Extending MAPE-K with Data Augmentation to Mitigate Data Scarcity

Catherine Saunders, Roy Sterritt, Peter Nicholl, Ian McChesney

Ulster University

ce.saunders@ulster.ac.uk

Background

- Future space exploration will involve the development of autonomous swarms.
- Autonomic Computing aims to **solve the complexity of managing** a large swarm or system of components.

- The Autonomic **MAPE-K feedback loop** enables a system to self-manage.
  - MAPE-K - Monitor, Analyse, Plan, Execute, with Knowledge.
  - Deploying a space swarm with MAPE-K makes it more **autonomous**

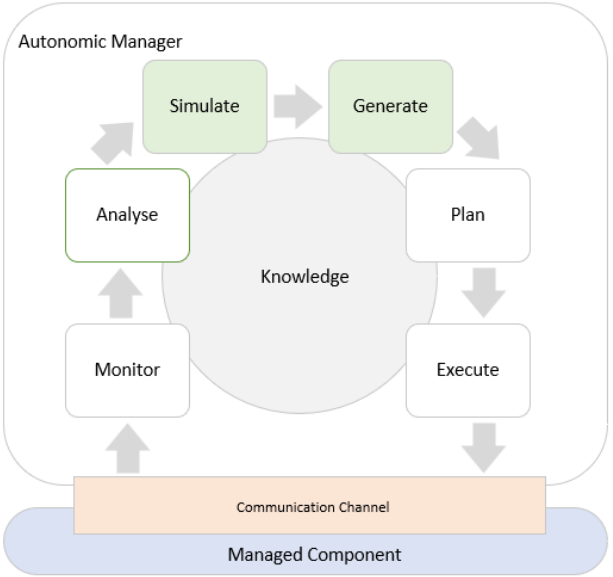
- **Data Scarcity Problem**
  - In remote swarm missions not all configurations of the swarm will be deployed
  - Consequence - the **dataset that is gathered may be incomplete.**
  - In hazardous conditions the dataset may have **missing data**
  - Swarm autonomic manager uses incomplete data to form decisions.

Previous Work – Data Generation in MAPE-K

- **Previous work**
  - Swarm simulation data used to train CTGAN model.
  - Adding a **Data Generation and Evaluation Pipeline** to MAPE-K.

- **Purpose**
  - Synthetic data could **increase the size of small datasets.**
  - Remote or hazardous conditions make it costly to collect. sufficient data.
  - More data **improves the reliability** of swarm manager decisions.

- **Solution**
  - Simulation and Generation phases added to form **MASGPE-K.**
  - Evaluation suite devised to compare synthetic data to original swarm data.



Generative Adversarial Networks (GANs)

- Recent ML technique to **increase dataset with synthetic data.**
  - Preserves the privacy of the original data, useful for medical studies.

- A GAN is a model consisting of 2 neural networks that compete against each other.
  - Generator produces data, Discriminator scores the data as real or fake.
  - It can produce Synthetic Data that is **statistically similar to the training dataset.**

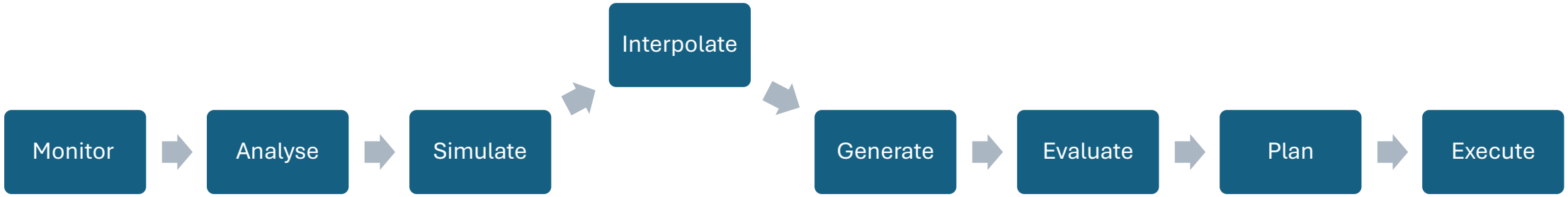
- Evaluating the Data Quality - important to determine its usefulness and faithfulness to the original data.

Why use Synthetic Data?

