M.E.D.I.C B.O.T: A.I. in Healthcare

Team 26, Team M.E.D.I.C Bot

New Mexico SuperComputing Challenge

Final Report

05 April 2023

Team Number: 25

School Name: Justice Code

Team Members: Chinyere Offor, Mekhi Bradford, Kingsley Walker, Noel Emma-Asonye

Sponsor: Becky Campbell

Project Mentor: William Fong, Justin Schmetterer

Table of Contents

M.E.D.I.C Bot: A.I. in Healthcare

Executive Summary

Artificial Intelligence (AI) has contributed significantly to the fight against COVID-19, with its ability to diagnose diseases being one of its most significant contributions. Tools powered by AI can analyze medical images such as chest X-rays and CT scans to detect COVID-19 symptoms such as lung and organ damage. This capability is especially important in areas with limited medical resources, as it enables physicians and nurses to rapidly identify patients who require immediate care. AI can also aid in the development of new COVID-19 treatments and vaccines. Algorithms for machine learning can sift through vast quantities of medical research data to identify potential therapies or methods to combat the virus. AI can identify potential drug targets that could prevent the virus from penetrating human cells.

In addition, chatbots and virtual assistants powered by AI can provide accurate and up-to-date information on COVID-19, its symptoms, and its prevention. This reduces the anxiety and stress caused by the pandemic and provides support for mental health. In addition, AI can help monitor the virus's spread and predict when new outbreaks will occur. Algorithms powered by artificial intelligence can examine data from various sources, including social media, news reports, and healthcare systems, to identify patterns that can assist healthcare providers in effectively preparing for and responding to outbreaks.

In addition, AI has emerged as a potent disease identification and diagnosis tool. AI algorithms can rapidly identify potential disease risks, patterns, and risk factors by analyzing vast

quantities of medical data. For instance, AI-powered medical image analysis has proven effective in diagnosing cancer, cardiovascular disease, and Alzheimer's. Medical records, clinical notes, and scientific literature can be analyzed by natural language processing (NLP) algorithms to identify disease patterns and risk factors. Moreover, AI can create predictive models for disease diagnosis by analyzing patient data, such as medical history, genetics, lifestyle, and environmental factors, allowing clinicians to take preventative measures and lower the risk of disease progression.

AI for this purpose already exists, but is occasionally incapable of diagnosing thousands of patients quickly enough. In the North American continent. Every day, 611 000 people are admitted to hospitals; this number inflates dramatically when including other continents. How would an AI even attempt to complete such a monumental task? Nothing is impossible, but this is highly improbable. What is required now is something that is both straightforward and precise. We need a program that can accurately identify and diagnose disease while keeping up with the number of patients that are seen. This can only be achieved if AI is used as a crutch for medical professionals to lean on. A fast-working AI that would work in tandem with doctors instead of replacing them. This would reduce the time and cost of the diagnostic process and could, eventually, make treatment options in America more accessible to all. AI offers enormous potential for combating illnesses such as COVID-19, diagnosing diseases early, and developing individualized treatments, resulting in improved patient outcomes and healthcare as a whole.

Introduction

Problem Statement

AI is a tool that, with today's technology, can be a pathway to the next step in human development. However, when questioning what the purpose of such technology could be, an inquiry arises. Would it be plausible to create or code an AI that can accurately recognize and diagnose disease in a medical setting?

AI for this purpose does exist but it is incredibly hard to integrate into the rigorous, high-stakes environment of a hospital, especially at a caliber where it can diagnose diseases faster than a human being. Human doctors are usually unable to see the "bigger picture" during the diagnostic process often falling short to things like personal biases and exhaustion. An A.I. would not fail in the places that a human doctor would. Technology like this would be used as a crutch in the medical environment, catching things that a human cannot. A system like this would be invaluable in the process of diagnosing diseases such as COVID-19.

Background Information

The model will be focused on COVID-19, or Coronavirus. COVID-19 is a highly contagious respiratory disease caused by SARS-CoV-2. The virus is transmitted through respiratory droplets and close physical contact with an infected individual. Those who are elderly, immunocompromised, or who have underlying health conditions such as diabetes, heart disease, or obesity are more likely to develop severe symptoms and complications from COVID-19.

