MCS Project Portfolio - Biocomputing (CSE 568)

Milind Parab
Arizona State University
Tempe, USA

mgparab@asu.edu

***Abstract*— This paper presents a portfolio of three projects in bio-inspired computing, focusing on two key concepts: Negative Selection (NS) and Ant Colony Collective Decision Making (ACO). The first project explores the application of Negative Selection for anomaly detection in sequence data, inspired by the human immune system's self/non-self-recognition. The project applies the Textor algorithm to classify English and non-English strings, using a model trained on English text to detect anomalies in other languages. Performance is evaluated using Receiver Operating Characteristic (ROC) analysis and Area Under the Curve (AUC), demonstrating the algorithm's ability to differentiate normal and abnormal data effectively. The second project adapts the Negative Selection algorithm for detecting anomalies in continuous cardiotocography (CTG) data, specifically for fetal health monitoring. By transforming continuous fetal heart rate and uterine contraction patterns into categorical representations, the model is trained to identify abnormal patterns, with AUC scores indicating strong detection performance. The third project utilizes agent-based modeling to simulate ant foraging behavior, specifically how ants collectively select food sources through exploration and tandem running. The Agent Based Modelling, implemented using the Mesa framework. Together, these projects showcase the versatility and real-world applications of bio-inspired algorithms in fields such as anomaly detection, medical diagnostics, and collective behavior modeling.**

Keywords—Negative Selection Algorithm, Anomaly Detection, Artificial Immune System, Evolutionary Computation, Ant Colony Optimization, Agent-Based Modeling, Path Optimization, Fetal Health Monitoring, Collective Decision-Making.

PROJECT 1 : INTRODUCTION TO NEGATIVE SELECTION PROJECT

I. INTRODUCTION

The concept of Negative Selection draws inspiration from the human immune system's remarkable ability to distinguish between "self" (the body's own cells) and "non-self" (foreign invaders). This crucial function, analogous to a castle's guards discerning between residents and intruders, is mediated by T-cells trained in the thymus to recognize and neutralize threats. In the realm of computational science, this principle translates into developing models that differentiate between normal (self) and abnormal (non-self) data.

This project investigates this concept by applying the Textor algorithm to classify text. Specifically, it aims to train a model on English text and then evaluate its ability to distinguish English from other languages, including Hiligaynon, Middle English, Plautdietsch, and Xhosa. By focusing on languages that share the same alphabet, this research explores the model's capacity to identify subtle linguistic differences.

The project utilizes a dataset consisting of 124 lines of English text and 500 lines of text from each non-English language. Each data point is labeled as either "self" (English) or "non-self" (non-English). The model is trained on fixed-length English strings extracted from "Moby Dick" and then tested on both English and non-English strings. The performance of the model is evaluated using anomaly scores, where each string is assigned a score based on its similarity to the "self" data. The final anomaly score for a given string is determined by summing the individual anomaly scores within the set.

II. PROBLEM STATEMENT AND APPROACH

In the initial phase of the project, I ran the model and verified that the final anomaly score was calculated correctly. This score is derived by summing the logarithms of the individual anomaly scores for each line in the text file.

In the second phase, I calculated the percentages of true positives (TP) and false positives (FP). Each language file contains 624 lines—124 in English and 500 in a non-English language. A true positive occurs when the model correctly identifies a non-English text, while a false positive occurs when the model misclassifies an English text as non-English.

For model evaluation, I used the Receiver Operating Characteristic (ROC) curve, which plots the false positive rate (FPR) against the true positive rate (TPR) for various threshold values. A classifier with an area under the curve (AUC) greater than 0.5 is considered meaningful, with values closer to 1 indicating near-perfect classification performance. Additionally, the ROC curve should lie above the diagonal, which represents a random classifier.

The ROC analysis, as described in Textor's paper, *"A Comparative Study of Negative Selection-Based Anomaly Detection in Sequence Data"*, was used as the second evaluation method. It provides insight into the trade-off between specificity and sensitivity, emphasizing the model's ability to distinguish between English (self) and non-English (non-self) text.

The model was trained with English sentences, each containing 10 characters, to teach the system to recognize English as the "self" language. Non-English text was introduced as "non-self" language, enabling the system to learn to distinguish between English and non-English languages. The training data included diverse language patterns, helping the system classify sentences based on their similarity to English. Following training, the model was

tested on languages like Hiligaynon, Plautdietsch, Middle English, and Xhosa, all of which were correctly classified as non-self. Notably, the AUC values for Hiligaynon, Xhosa, and Plautdietsch were higher than those for Middle English.