- Helps to **mitigate data scarcity issue** by increasing dataset size
  - Addresses **data scarcity** by expanding datasets.
  - More **cost effective** than deploying large swarms.
  - Helps fill gaps caused by unpredictable conditions.
  - **Enhances real data**, reducing reliance on extensive real-world collection.
  - Enables machine learning training when real data is insufficient.

- Enhance capabilities of Overseer spacecraft that manage the mission.
  - Equipping Overseer craft's Autonomic Manager with data generation so that it can **increase size of dataset**.
  - Synthetic data can be used as a proxy for the real data.
  - Train prediction algorithms on enhanced dataset.

Interpolation for Missing Swarm Configurations



- Proposed an **interpolation step** as an additional option before Generate within MASGPE-K.
- **Interpolate missing swarm configurations** in a dataset before generating synthetic data.
- Can interpolate missing data between known configuration using linear interpolation.
  - Configuration is a combination of various swarm parameters.
  - Sub-swarm robot types, sub-swarm size, swarm collaboration strategy, signal reach.
  - **Different configurations produce varying outcomes in terms or performance.**

Interpolation before Generation

- **Two numeric interpolation** approaches were tested and compared -
    1. **Direct Row Scaling** using linear interpolation.
    2. **Means Based** Interpolation using ‘Scipy’s’ Interp1d function.

- Training dataset - Swarm split into **2 sub-swarms**
    - 3 configurations exist in simulation dataset of 200 robots, these are -
        - Sub-swarm split evenly (50-50) – 2 robot types, 50% of each type.
        - Uneven split (70-30) – 70% of one type of robot, 30% of the other.
        - Extremely uneven split (90-10).

- New Configurations Interpolated before generation -
    - Interpolated new sub-swarm splits - **60-40 and 80-20.**

