Detection of Anomalous Behavior in a Robot System Based on Machine Learning

Mahfuzul Nissan

Sharmin Aktar

CSCI 6650 - Intelligent Agents

THE UNIVERSITY of NEW ORLEANS

Introduction & Motivation

- The growth of robotic systems in various industries has led to significant improvements in productivity and efficiency.
- Ensuring the safe and reliable operation of these systems is crucial to prevent potential disasters and ensure human well-being.
- Robotic systems can malfunction despite the best design and engineering practices, leading to potential safety risks.
- Researchers and engineers are working on developing new methods to improve the safety and reliability of robotic systems.
- Advanced control techniques, safety sensors, and machine learning algorithms are being used to detect and prevent potential anomalies in the system's behavior.

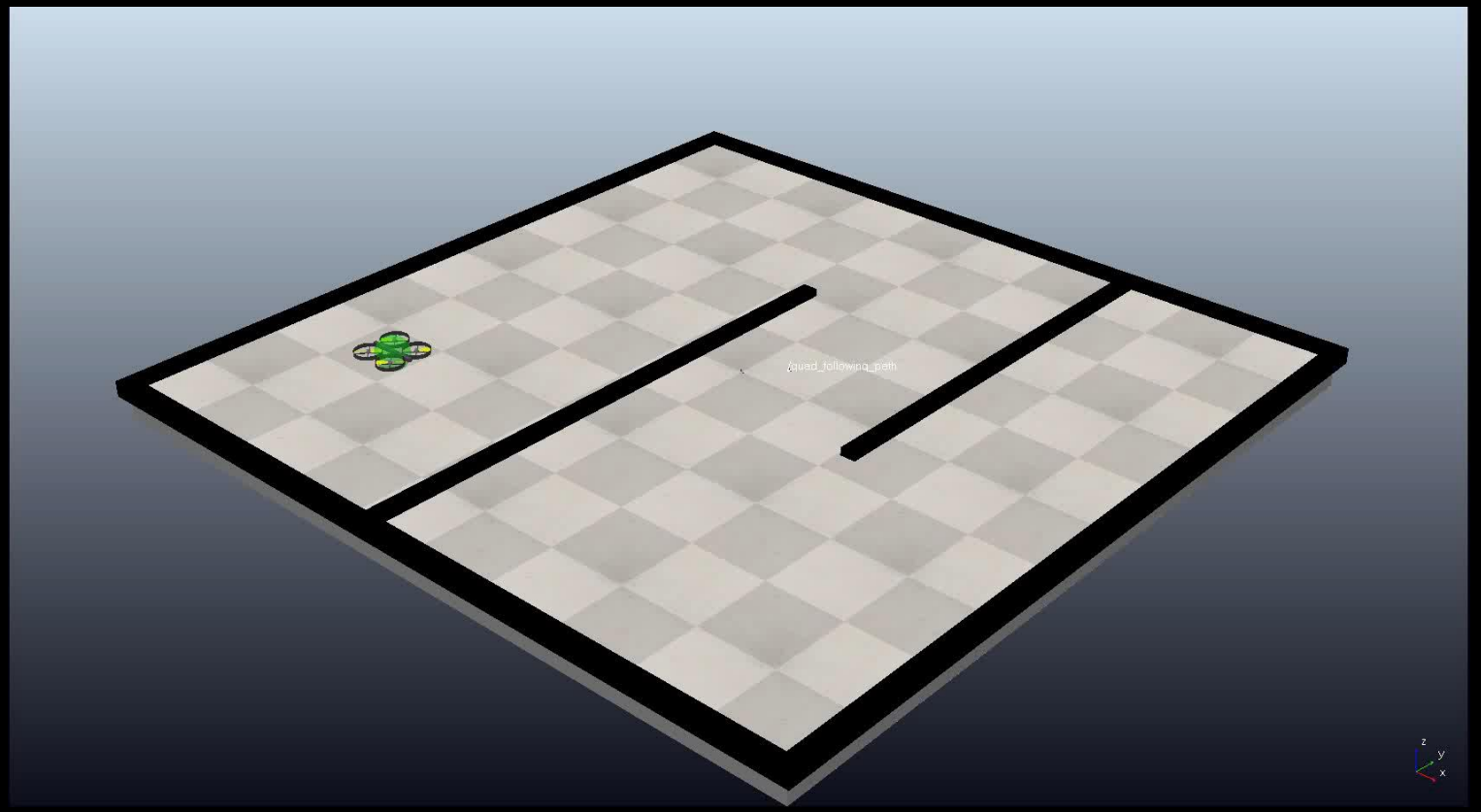
Objective

- Develop a Machine Learning based approach for detecting anomalous behavior in robotic system.
- Formulate the problem of anomaly detection in robotic systems as a binary classification problem using system logs.
- Create a context-specific data set for normal behavior in robotic systems, using a defined path planning algorithm.
- Create a context-specific data set for abnormal behavior in robotic systems, by introducing deviations from the normal behavior path planning algorithm.
- Evaluate the performance of our method for anomaly detection in robotic systems using the normal and abnormal behavior data sets.

