivalent to using an importance function that is not alpha-modified; in this system, there is no recognition of, or defense against, spoofed clients.



(a)



(b)

Figure 5: We ran another client-server simulation using an alpha-modified importance function. In this simulation, alpha values were sampled from normal distributions whose parameters are listed in the plot subtitles.

Problem 2

Note: On line 13 of spoofing.py, you can change the variable "question" to any of the strings in the list "Q2" (on line 10) to run different graphs. For example, if question = "2c", then the program will output the graphs for question 2C. Furthermore, we created a consensus-detection system to indicate convergence on our plots (with a star). This system is not perfect and, as you may see in some plots, placed the star a bit earlier than it should have. Nevertheless, it served as a good tool with which to estimate final consensus values.

A. We have included an image of our program output in Figure 6 below.

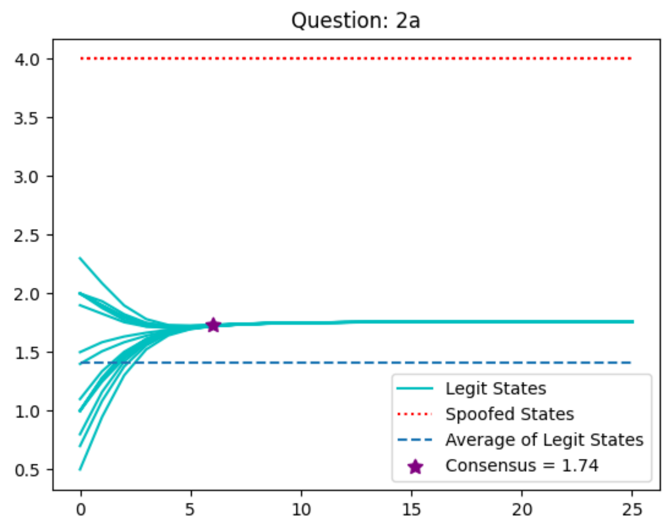
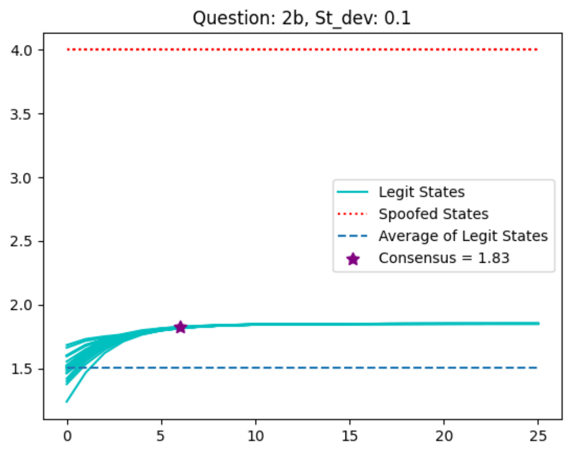
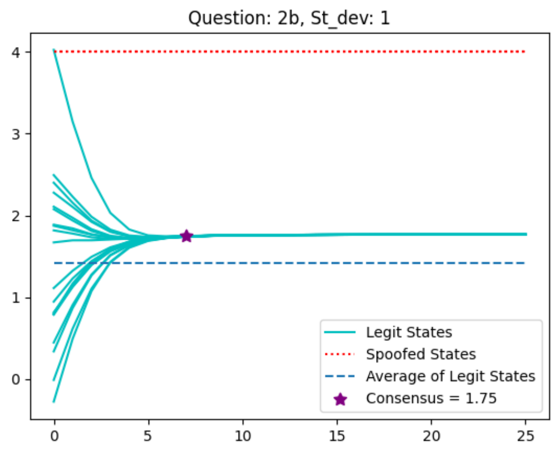


Figure 6: We implemented updates to the W matrix according to Gil and her colleagues' paper, then tested our implementation using the given parameters.

