A Model System for Study of Cyberattacks on Biomanufacturing

Brenden Fraser-Hevlin, Adam Caudle, Mikayden Weise, Yaacov Mitsourov, Alec Schuler, Arda Gozen, Bernard Van Wie

BACKGROUND

- **Cyberattacks: a major concern for biological systems and healthcare networks**
- **Healthcare cyberattacks make up 25% of all hacks, and each one costs avg. $5 million (*Nature*, 2022)**
- We designed a model bioreactor to study cyberattacks in biomanufacturing
- The reactor improves the efficiency of immunotherapy, a promising but expensive new cancer treatment
- The centrifugal bioreactor (CBR), **Figure 1,** balances fluid and centrifugal forces (**Figure 2**) to quickly grow killer T cells up to high density
- We have designed models to automate the process and account for effects of cyberattacks and interruptions

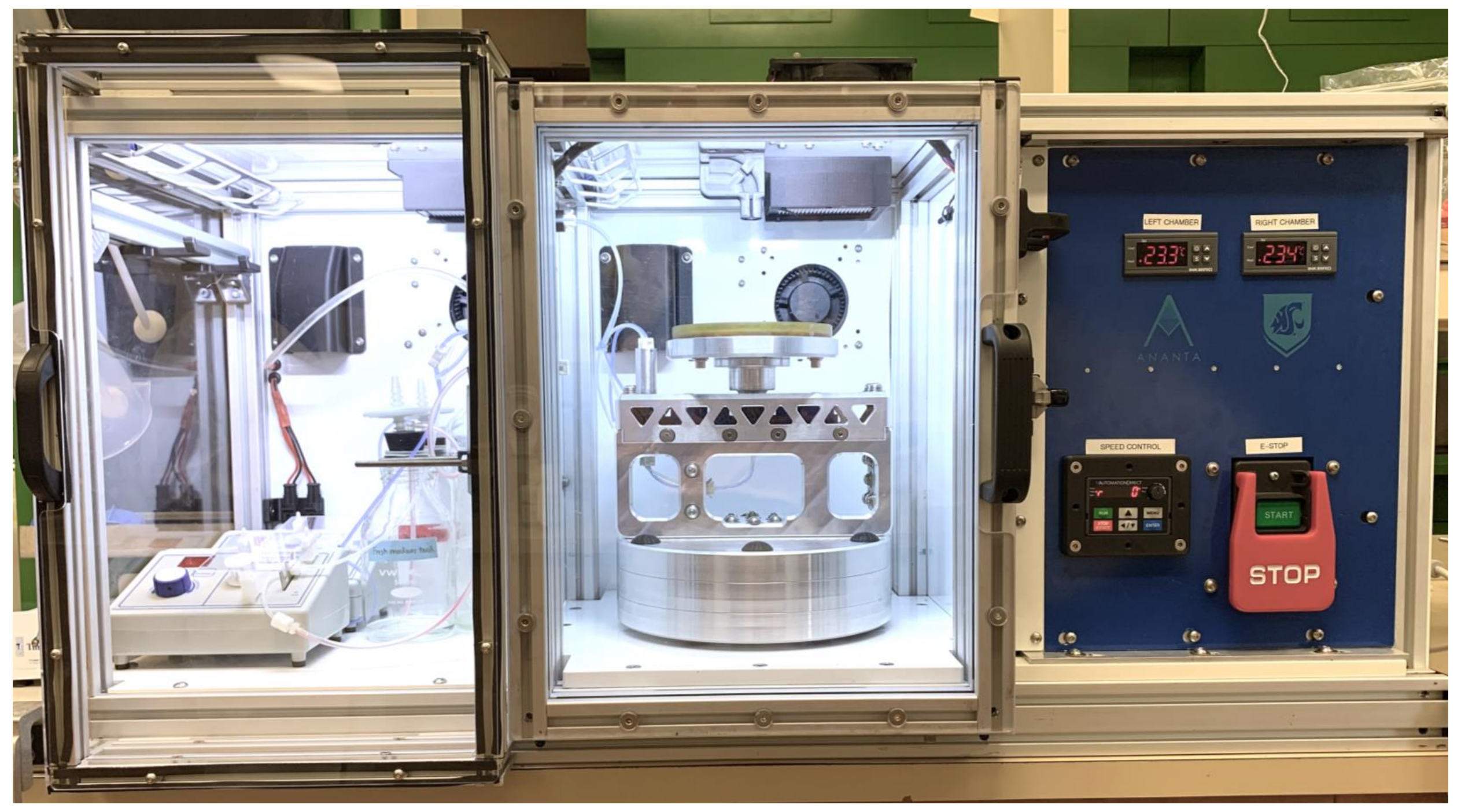