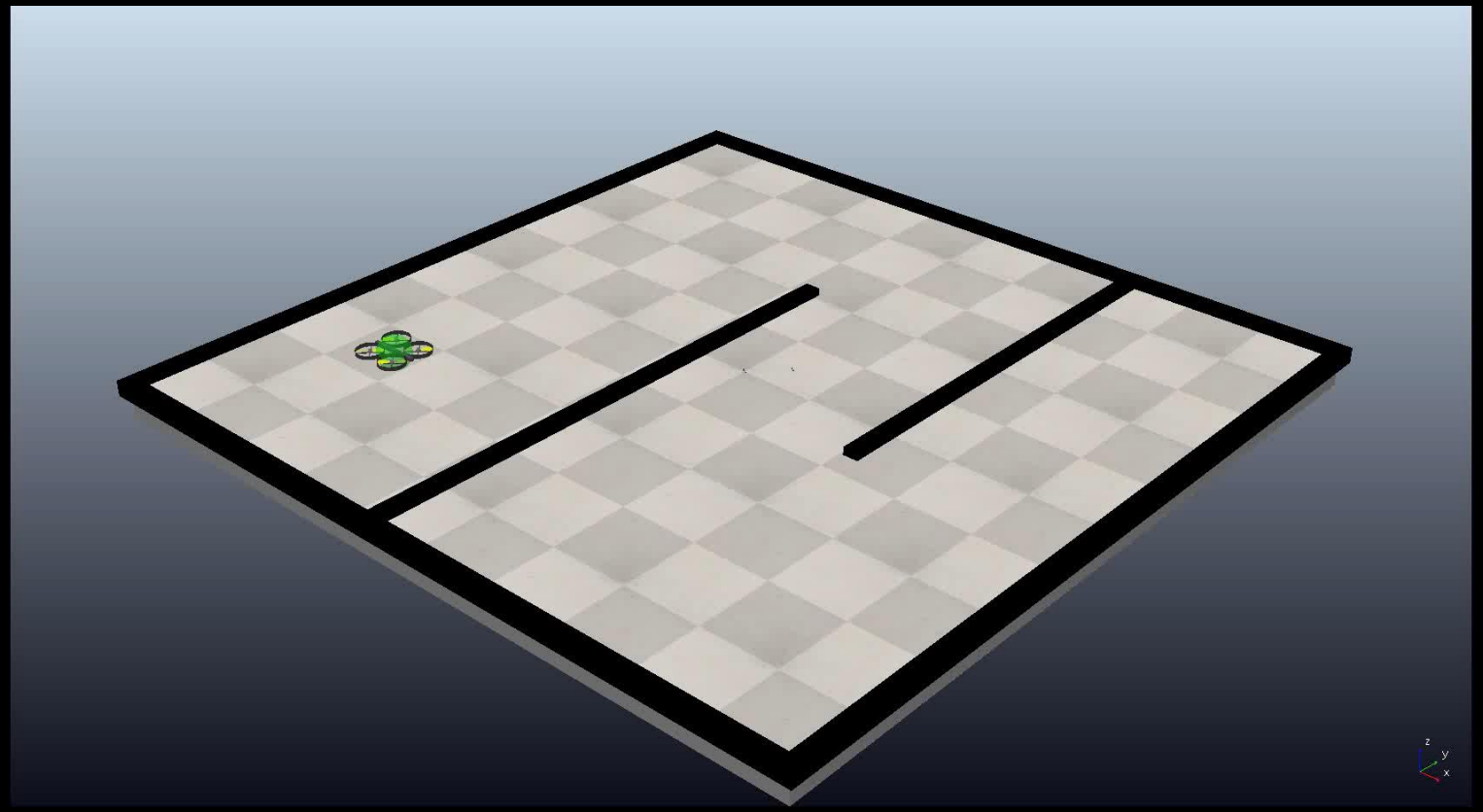
Problem Statement

- Our goal is to detect anomalous behavior in a robotic system using captured logs from simulations run in CoppeliaSim.
- We created two **context-specific** problems for our experiment:
- In the first context, a quadcopter follows a geometric path generated by the D* path planning algorithm. The normal scenario involves smooth and gradual movements, while the anomalous scenario introduces deviations in position and velocity.
- In the second context, two Pioneer robots follow the same path from different locations and meet in the middle while avoiding each other. The normal scenario involves the robots avoiding each other and following the path, while the anomalous scenario introduces deviations that result in a collision between the two robots.
- We logged data for each simulation run, including the time, position, orientation, velocity, acceleration, and angular velocity of the robots.
- We saved the logged data to a text file for further analysis and processing.