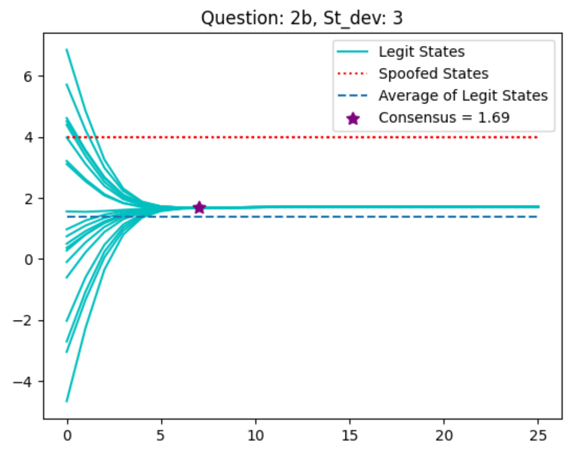
B. As shown in Figure 7, the standard deviation does not seem to have much of an effect on the system's convergence speed (but may slightly decrease it). Furthermore, the higher the standard deviation of the initial legitimate states, the closer the converged legitimate states were to their average. We suspect that this is because higher standard deviations decrease the effects of spoofed states on the system–reducing the extent to which the spoofed states are outliers. We imagine that, if there is already significant variance in the legitimate states, an especially high or low spoofed state would not be very impactful.


Question: 2b, St_dev: 0.1
Legit States
Spoofed States
Average of Legit States
Consensus = 1.83
4.0
3.5
3.0
2.5
2.0
1.5
0
5
10
15
20
25

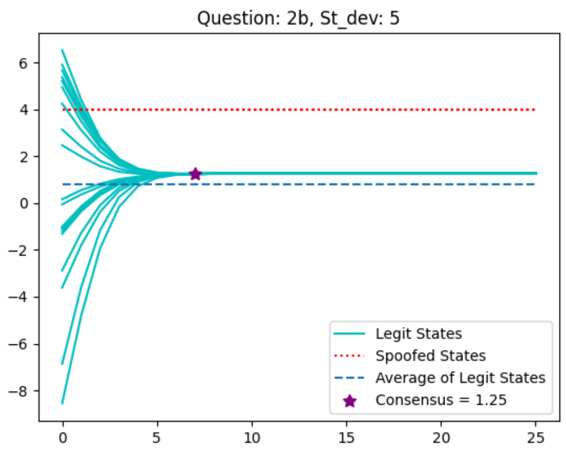
(a)


Question: 2b, St_dev: 1
Legit States
Spoofed States
Average of Legit States
Consensus = 1.75
4
3
2
1
0
0
5
10
15
20
25

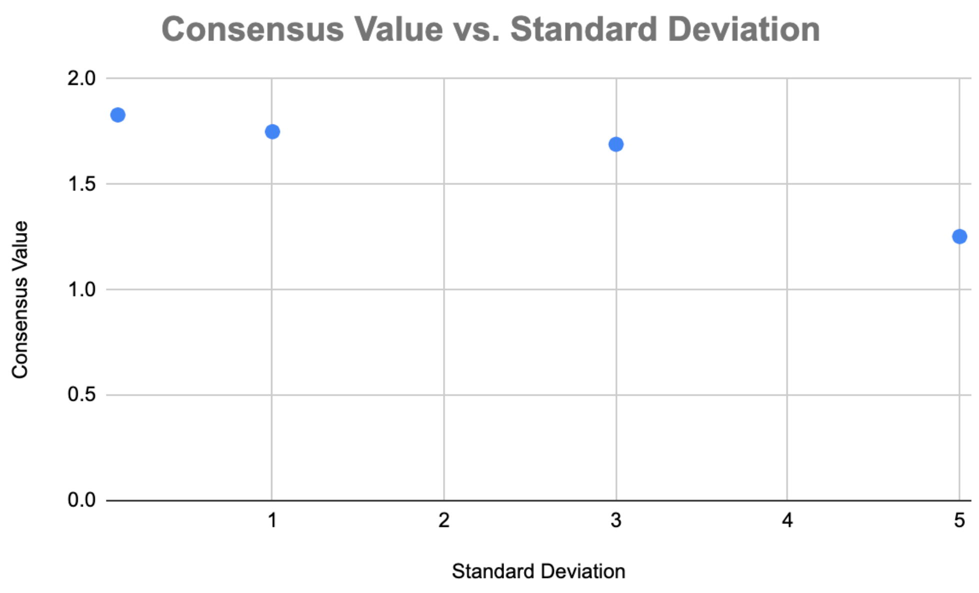
(b)


Question: 2b, St_dev: 3
Legit States
Spoofed States
Average of Legit States
Consensus = 1.69
6
4
2
0
−2
−4
0
5
10
15
20
25

(c)