Early COVID-19 symptoms include fever, cough, fatigue, loss of taste or smell, and body aches. As the disease advances, it can cause respiratory distress, pneumonia, acute respiratory distress syndrome (ARDS), and, in severe cases, multi-organ failure and death. Variables such as vaccination rates, public health measures, and the emergence of new variants influence the incidence of COVID-19 infection.

COVID-19 has caused millions of confirmed cases and, according to the World Health Organization, over six million deaths worldwide. It is crucial to practice preventive measures, such as wearing masks, maintaining social distancing, and practicing proper hand hygiene, to help slow the spread of the virus and protect vulnerable populations. Detection and diagnosis at an early stage are crucial to the management and treatment of COVID-19 in order to prevent severe illness and fatalities.

Computational Model

Selection

The language that would best fit the purposes of this experiment would be Python, with its user input capabilities and expansive libraries, it would be perfect for a project like this. A Netlogo model can be used to demonstrate how well the A.I. would perceive and acknowledge some factors that may help diagnosis.

Modifications

Near the end of the challenge, team 26 realized that our original choice of disease could not be accurately diagnosed by both a medical professional and an AI using symptoms alone. So, the code for our computational model was promptly changed to a disease of which would fit the criteria of only symptom diagnosis such as COVID-19.

Visualization

Throughout this coding portion, we have changed our coding language multiple times. Rendering out code unfinished due to the stopping of development in NetLogo. What we have so far is a grey grid and a blue circle spawned by this code:

```
to Test-1
  create-turtles 1
[     setxy random-xcor random ycor
      set shape "circle"
      set size 5
      set color blue
]
  draw-grid
end

to draw-grid
  ask patches [
    sprout 1 [
    set color gray
    set heading 0
    fd .5
    rt 90
    pd
    repeat 4 [fd .5 rt 90 fd .5]
    die
  ]
]
End
```

We also have typed code to spawn 4 circles in each corner of the screen. Although this portion of the code is not finished. It was meant to spawn four different colored circles on each corner of the screen to symbolize the infected zone. Then we would have 4 different turtles spawn from each of them. Symbolizing those who caught the disease in the infected zone. The code for this is:

```
to create-circles
 [
  create-circle [
    set color red
   setxy min-pxcor +1 max-pxcor -1
 ]
 create-circle 1 [
   set color green
   setxy min-pxcor +1 max-pxcor -1
 ]
 create-circle 1 [
   set color blue
   setxy min-pxcor +1 max-pxcor -1
 ]
 create-circle 1 [
   set color yellow
   setxy min-pxcor +1 max-pxcor -1
 ]
 ]
 ask turtle 151 [
   setxy min-pxcor +1 max pxcor -1
 ]
 ask turtle 152 [
   setxy min-pxcor +1 max pxcor -1
 ]
 ask turtle 153 [
   setxy min-pxcor +1 max pxcor -1
 ]
```

```
  ask turtle 154 [
    setxy min-pxcor +1 max pxcor -1
  ]
]
end
to create-circle [n]
  set shape "circle"
  set size 2
]

End
```

Along with this code we also have some base code changing the color of the turtles when they interact with each other, allowing the turtles to be healthy, to be sick, to be treated, and diseased. The previously mentioned code allowing for the four circles on each side is not able to be screenshotted in action because of the unfinished aspect. However, Here are some screenshots of the code we do have.