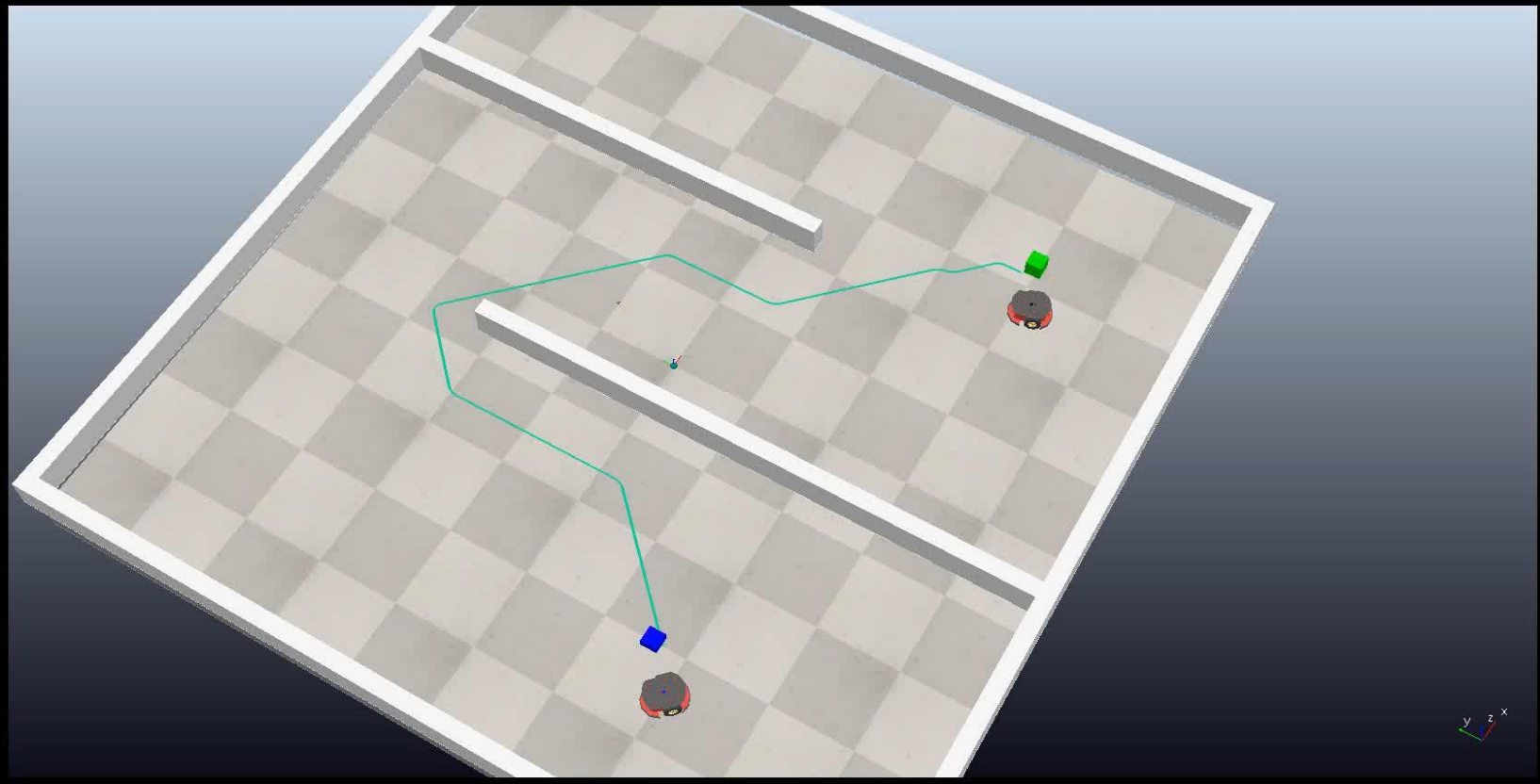
Normal Behavior-Quadcopter



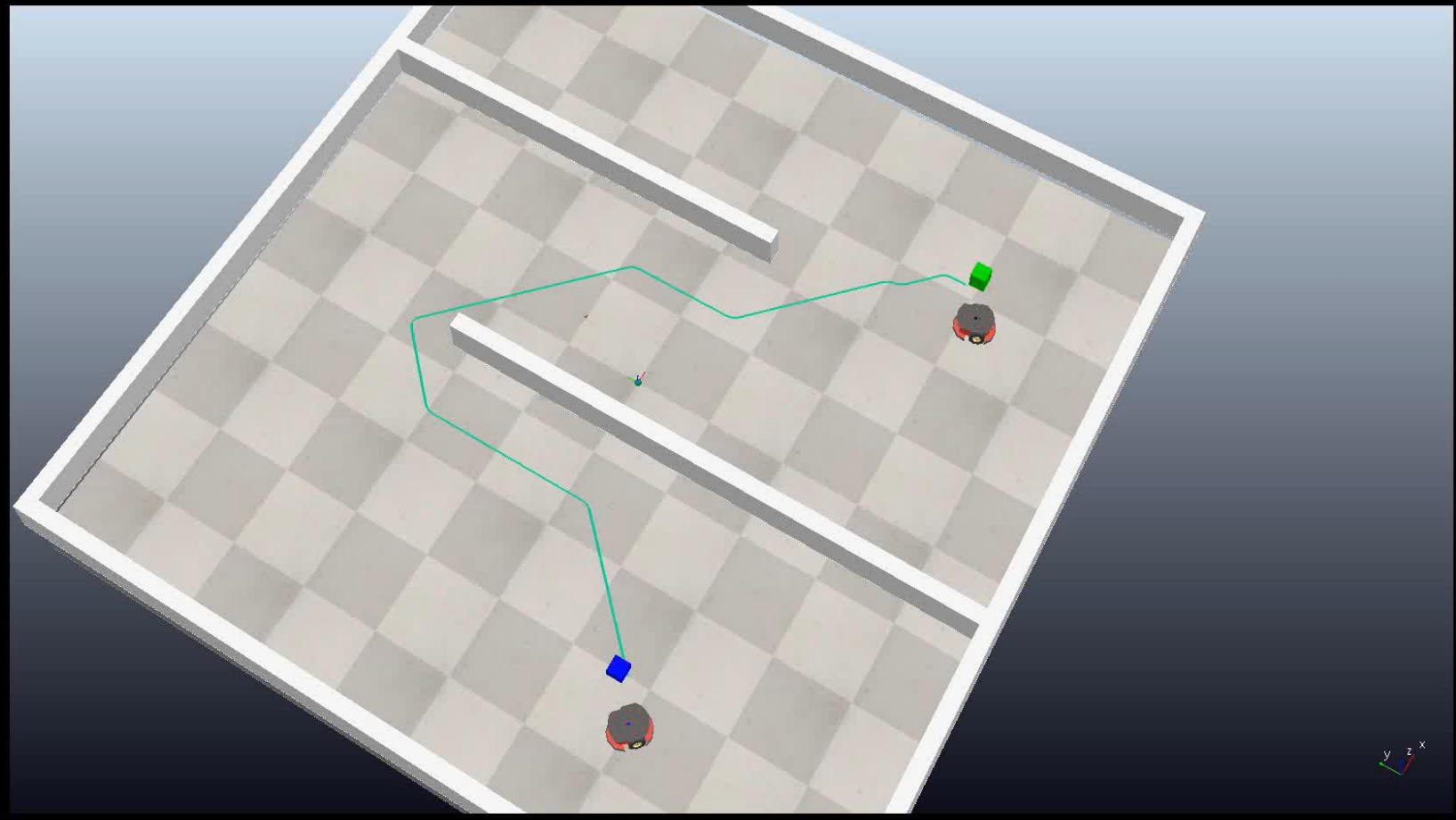
Anomalous Behavior-Quadcopter



Normal Behavior-Pioneer Robot



Anomalous Behavior-Pioneer Robot



Our Detection Methodology

- Load normal and anomalous logs from text files. Assign labels (0 for normal, 1 for anomalous) to the logs.
- Utilize separate text data for training and testing.
  - Training Data: Use logs with known normal and anomalous behavior.
  - Testing Data: Introduce logs with different goal positions to simulate unknown and unseen data.
- Supervised Machine Learning (SVM and Logistic Regression):
  - Train a SVM classifier with a linear kernel using the training data.
  - Train a logistic regression model using the training data.
  - Utilize labeled training data to learn patterns of normal and anomalous behavior.

Our Detection Methodology

- Predict and Evaluate (Supervised Models):
  - Apply the trained SVM classifier and logistic regression model to predict labels for the testing data.
  - Compute metrics such as accuracy, precision, recall, and F1-score to evaluate the models' performance.
- Semi-Supervised Deep Learning (Autoencoder):
  - Build an autoencoder model for anomaly detection using the training data.
  - Train the autoencoder to reconstruct normal instances accurately.
  - Set the threshold for anomaly detection based on the maximum reconstruction error from the training data (using only normal instances).
- Predict and Evaluate (Autoencoder):
  - Calculate the reconstruction error between the original and reconstructed instances for the testing data.
  - Identify anomalies based on the threshold.
  - Evaluate the performance of the autoencoder in detecting anomalies.