Question: 2b, St_dev: 5
Legit States
Spoofed States
Average of Legit States
Consensus = 1.25
6
4
2
0
−2
−4
−6
−8
0
5
10
15
20
25

(d)



(e)

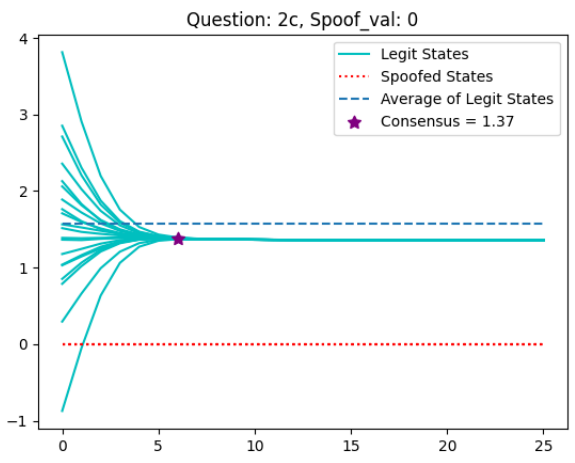
Figure 7: We ran a simulation in which 20 legitimate agents drew their initial states from a normal distribution with a mean of 1.5 and standard deviations that are listed in the plot subtitles. This simulation included four spoofed nodes, whose states were 4.

C.

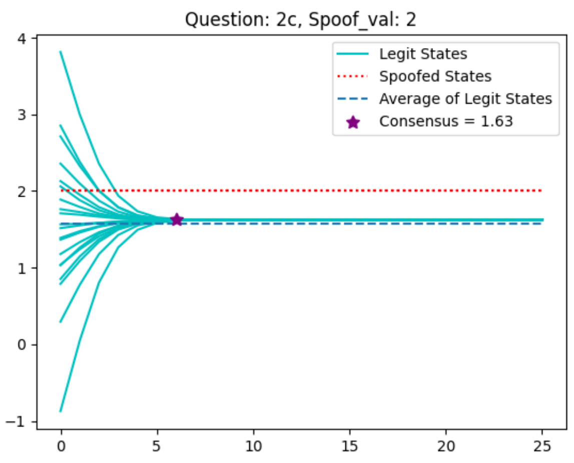
Spoofed State	Consensus Value
0	1.374566554
2	1.625258128
4	1.881125778
6	2.157792113

Table 1

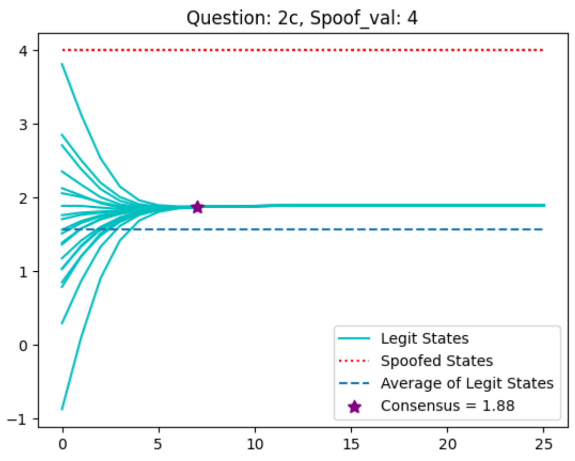
As shown in Table 1, the spoofed value seems to have a positive (and somewhat linear) effect on the consensus value. This is expected: although the system is resilient against spoofing, it is still somewhat affected by it, especially when spoofed states are notably outlying legitimate states. With more "different-from-legitimate" states, spoofed agents can more strongly attract legitimate agents and, thus, draw the system's consensus value toward their spoofed states.



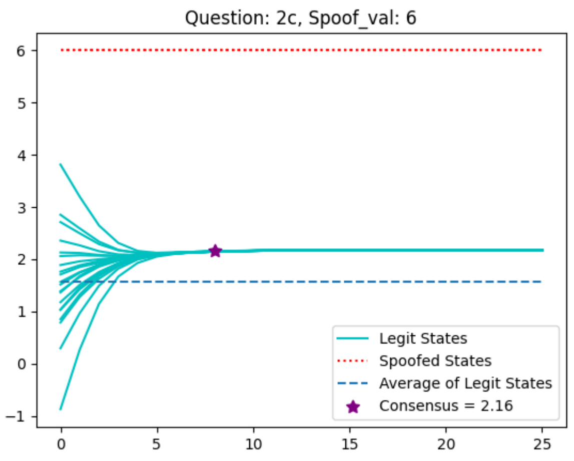
(a)