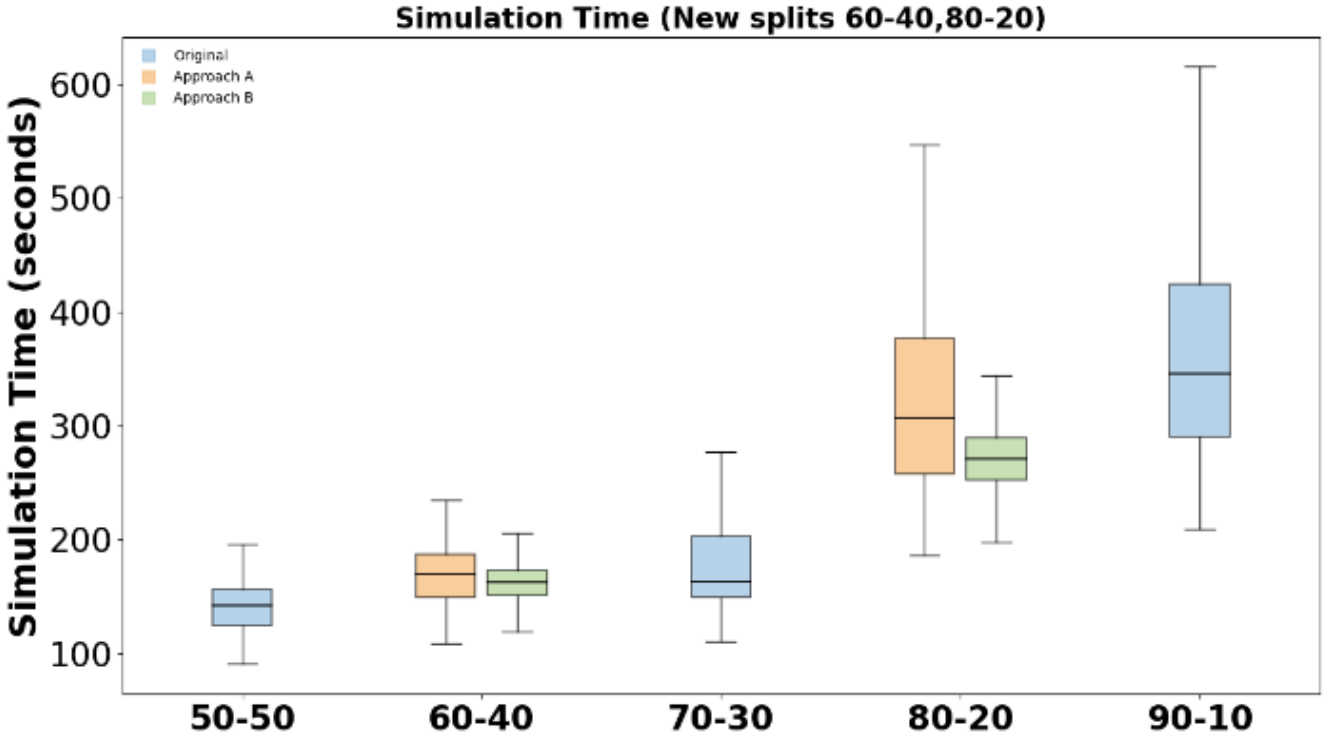
Interpolation Approaches

- **Direct Row Scaling**
  - Scaled each row in the nearest configuration to get new data.
  - 60-40 created by **multiplying 50-50** sub-swarm numeric data by **scaling factor.**
  - 80-20 used existing 90-10 configuration for scaling.

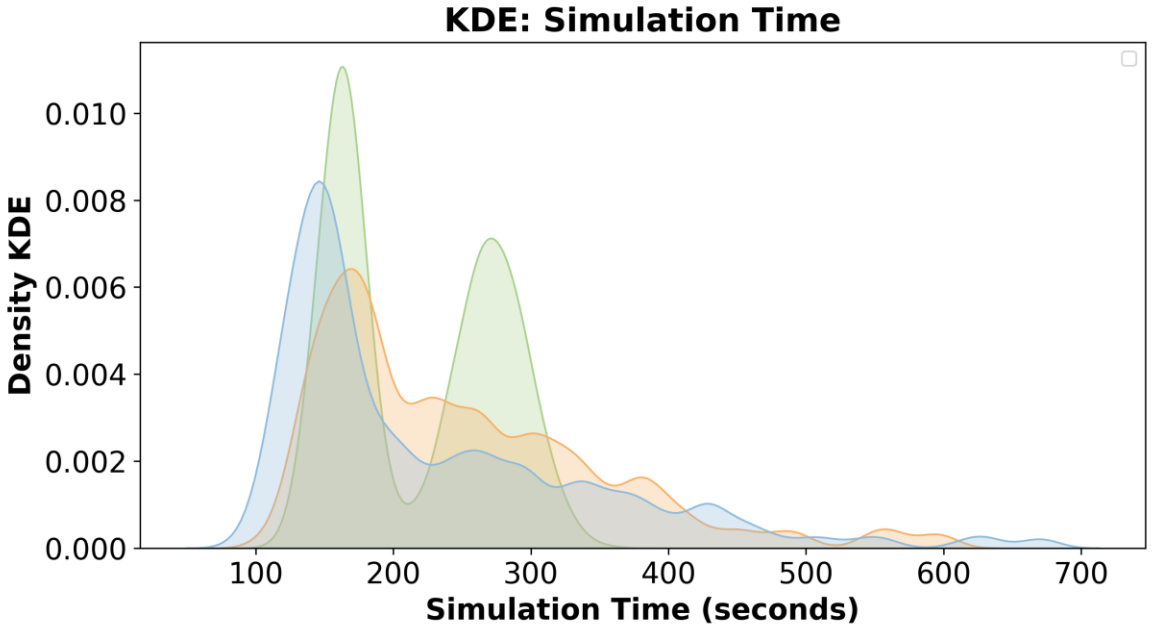
- **Means Based Interpolation**
  - Scipy ‘Interp1d’ function to interpolate **from the mean of existing configurations**.
  - 10% noise added to introduce variability around the mean.
  - **Reference anchor points** used – 70-30 and 90-10 used to create 80-20.
    - 50-50 and 70-30 configurations used as anchors for 60-40.

Evaluation of Interpolated Data

- Boxplot for *'Simulation Time'* feature showed that Approach A **Direct Row Scaling produced data more consistent** with the existing configurations.