III. Solution and Results

The This project focuses on using the Negative Selection concept, inspired by the immune system's ability to distinguish self from non-self, to classify text as either English (self) or non-English (non-self).

Key Components:

1. **Model Training**: The model is trained on English text to establish a representation of "self" (normal) text.
2. **Anomaly Detection**: Non-English text is treated as anomalies, detected based on its deviation from the learned representation of English.
3. **Performance Evaluation**: The model's performance is assessed through metrics such as True Positives (TP), False Positives (FP), and Area Under the Curve (AUC), using Receiver Operating Characteristic (ROC) analysis.

Data:

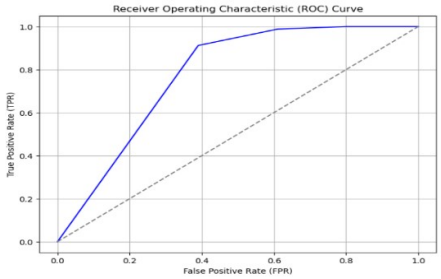
- **Training Data**: 124 lines of English text and 500 lines of non-English text.
- **Labels**: 0 for English (non-anomalous) and 1 for non-English (anomalous).
- **Anomaly Scoring**: Each string receives an anomaly score based on its similarity to English.

Methodology:

- The model is trained using fixed-length English strings from *Moby Dick* and then tested on non-English data, including languages like Hiligaynon, Middle English, Plautdietsch, and Xhosa.
- **Anomaly Scoring**: The final anomaly score is calculated by taking the log sum of individual anomaly scores for each test string.
- **Evaluation Metrics**: ROC curves and AUC are used to evaluate performance, with a high AUC (close to 1) indicating a strong classifier.

Results:

- The model successfully classified non-English languages such as Plautdietsch, Hiligaynon, and Xhosa as anomalies after training on English.
- Middle English posed a challenge due to its similarities to modern English, resulting in lower classification accuracy.
- **AUC**: The AUC values for Hiligaynon, Xhosa, and Plautdietsch were higher than for Middle English, indicating better performance on those languages.
- The below graph shows AUC for Hiligaynon



Learning Outcomes:

- **Model Evaluation**: The project deepened understanding of evaluating model performance using metrics like True Positives, False Positives, and AUC.
- **Hyperparameter Tuning**: Insights were gained on how hyperparameter tuning impacts model performance.
- **Real-World Applications**: The techniques used in this project are applicable to real-world anomaly detection tasks, such as fraud detection and spam filtering.

PROJECT 2: ANOMALY DETECTION WITH NEGATIVE SELECTION PROJECT

IV. Introduction

This project explores the application of the Textor algorithm, a bio-inspired anomaly detection technique rooted in the immune system's "self/non-self" recognition principle, to the domain of fetal health monitoring. Specifically, it focuses on detecting anomalies within cardiotocography (CTG) data, comprising continuous measurements of fetal heart rate (FHR) and uterine contractions. The study leverages the UC Irvine Machine Learning Repository's CTG dataset, encompassing both normal and abnormal fetal monitoring signals. The primary objective is to adapt the Textor algorithm, initially developed for textual data analysis, to identify abnormal patterns within these continuous physiological signals. This involves training the model on normal FHR and uterine contraction patterns and subsequently evaluating its ability to detect deviations from these established norms. The performance of the anomaly detection model is rigorously assessed using Receiver Operating Characteristic (ROC) analysis and the Area Under the Curve (AUC) metric.

V. Application of the Negative Selection Algorithm

The Textor algorithm, underpinned by the negative selection principle, draws inspiration from the immune system's ability to distinguish between "self" and "non-self" entities. In the context of anomaly detection, this algorithm operates by creating detectors that represent "self," or normal patterns, within the data. These detectors are designed to recognize and match patterns characteristic of normal data points while differentiating them from abnormal data points (outliers or "non-self"). Essentially, the core concept involves generating detectors that represent specific classes or patterns, such as normal heartbeats. Any data point that does not conform to these established detector patterns is flagged as an anomaly.