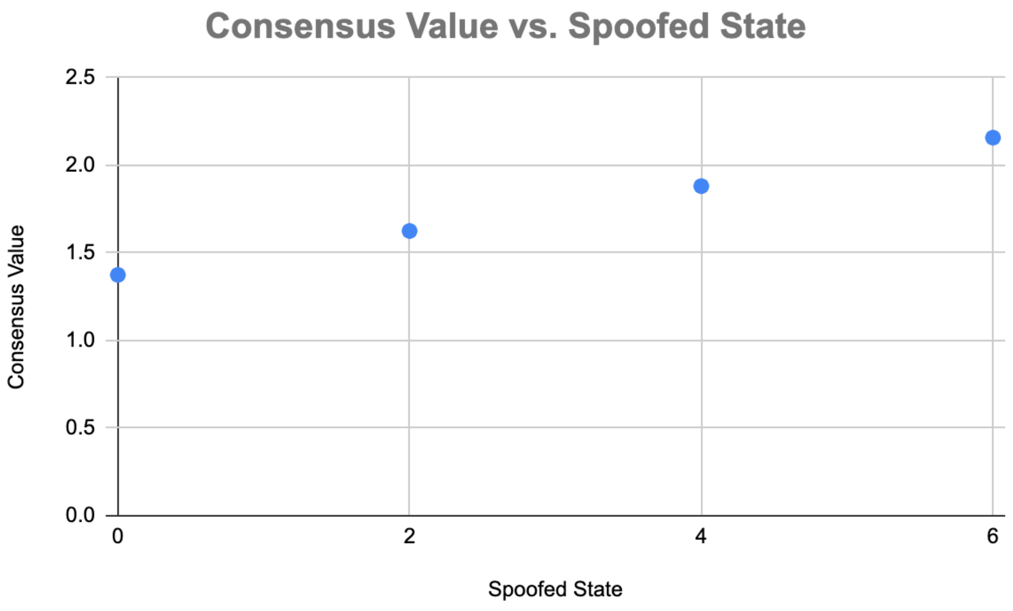
(b)



(c)



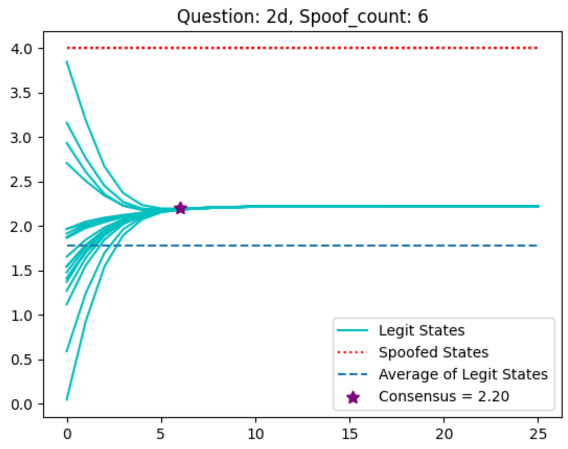
(d)



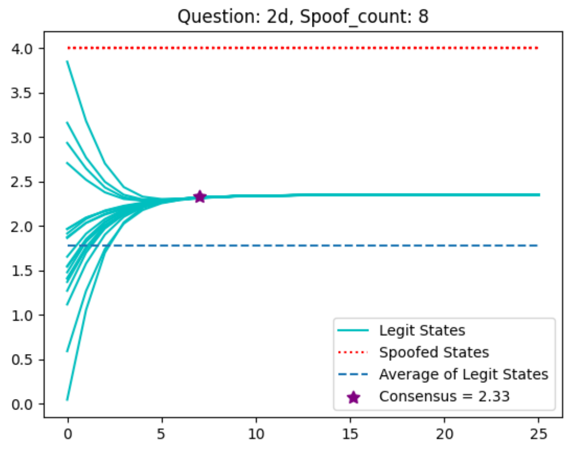
(e)

Figure 8: We ran another simulation in which legitimate agents drew their initial states from a normal distribution with a mean of 1.5 and a standard deviation of 1. In this simulation, four spoofers have initial states that are listed in the plot subtitles.