Figure 1: Current CBR prototype with centrifuge chamber visible in the center

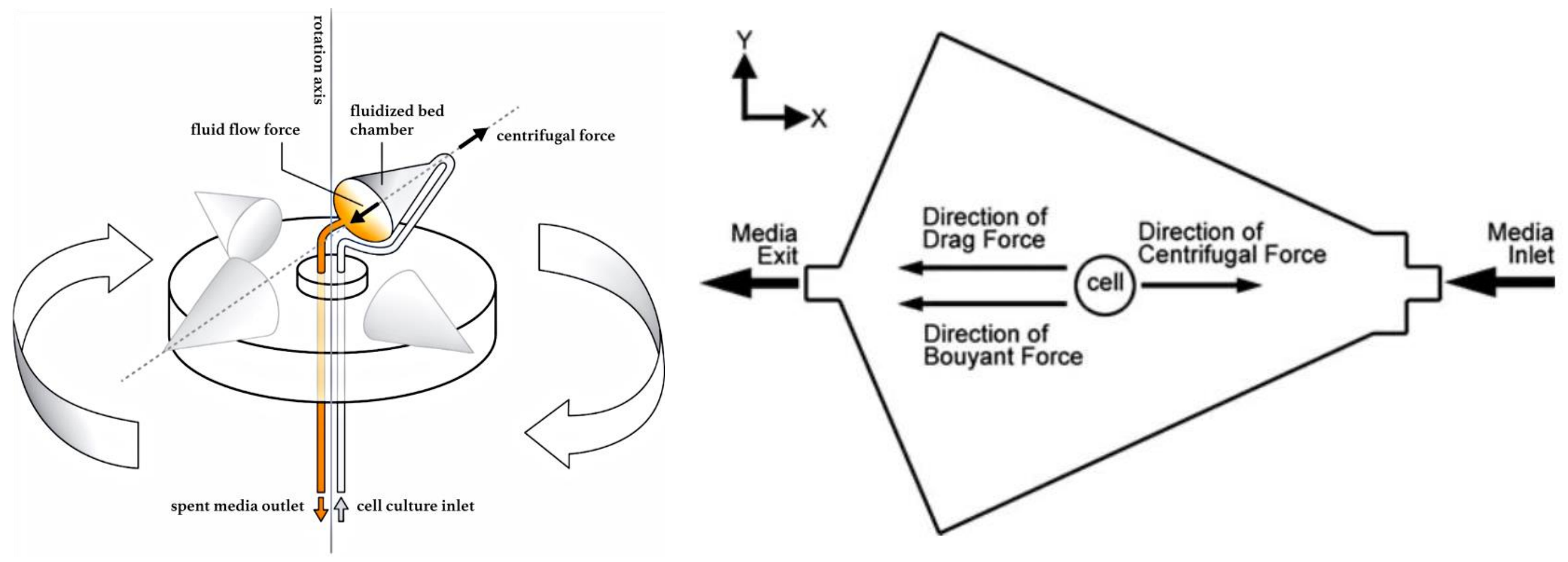


Figure 2: CBR chamber on top of centrifuge disc; balance of forces on each cell within the chamber

METHODS & RESULTS

Kinetic Modeling

- **Glucose and oxygen**: key nutrients for T cells. We optimized cell growth in the CBR using the following model:

$$(1)\ \frac{\partial C_{cell}}{\partial t} = \mu_{max} \cdot \left(1 - \frac{C_L}{C_{L\_max}}\right)^n \cdot \left(1 - \frac{C_A}{C_{A\_max}}\right)^m \cdot C_{Cell}$$

$$(2)\ \frac{\partial C_G}{\partial t} = D(C_{G0} - C_G) - Y_{GC} \cdot \left(\frac{\partial C_{Cell}}{\partial t}\right)$$

$$(3)\ \frac{\partial C_A}{\partial t} = D(C_{A0} - C_A) + Y_{AC} \cdot \left(\frac{\partial C_{Cell}}{\partial t}\right)$$

$$(4)\ \frac{\partial C_L}{\partial t} = D(C_{L0} - C_L) + Y_{LC} \cdot \left(\frac{\partial C_{Cell}}{\partial t}\right)$$