Distribution Comparison

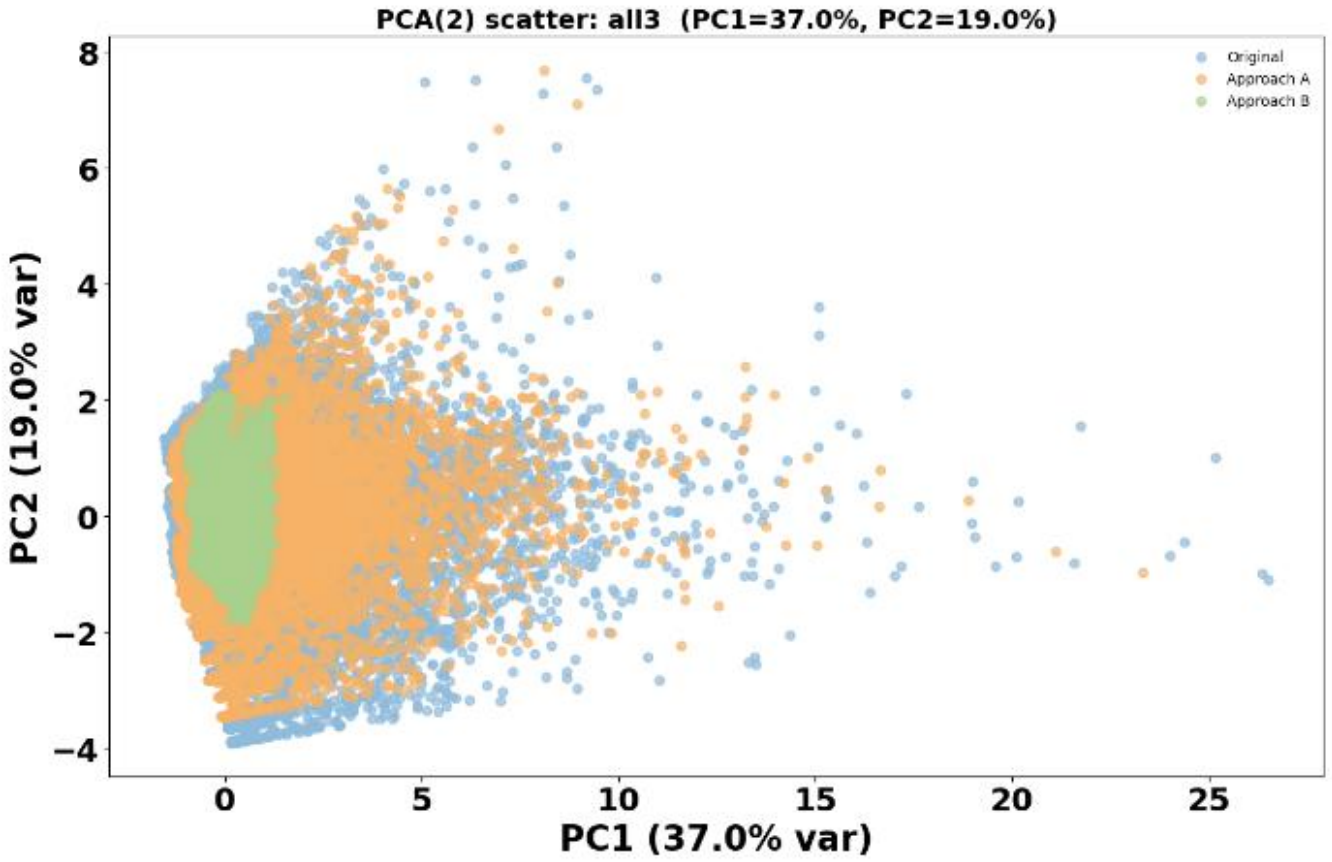


- The KDE plot shows the distribution for Simulation Time.
  - Simulation Time - simulation running time before all items were found.

- Direct Row Scaling (orange) vs Means Based (green), existing configurations (blue)
- **Direct Row Scaling data follows the shape of the existing configurations** and falls within extremes.
- **Means Based** is concentrated in two peaks and narrower time ranges, **didn't capture the pattern**.

PCA Comparison

- **Simulation** dataset (blue cloud)
- **Direct Row Scaling** (**orange** cloud)
  - **Accurately replicated** the structure and patterns of the original simulation data.
- **Means Based** interpolation (**green** cluster)
  - Data points are concentrated in a small area indicating that it **failed to capture** the range and variability.