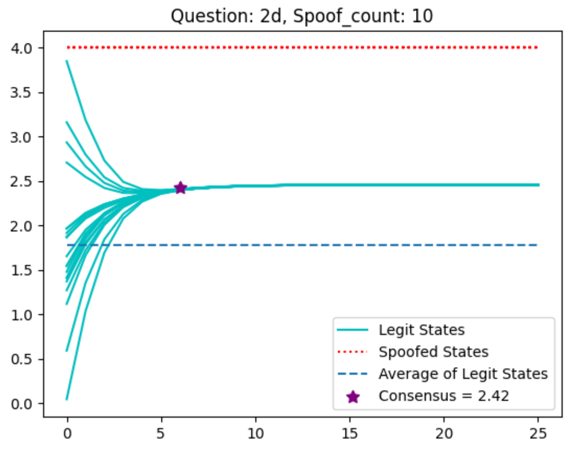
D. An increase in the number of spoofed nodes seems to cause an increase in the system's final consensus value. Reflected in Figure 9, this makes sense, since a greater number of spoofed nodes can have a larger effect on legitimate nodes–drawing them (and, thus, the entire system's consensus value) toward spoofed values. In fact, as we discussed in lecture, if the number of spoofed nodes is great enough relative to the system's underlying connectivity, it can be "fatal." It is worth noting that, in this case, the effect of spoofed nodes is fairly small, thanks to the resilient algorithm.


Question: 2d, Spoof_count: 6
Legit States
Spoofed States
Average of Legit States
Consensus = 2.20

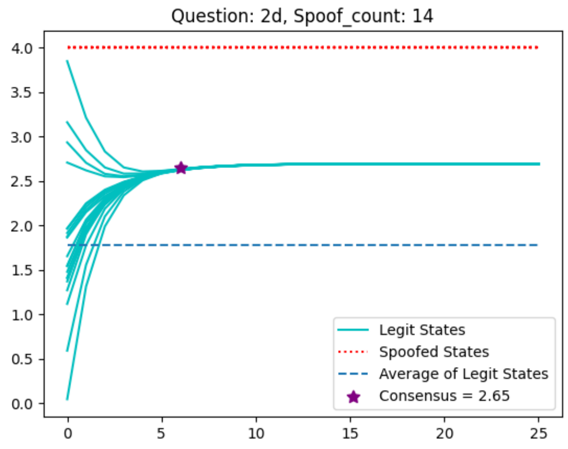
(a)


Question: 2d, Spoof_count: 8
Legit States
Spoofed States
Average of Legit States
Consensus = 2.33

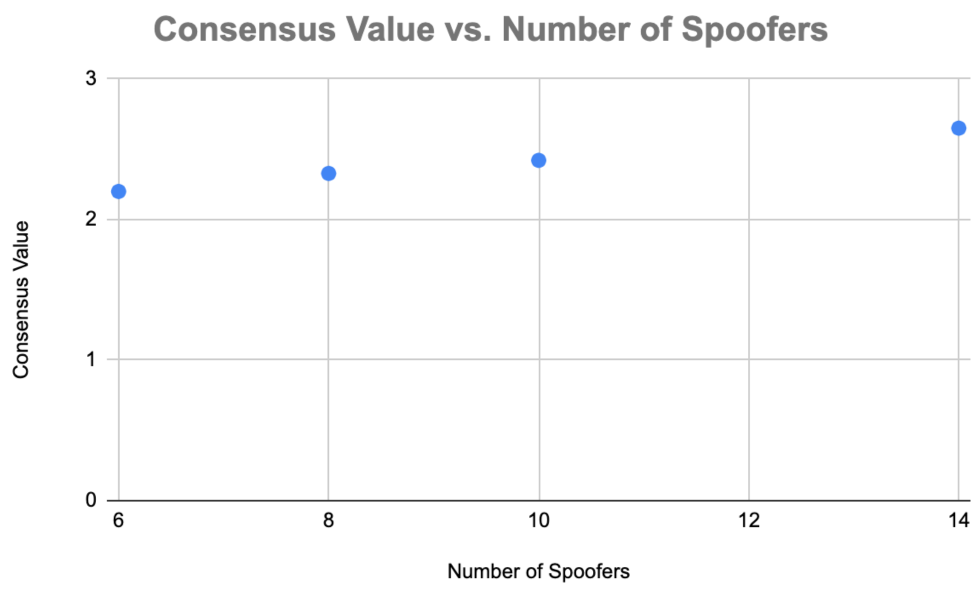
(b)


Question: 2d, Spoof_count: 10
Legit States
Spoofed States
Average of Legit States
Consensus = 2.42

(c)


Question: 2d, Spoof_count: 14
Legit States
Spoofed States
Average of Legit States
Consensus = 2.65

(d)



(e)

Figure 9: We ran another simulation that used the same parameters as the simulation in part C, but included a greater (and variable) number of spoofed agents. The numbers of spoofed agents are listed in the plot subtitles.