- Model calculates cell growth (**equation 1**) as function of glucose (**equation 2**), ammonium and lactate (toxic byproducts, **equations 3, 4**), additionally oxygen (not shown)
- Kinetic studies were performed in lab to measure each term in model. Results of modeling are shown below in Figure 3

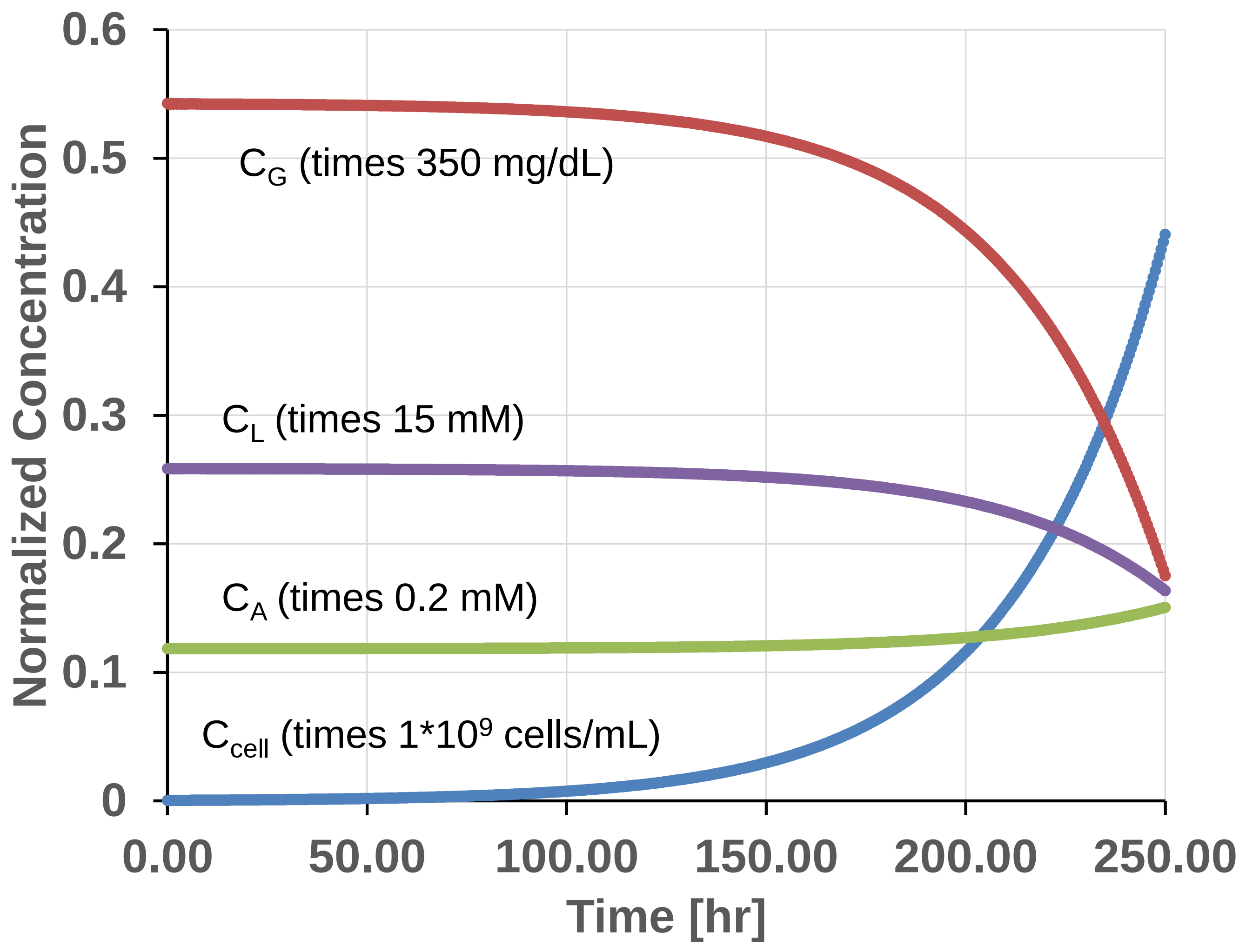


Figure 3: Model predictions of cell growth over 250 hours

Cyberattack Modeling and Prevention

- Our planned process, shown in **Figure 4**, will**:**
  - Collect glucose/oxygen data from sensors,
  - Send it to Simulink, compare real data to model
  - Use Arduino controllers to make adjustments
- Current process flow: software run through a browser (DOTS by SBI), with data sent to an Excel sheet in real time to compare reactor data to model predictions
  - Web browser makes system vulnerable to online attacks, as browsers on either side could be hacked, shown by red line (untrusted source)