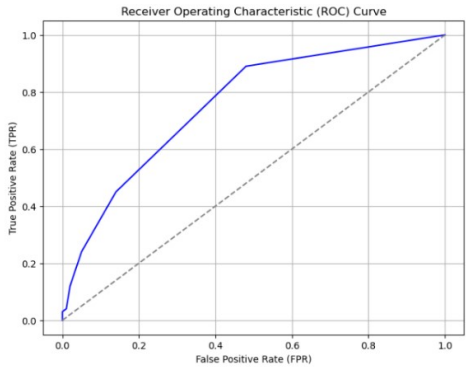
In this project, the Textor algorithm, specifically designed for sequence-based anomaly detection, was applied to identify

abnormalities within the CTG data. By defining normal fetal heart rate and uterine contraction patterns as "self," the algorithm was trained to recognize and identify any irregularities in the signals, such as those indicative of abnormal fetal heart rates or contractions. This approach necessitated the conversion of continuous CTG data into categorical representations (strings of symbols) to align with the Textor algorithm's requirements. Subsequently, the algorithm generated detectors that effectively matched the normal patterns within the transformed data. Upon encountering a test data point that did not conform to any of the established normal patterns, the algorithm classified it as an anomaly, suggesting a potential abnormality in the fetal heart rate or uterine contractions.

VI. Solution and Results

To prepare the data for the Textor algorithm, I first transformed the continuous variables into categorical data by binning them into 10 categories using pandas' `cut` function. This step converted the data into discrete symbols, making it compatible with the Textor algorithm, which requires discrete inputs. After preprocessing both the training and testing datasets into the required string format, I executed the Textor algorithm via subprocesses and evaluated its performance by calculating the Area Under the Curve (AUC). The **R-contiguous** detector focuses on identifying and excluding **consecutive patterns** or sequences of elements (denoted by "R") within the dataset. This type of detector is sensitive to linear, uninterrupted runs of these specific patterns, which may represent repetitive or undesirable features. In the context of negative selection, the goal is to filter out these continuous, unwanted patterns to refine the dataset and improve model accuracy. By adjusting the R value, we control the length and strictness of the detected contiguous patterns, optimizing model performance.

In this project, we are utilizing the **R-contiguous detector** type as part of our **negative selection** process. The objective is to fine-tune the detector value between 2 and 10 to identify the R value that yields the optimal **Area Under the Curve (AUC)** for the model's performance. So far, with **R = 2**, we achieved an AUC of 0.75, which is shown in the plot below.



when This project showcased the application of the Textor algorithm to real-world data and the evaluation of its performance using ROC analysis. The process of converting continuous data into categorical bins and running the negative selection algorithm emphasized the importance of preprocessing for effective anomaly detection. I learned the significance of adjusting model parameters, such as the contiguous pattern size, to optimize anomaly detection. The AUC values provided a reliable metric to quantify the performance of the model, confirming that the Textor algorithm can be successfully applied to diverse datasets when properly tuned. The model performed well in identifying anomalies in cardiovascular data, accurately detecting abnormal heart rate patterns. By fine-tuning the detector count and selecting the right detector types, the model's performance was notably improved. The r-contiguous and r-chuck detectors offered complementary insights, enhancing the model's ability to differentiate between normal and abnormal patterns.

Through this project, I developed a deeper understanding of how Negative Selection can be applied in medical diagnostics, particularly for anomaly detection in health data. Key takeaways included the importance of optimizing the detector count to achieve a balance between sensitivity and specificity.

1. **Intrusion Detection Systems (IDS)**: NSA detects abnormal network traffic to identify potential cyber-attacks by distinguishing between normal (self) and suspicious (non-self) behaviors.

2. **Medical Diagnostics**: NSA identifies abnormal health patterns in ECG, EEG, or CTG data, such as detecting fetal distress in pregnancy monitoring.

3. **Predictive Maintenance**: NSA monitors industrial equipment by detecting unusual sensor data, signaling potential failures in machinery to prevent costly breakdowns.

4. **Fraud Detection**: NSA detects fraudulent transactions by identifying anomalies in financial data, distinguishing legitimate transactions from potentially fraudulent ones.

5. **Image/Video Anomaly Detection**: NSA identifies abnormal objects or actions in images or video, such as detecting defects in manufacturing or unauthorized activities in surveillance footage.

PROJECT 3: Ant Collecting Behaviour Based on Agent Based Modelling

VII. Introduction

In this project, aim to simulate the collective foraging behavior of ants using an agent-based model (ABM), and compare its results with an existing ordinary differential equation (ODE) model. This approach helps in understanding how ants collectively select a food source, primarily through behaviors such as exploration and tandem running. The core focus is on translating an ODE model, which traditionally represents the behavior of ants in aggregate form, into an agent-based model, where each ant is modeled as an individual agent that interacts with others within a specified environment. The simulation is implemented using the Mesa package in Python, a powerful tool for building and analyzing agent-based models. By examining the ants' behavior and recruitment patterns, we explore how ants dynamically adapt to their environment and make decisions collectively.