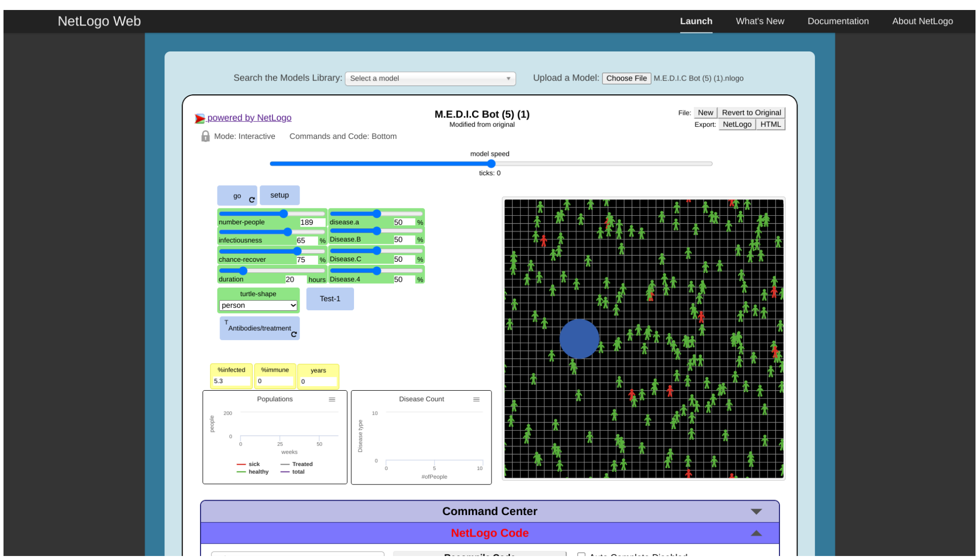

FIG1: This is the grey grid portion of the code along with the infected circle zone.

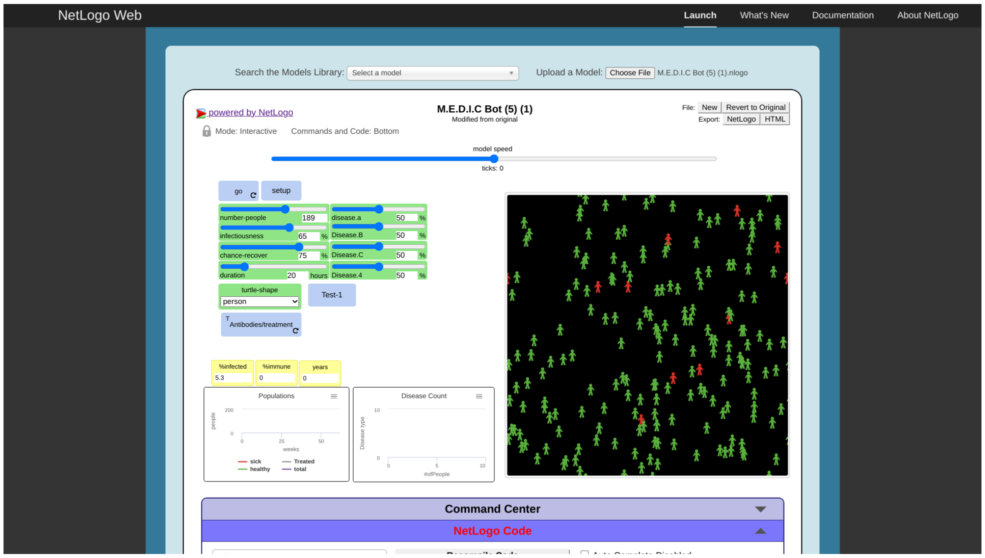

FIG 2: This is the turtle-interacting portion of the code allowing for the turtles to be healthy, sick, immune, and deceased.

Limitations

There have been many setbacks over the course of this experiment. Firstly, our team realized too late into the challenge that NetLogo would not serve our needs, and could only be used as a tool for visualizing how the AI might “see the big picture”. Another hurdle that the team had to jump was the original disease that was picked for the challenge, Necrotizing Fasciitis (NF). With the help of several medical professionals, we learned that this illness could not be identified by an AI without external tests and possibly surgical intervention, which would render any data we collected useless. So, in a last-minute decision, the team decided to switch our subject matter to COVID-19 instead of NF. The model that team 26 has currently, is limited to a simple display of how the AI would visualize the spread of a specific disease across a population to aid a doctor in making a more accurate guess. Unfortunately, however, the model that we have currently is not a functioning artificial intelligence.

Problem-Solving Method

The model that the team has is only capable of visuals demonstrating how the AI would model the spread of disease. But, in pure speculation, AI has already been used to identify the effects of COVID in X-rays and other tests outside of the symptoms of the patient. So, in later years, M.E.D.I.C Bot will be more effective and a working artificial intelligence will be, hopefully, finished.

Conclusion

Results

The finished model is able to accurately simulate how the AI would " see the big-picture" in relation to how disease transfers between different people and the AI's response to this. This AI is only in the beginning of its development, hopefully, with the experience provided from this year, the M.E.D.I.C bot will continue its slow improvement.