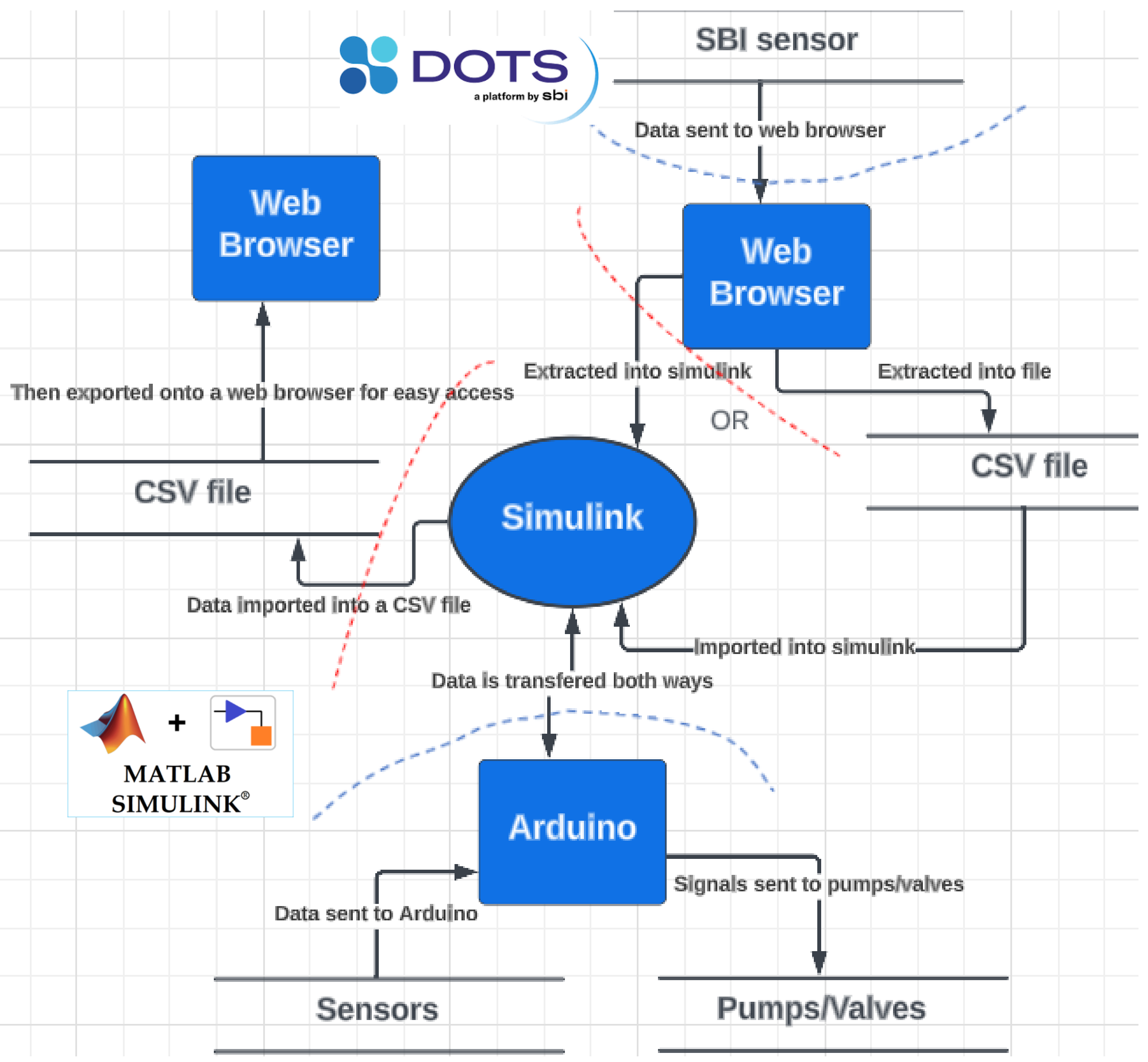


Figure 4: Overview of the reactor process and cybersecurity risks