E(i). When the spoofers follow a sinusoidal pattern over time, there does not appear to be a significant change in the system's consensus. The system achieves consensus around the 10th time step and is thereafter unaffected by changes in spoofed states.

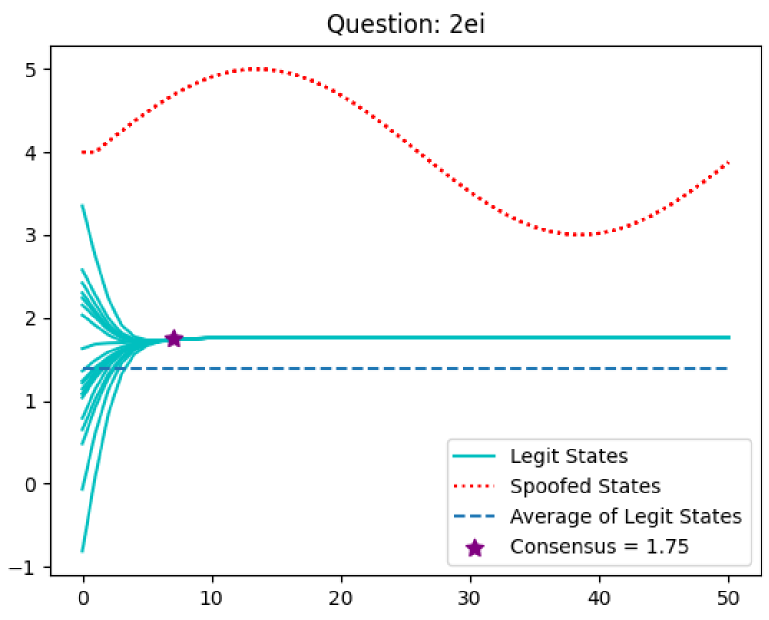


Figure 10: We ran a simulation in which spoofed agents changed their states over time, following a sinusoidal pattern.

E(ii). When the spoofers follow an exponential pattern over time, there does not appear to be a significant change in the system's consensus. (Indeed, the system is resilient!) There does, however, appear to be a more positive effect on the system's consensus value than in the case of part E(i). This is expected, since, prior to consensus, the exponential pattern results in an increasingly large difference between the spoofed states and legitimate states. As we saw in part 2(C), more "different-from-legitimate" spoofed states have larger effects on consensus.

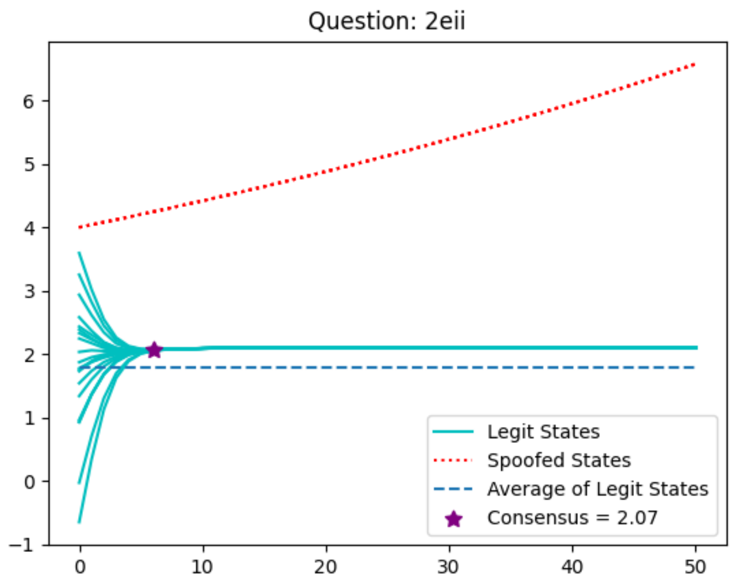


Figure 11: We ran a simulation in which spoofed agents changed their states over time, following an exponential pattern.