CTGAN Training on Interpolated Dataset

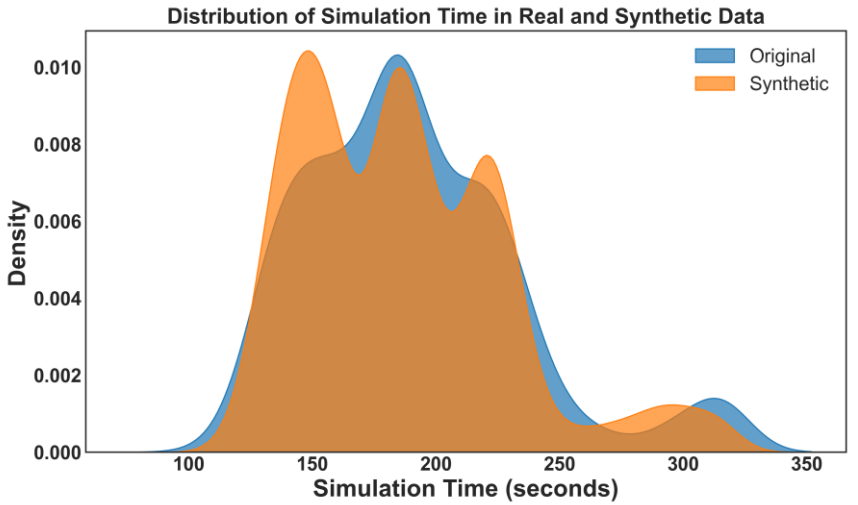
- Approach A Direct Row Scaling selected for training CTGAN model.

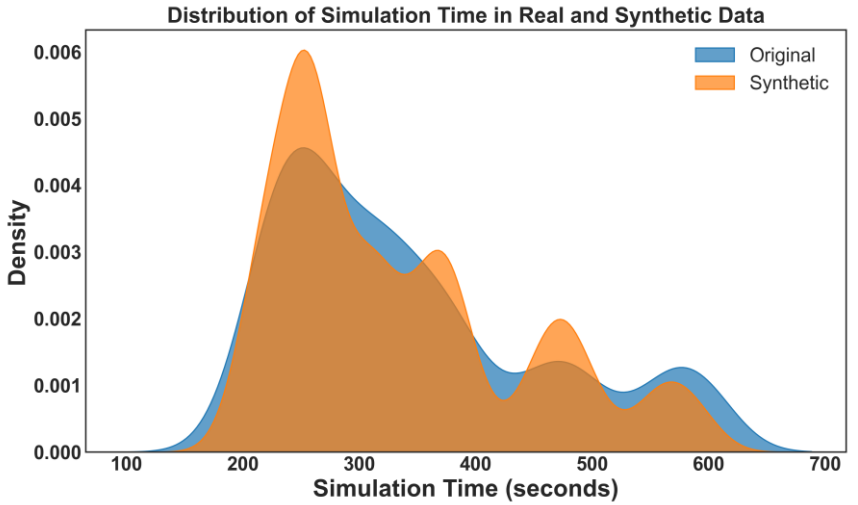
- **Two CTGAN models were trained**, one per new swarm configuration.
  - CTGAN 1 for 60-40.
  - CTGAN 2 for 80-20.
  - Splitting the training allowed the models to learn the nuances of each configuration.

- Training parameters –
  - **2000 epochs, batch size of 500**.
  - Provides an acceptable tradeoff between performance and training times.

KDE Evaluation of Interpolated & Generated Data

- The **interpolated data was created using Approach A Direct Row Scaling**.
- Interpolated data was used to **train the CTGAN models**.
- CTGAN models used to generate new synthetic data for the configurations 60-40 and 80-20.
- The KDE plots show the comparison of the new 60-40 and 80-20 synthetic data to the interpolated test dataset.
- The plots show that the **CTGAN learned the shape of the distribution.**





Conclusion and Future Work

- Direct Row Scaling approach outperformed Means Based interpolation
  - It preserved the shape and distribution of the data.
  - Good quality data in terms of numerical distribution.

- Data-scarce Environments – having interpolated data for missing configurations of the swarm could prove useful for analysis and planning.

- Numeric interpolation followed by generative modelling is a viable technique for augmenting datasets with missing data.

- Future Work – compare the usefulness of the interpolation and generation on planning performance.
  - Interpolated synthetic data vs Incomplete dataset