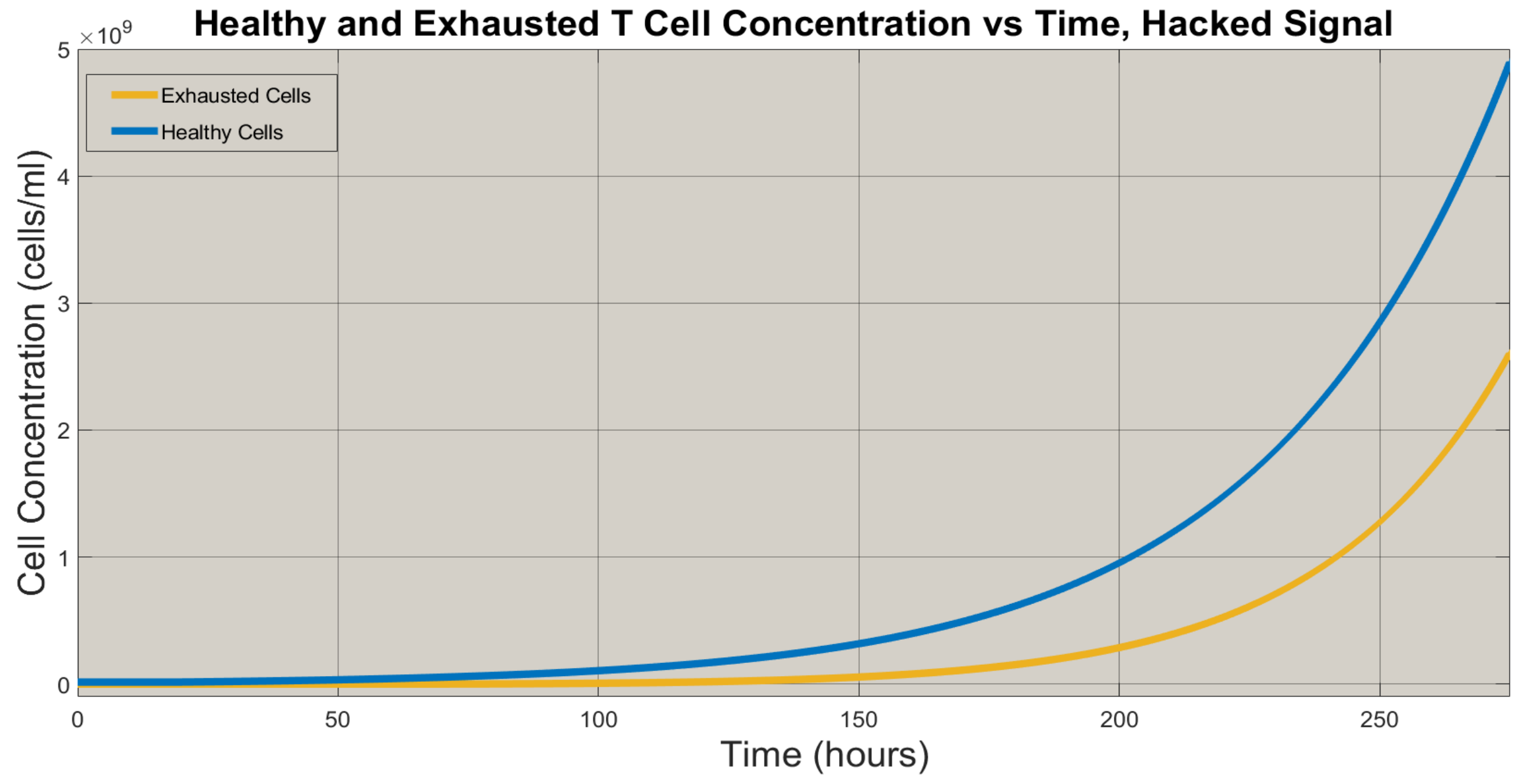
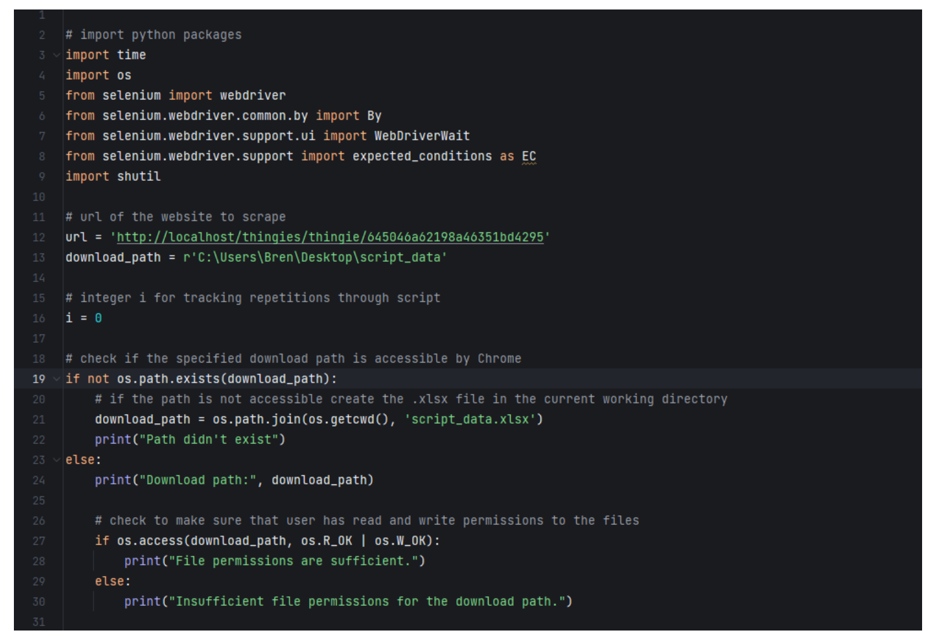