In real-life ant colonies, ants employ a fascinating and highly efficient decision-making process to locate food sources and communicate with each other. This process is governed by

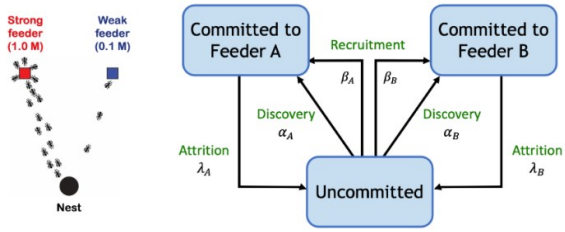
the principles of Ant Colony Optimization (ACO) and the collective decision principle, which is rooted in the concept of pheromone trails.

When ants are foraging for food, they leave behind a trail of pheromones as they move. The intensity of this pheromone trail depends on the strength and proximity of the food source. Initially, ants explore different paths randomly, but as they discover a food source, they begin to reinforce their path by depositing more pheromones along it.

The key mechanism in this process is positive feedback: the more ants follow a particular path and reinforce it with pheromones, the stronger the trail becomes. Other ants, sensing this increase in pheromone concentration, are more likely to follow the same path, reinforcing the trail further. This collective behavior allows the ants to gradually converge on the strongest food source, the one with the highest pheromone accumulation, which is often the most abundant or easily accessible.

In contrast, paths that are less frequently traveled or lead to weaker food sources will have fewer pheromones and thus, will be less attractive to other ants. Over time, the pheromone concentration on these paths evaporates, making them even less likely to be chosen. The collective decision-making principle ensures that, through this decentralized and dynamic process, the colony can efficiently locate the optimal food source, demonstrating remarkable adaptability and efficiency in their foraging behavior.

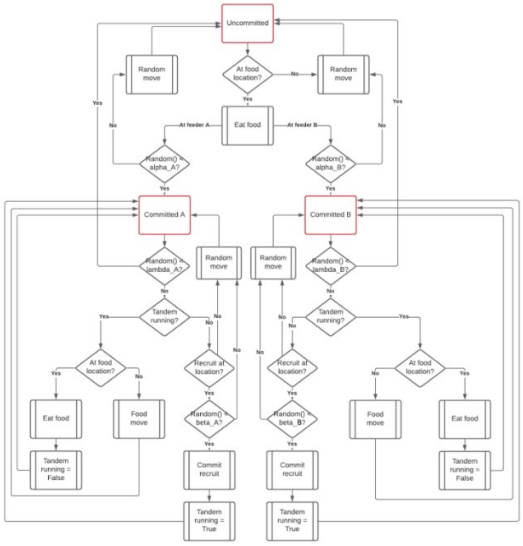
This form of self-organization in ant colonies, driven by pheromone communication, not only helps ants in finding food but also serves as a foundational concept for algorithms in optimization problems, such as Ant Colony Optimization, which mimic this natural process to solve complex problems in areas like logistics, network routing, and machine learning.



VIII. Results and Conclusion

The simulation, based on the agent-based model developed using the Mesa framework, successfully replicates the key features of the foraging experiment as described by Dr. Stephen Pratt in the "Collective Cognition in Ant Societies" lecture. The model allows ants to be in one of three states: uncommitted, committed to feeder A, or committed to feeder B, and exhibits behaviors such as tandem running for recruitment. The results of the simulation were compared to the ODE model, with a particular focus on the transition between food sources and the dynamics of recruitment.

Ants explore two food sources (feeders) placed equidistant from their nest. One feeder is stronger, with 1M fructose concentration, and the other is weaker, with 0.1M fructose. Ants discover the feeders with equal probability. The discovery rate for the strong feeder (A) is represented by alpha A, while for the weak feeder (B), it is alpha B. The recruitment rate for feeder A is beta A, and for feeder B, it is beta B. Attrition rates are represented by lambda A for feeder A and lambda B for feeder B.
At the start, all ants are at the nest. Initially, they move in random directions. Ants can be in one of three states: uncommitted, committed to feeder A, or committed to feeder B.



If an ant is committed to feeder A:

- The ant first checks the probability of transitioning to the uncommitted state based on lambda A.
  - If this happens, the ant becomes uncommitted.
- If the ant remains committed and is tandem running, then move closer to the food:
  - If food is present, it eats and stops tandem running.
- If the ant is still committed but not tandem running, it attempts to recruit another ant:
  - If a recruit is found, the ant recruits it with a probability of beta A and starts tandem running.
  - If no recruit is found, the ant moves randomly.