Discussion

The creation of this model led to some questions among personnel. Is there a moral obligation to continue to develop technology of this manner in an attempt to improve our world, or is the unknown better left undiscovered? This take is supported by the recent discussion about the threat AI poses to the public. When creating an algorithm that will be used in hospitals, a lack of any bias whatsoever is imperative to prevent the deaths of any patients.

Future Work

As previously stated, AI for this purpose does exist but is incredibly hard to integrate into the rigorous environment of a hospital, especially to the point where it can diagnose diseases faster than a human being. We hope to build an A.I. that is capable of diagnosis after being given symptoms, medical history, ethnicity/race, and other variables. The goal of this experiment was to write a program that can identify and diagnose different diseases faster and more accurately than a human doctor. This would allow for faster and cheaper procedures for healthcare in the United States. Medical professionals could simply input the symptoms of any patient into a reliable diagnostic tool and receive a list of tests and illnesses that correspond with any input provided. A.I. is the next step to improving healthcare in the United States of America, and across the world. Utilizing this technology correctly could make healthcare cheaper, faster, and easily accessible to everyone. The M.E.D.I.C Bot team hopes to demonstrate this through our model, one disease at a time.

Acknowledgments

Team 26 would like to commend our NSBE Director, Ms. Caia Brown, for her guidance and support throughout this project. Becky Campbell and Ryan Palmer were a source of inspiration and support while we worked on our project, and we would like to thank them both. We would also like to express our gratitude to Justin Schmetterer and William Fong for providing us with mentorship and useful tips throughout the whole process. Thank you.

References

American Lung Association. 20 Jan. 2022, www.lung.org/lung-health-diseases/lung-disease-lookup/pulmonary-embolism/symptoms-diagnosis. Accessed 17 Feb. 2023.

American Lung Association. 17 Nov. 2021, www.lung.org/lung-health-diseases/lung-disease-lookup/pneumonia/symptoms-and-diagnosis. Accessed 17 Feb. 2023.

Davenport, Thomas, and Ravi Kalakota. "The Potential for Artificial Intelligence in Healthcare." *Future Healthcare Journal*, vol. 6, no. 2, June 2019, pp. 94-98, https://doi.org/10.7861/futurehosp.6-2-94. Accessed 21 Jan. 2023.

Ditsios, Konstantinos, et al. "Necrotizing Fasciitis of the Upper Extremity – a Review." *Orthopedic Reviews*, vol. 14, no. 3, 25 Aug. 2022, https://doi.org/10.52965/001c.35320. Accessed 8 Mar. 2023.

Fatimah, Lateef, et al. "Necrotizing Fasciitis: How Reliable Are the Cutaneous Signs?" *Journal of Emergencies, Trauma, and Shock*, vol. 10, no. 4, 2017, p. 205, https://doi.org/10.4103/JETS.JETS_42_17. Accessed 8 Mar. 2023.

"Group A Streptococci (G.A.S)." *Centers for Disease Control and Prevention*, 3 Mar. 2022, www.cdc.gov/groupastrep/index.html. Accessed 8 Mar. 2023.

IBM. www.ibm.com/topics/machine-learning. Accessed 17 Feb. 2023.

"Necrotizing Fasciitis." *National Organization for Rare Diseases*, 2009, rarediseases.org/rare-diseases/necrotizing-fasciitis/. Accessed 8 Mar. 2023.

"Necrotizing Fasciitis: All You Need to Know." *Centers for Disease Control and Prevention*, 17 Nov. 2022, www.cdc.gov/groupastrep/diseases-public/necrotizing-fasciitis.html#diagnosis. Accessed 8 Mar. 2023.

Wilensky. M.E.D.I.C. Bot. *NetLogo Web*, Northwestern University, 1998, www.netlogoweb.org/launch#Load. Accessed 4 Apr. 2023.

World Health Organization. "WHO Coronavirus(COVID-19)Dashboard." *WHO*, covid19.who.int/. Accessed 4 Apr. 2023.