Figure 5: Simulated effect of hacks on the system introducing false data in Simulink

Automated Data Acquisition

- We designed a script:
  - Automatically logs into oxygen sensor software, downloads/overwrites current CBR data, locally updating while CBR runs
  - Uses Python and its libraries, such as Pandas and Selenium.
- Has applications for tracking local data compared to online data to maintain records, ensuring data validity



```python

# import python packages
import time
import os
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import shutil

# url of the website to scrape
url = 'http://localhost/thingies/thingie/645046a62198a46351bd4295'
download_path = r'C:\Users\Bren\Desktop\script_data'

# integer i for tracking repetitions through script
i = 0

# check if the specified download path is accessible by Chrome
if not os.path.exists(download_path):
    # if the path is not accessible create the .xlsx file in the current working directory
    download_path = os.path.join(os.getcwd(), 'script_data.xlsx')
    print("Path didn't exist")
else:
    print("Download path:", download_path)

    # check to make sure that user has read and write permissions to the files
    if os.access(download_path, os.R_OK | os.W_OK):
        print("File permissions are sufficient.")
    else:
        print("Insufficient file permissions for the download path.")

```

Figure 5: Script in Python which automates data collection

FUTURE WORK

- Further develop model and automated data collection system
- Goal: design process controllers that adjust key conditions (glucose/oxygen) in response to interruptions and hacks to the process

ACKNOWLEDGEMENTS & FUNDING

- The authors are grateful for funding from The Griffiss Institute under contract no. SAA 10012021MM0336, a VICEROY Project entitled Northwest Virtual Institute for Cybersecurity Education & Research (CySER)
- We would like to acknowledge funding and support from NSF Grant #1645249, the WSU Voiland College Shop, and the following organizations.