If an ant is committed to feeder B:

- The process is identical to the steps for feeder A, but with the respective parameters for feeder B (lambda B, beta B).

Ants use tandem running to recruit other ants to the stronger feeder, which demonstrates **collective decision-making**. The ODE model assumes that ants can be in one of three states: uncommitted, committed to feeder A, or committed to feeder B. Ants transition between these states based on discovery and recruitment rates, with recruitment occurring

when ants interact with others already committed to a feeder.

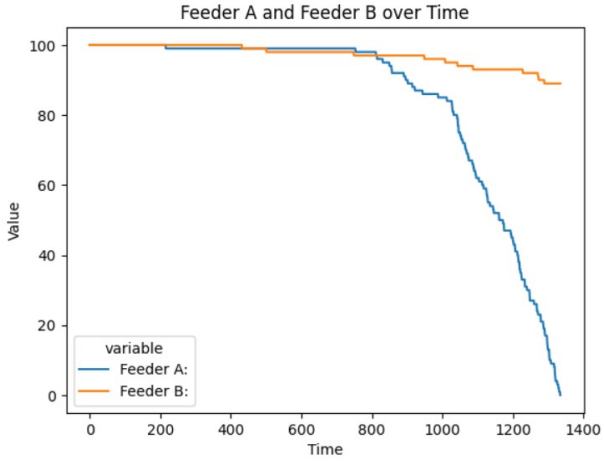
Key Agent Classes:

1. **Environment Agent**: Assigns unique identifiers to each grid cell.
2. **Nest Agent**: Represents the location of the nest on the grid.
3. **Food Agent**: Represents the two food sources (feeders) and tracks their food availability.
4. **Ant Agent**: The main agent, which has three states (uncommitted, committed to Feeder A, or Feeder B) and behaviors such as moving randomly, committing to a feeder, and performing tandem running. The ant's state transitions based on food discovery, recruitment, and attrition rates.

Ant Behavior and State Transitions:

- **Uncommitted State**: Ants explore their environment and commit to a feeder if they find one, based on discovery rates.
- **Committed States**: Once committed, ants either stay at the feeder or uncommit based on attrition rates. Ants also attempt to recruit others to their feeder and move towards it if they are recruiting.
- **Tandem Running**: Ants in the tandem running state guide other ants to the food source and return once the food is consumed.

The simulation runs until one of the feeders is depleted, which is expected to occur more quickly for the stronger feeder. As you can see below graph, feeder A is a strong feeder and over a period of time, feeder A depleted faster than weaker feeder B.



Summary:

This project focuses on converting an ODE model of ant foraging behavior into a computational agent-based model using the Mesa framework. It explores dynamics such as recruitment, food discovery, and state transitions in ants. The model simulates the collective foraging behavior of ants, including how individual ants interact with each other and the environment, mimicking the collective decision-making processes seen in ant colonies.

One key finding from the simulation is that the ants consistently prefer one feeder (feeder A) over the other, as predicted by the model. Over time, feeder A was depleted much faster than feeder B, aligning with the expected outcome. This result further demonstrates the utility of agent-based modeling in capturing the intricate dynamics of individual behaviors and their collective outcomes, which may not be easily captured by ODE models.

REFERENCES

The Negative selection algorithms, inspired by the immune system, provide an effective approach for anomaly detection in sequence data. Elberfeld and Textor's work [1][4] explores efficient training and linear-time classification of strings using negative selection, highlighting its potential in pattern recognition tasks. Hofmeyr and Forrest's "Immunity by Design" [2] emphasizes the design of artificial immune systems, further enriching anomaly detection methodologies. Textor's comparative study [3] evaluates negative selection in sequence data, underscoring its utility in identifying deviations from normal patterns. These insights draw from biological principles, such as immune system functions outlined by Sompayrac [5] and Gordon [6], emphasizing the complexity and organization in natural systems.

[1] M. Elberfeld and J. Textor "Negative selection algorithms on strings with efficient training and linear-time classification"

[2] Immunity by Design: An Artificial Immune System (Hofmeyr and Forrest)

[3] Textor, J. A Comparative Study of Negative Selection Based Anomaly Detection in Sequence Data. Universiteit Utrecht, Theoretical Biology & Bioinformatics.

[4] Elberfeld, M., & Textor, J. (2011). *Negative Selection Algorithms on Strings with Efficient Training and Linear-Time Classification*. Theoretical Computer Science, 412, 534-542.

[5] L. Sompayrac "How the Immune Systems Works", *Ch. 1: An Overview*

[6] Ch. 1: The Ant Colony as a Complex System; Ch 2: Colony Organization (Gordon)