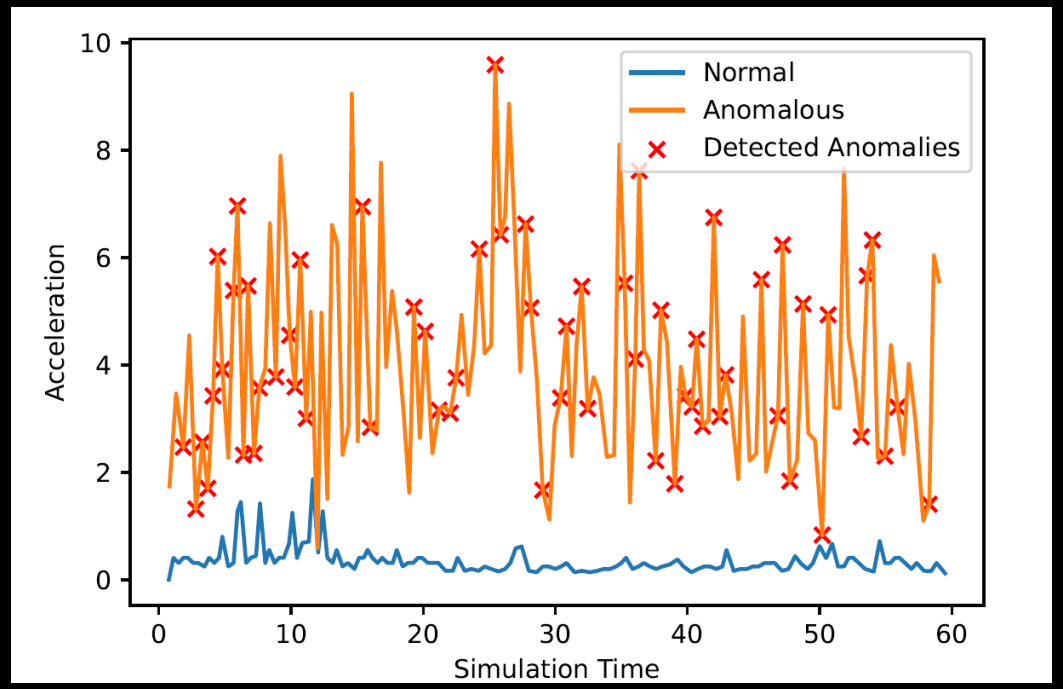
Experimental Results(Context 1)

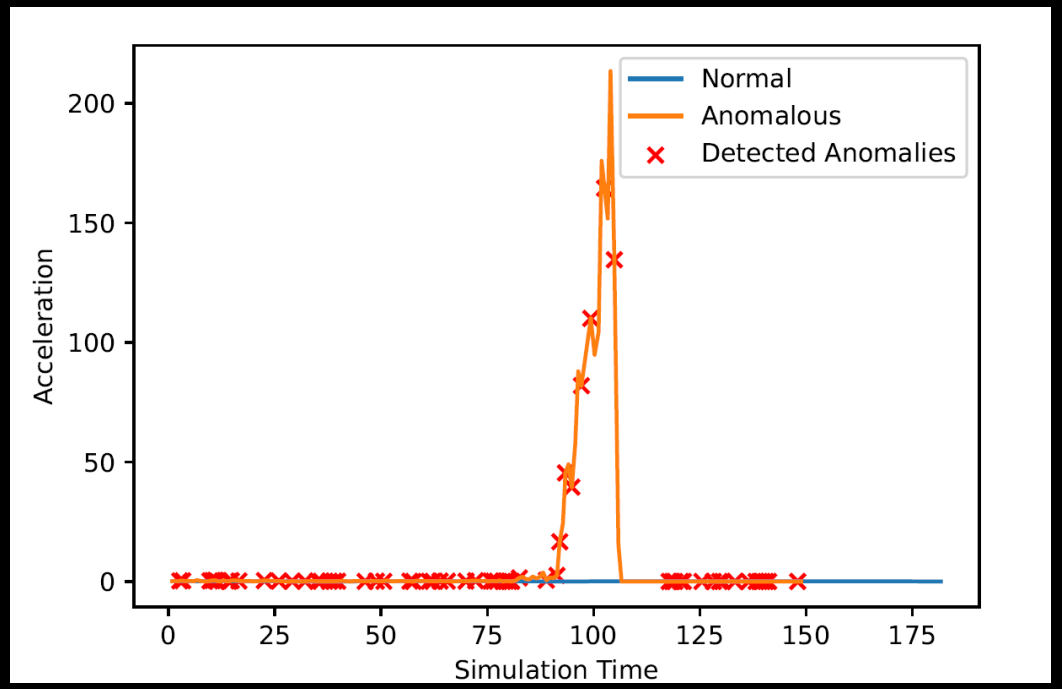
Metric	SVM	Logistic Regression
Average ROC score	0.8844	**0.9552**
Average precision	0.8888	0.9597
Average recall	0.8832	0.9562
Average Accuracy	0.8832	0.9562
Average F1-score	0.8830	0.9561