Appendix: Code

```
turtles-own
  [ sick?
    remaining-immunity
```

```
    sick-time
    age ]

globals
  [ %infected
    %immune
    lifespan
    chance-reproduce
    carrying-capacity
    immunity-duration ]

;; The setup is divided into four procedures
to setup
  clear-all
  setup-constants
  setup-turtles
  update-global-variables
  update-display
  reset-ticks
end

;; We create a variable number of turtles of which 10 are infectious,
;; and distribute them randomly
to setup-turtles
  create-turtles number-people
    [ setxy random-xcor random-ycor
```

```
      set age random lifespan
      set sick-time 0
      set remaining-immunity 0
      set size 1.5  ;; easier to see
      get-healthy ]
  ask n-of 10 turtles
    [ get-sick ]
end

to Test-1
  create-turtles 1
[     setxy random-xcor random ycor
      set shape "circle"
      set size 5
      set color blue
]
  draw-grid
end

to draw-grid
  ask patches [
    sprout 1 [
    set color gray
    set heading 0
    fd .5
    rt 90
```

```
    pd
    repeat 4 [fd .5 rt 90 fd .5]
    die
  ]
]
end

to get-sick
  set sick? true
  set remaining-immunity 0
end

to get-healthy
  set sick? false
  set remaining-immunity 0
  set sick-time 0
end

to become-immune
  set sick? false
  set sick-time 0
  set remaining-immunity immunity-duration
end

to setup-constants
  set lifespan 50 * 52      ;; 50 times 52 weeks = 50 years = 2600 weeks old
```

```
  set carrying-capacity 300
  set chance-reproduce 1
  set immunity-duration 52
end

to go
  ask turtles [
    get-older
    move
    if sick? [ recover-or-die ]
    ifelse sick? [ infect ] [ reproduce ]
  ]
  update-global-variables
  update-display
  tick
end

to update-global-variables
  if count turtles > 0
    [ set %infected (count turtles with [ sick? ] / count turtles) * 100
      set %immune (count turtles with [ immune? ] / count turtles) * 100 ]
end

to update-display
  ask turtles
    [ if shape != turtle-shape [ set shape turtle-shape ]
```

```
    set color ifelse-value sick? [ red ] [ ifelse-value immune? [ grey ] [ green ] ] ]
end


;;Turtle counting variables are advanced.
to get-older ;; turtle procedure
  ;; Turtles die of old age once their age exceeds the
  ;; lifespan (set at 50 years in this model).
  set age age + 1
  if age > lifespan [ die ]
  if immune? [ set remaining-immunity remaining-immunity - 1 ]
  if sick? [ set sick-time sick-time + 1 ]
end


;; Turtles move about at random.
to move ;; turtle procedure
  rt random 100
  lt random 100
  fd 1
end


;; If a turtle is sick, it infects other turtles on the same patch.
;; Immune turtles don't get sick.
to infect ;; turtle procedure
  ask other turtles-here with [ not sick? and not immune? ]
    [ if random-float 100 < infectiousness
      [ get-sick ] ]
```

```
end


;; Once the turtle has been sick long enough, it
;; either recovers (and becomes immune) or it dies.
to recover-or-die ;; turtle procedure
  if sick-time > duration
    [ ifelse random-float 100 < chance-recover
      [ become-immune ]
      [ die ] ]
end


;; If there are less turtles than the carrying-capacity
;; then turtles can reproduce.
to reproduce
  if count turtles < carrying-capacity and random-float 100 < chance-reproduce
    [ hatch 1
      [ set age 1
        lt 45 fd 1
        get-healthy ] ]
end


to-report immune?
  report remaining-immunity > 0
end


to startup
```

```
  setup-constants ;; so that carrying-capacity can be used as upper bound of number-people slider


end


to create-circles
  [
   create-circle [
      set color red
    setxy min-pxcor +1 max-pxcor -1
  ]


  create-circle 1 [
    set color green
    setxy min-pxcor +1 max-pxcor -1
  ]


  create-circle 1 [
    set color blue
    setxy min-pxcor +1 max-pxcor -1
  ]


  create-circle 1 [
    set color yellow
    setxy min-pxcor +1 max-pxcor -1
  ]
  ]
```

```
  ask turtle 151 [
    setxy min-pxcor +1 max pxcor -1
  ]
  ask turtle 152 [
    setxy min-pxcor +1 max pxcor -1
  ]
  ask turtle 153 [
    setxy min-pxcor +1 max pxcor -1
  ]
  ask turtle 154 [
    setxy min-pxcor +1 max pxcor -1
  ]
]

end

to create-circle [n]
  set shape "circle"
  set size 2



end
```