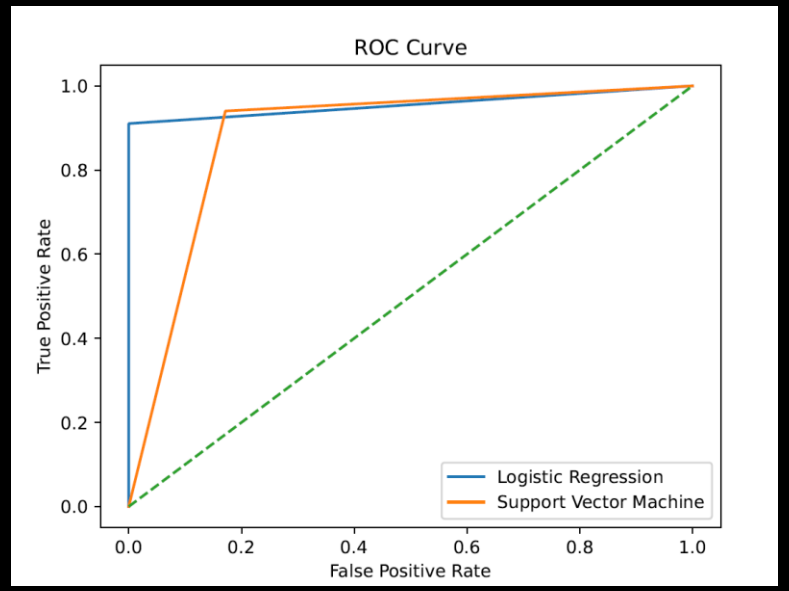
Experimental Results(Context 2)

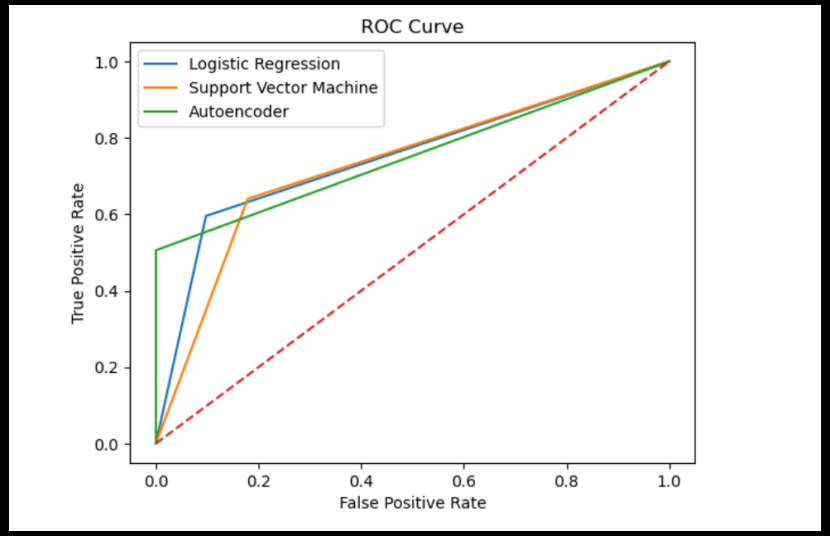
Metric	SVM	Logistic Regression	Autoencoder
Average ROC score	0.7308	0.7490	0.7675
Average precision	0.7435	0.7804	0.8488
Average recall	0.7453	0.7736	0.8038
Average Accuracy	0.7453	0.7736	0.8038
Average F1-score	0.7427	0.7660	0.7888

Context 1 & 2 Acceleration vs Simulation Time









Context 1 & 2 ROC Curve

Future Work

- Handle more complex logs: Currently, our system logs are assumed to be simple tuples of values. We plan to explore techniques to handle some complex logs.
- Improve the accuracy of the classifier: We plan to use different machine learning models and algorithms to improve the accuracy of the classifier.
- Evaluate the system's performance under different scenarios: We will try to evaluate the performance of our model under different scenarios, such as changes in the environment, changes in the system's parameters, or changes in the types of anomalies present in the data. This can help us to identify the strengths and limitations of the proposed system and guide further improvements.

Thank you