# REPORT

Worked on generating a synthetic sensor dataset using an Autoregressive (AR) model. The reason for using an AR model is based on a research paper that describes how sensor errors, such as bias and scale factor, can be effectively modeled and simulated using autoregressive processes. The paper shows that this approach realistically replicates the time-correlated noise and drift found in real inertial sensors.

The specific research paper used as a reference is:  
"A Rigorous Temperature-Dependent Stochastic Modelling and Testing for MEMS-Based Inertial Sensor Errors" which details how AR models can generate realistic sensor data for simulation and testing.

Using these base values (find those in next pages) and the AR modeling approach from the paper, generated a large synthetic dataset that mimics the behaviour of real sensor data with realistic errors and correlations over time.

After generating the data, trained two different neural networks to compare their performance. I trained a Multi-Layer Perceptron (MLP) and a Long Short-Term Memory (LSTM) network on the same synthetic dataset.

The results showed a clear difference:  
MLP RMSE: 0.009604, MAE: 0.005664  
LSTM RMSE: 0.004096, MAE: 0.002525

The LSTM showed 57.35% improvement in RMSE and 55.42% improvement in MAE over the MLP. This significant improvement makes sense because the LSTM is designed to handle sequential data and time-based patterns, which matches the autoregressive nature of the synthetic data we generated.

The next step is to apply this approach to real lab data. Since the real data doesn't have timestamps, will assume the data points are in chronological order based on their sequence in the file, will use Gaussian Process regression to create a smooth model of the sparse lab data, then generate a dense synthetic dataset from it. For constant parameters like bias and scale factor, we will use their average values from the lab data.

This approach allows to create a large, realistic training dataset from limited lab measurements, which we can then use to train models for new sensors through transfer learning.

**Synthetic Dataset Generation**

**Dataset Characteristics:**

* **Temperatures**: -40°C to +60°C in 20°C increments
* **Samples**: 100,000+ samples per temperature point
* **Parameters included**: Raw accel/gyro outputs, true values, bias components, scale factors, and temperature data
* **Sampling rate**: 200 Hz (matching ADIS16364 specifications)

**Implementation Details:**  
The data generation followed the ADIS16364 IMU specifications:

* Gyro initial bias error: ±3%
* Gyro in-run bias stability: 0.007%
* Gyro bias temperature coefficient: ±0.01%/°C
* Accelerometer initial bias error: ±8 mg
* Accelerometer in-run bias stability: 0.1 mg

**Neural Network Architecture**

**MLP Model:**

* Input layer: Flattened sequence data
* Hidden layers: 256-128-64-32 neurons with ReLU activation
* Output layer: 6 nodes (bias and scale factor for each axis)
* Regularization: Dropout (0.3) between layers

**LSTM Model:**

* Input shape: (sequence\_length, 7 features)
* LSTM layers: 128 and 64 units with dropout (0.3)
* Dense layer: 32 neurons with ReLU activation
* Output layer: 6 nodes for error parameters

**Training Parameters:**

* Optimizer: Adam (learning rate: 0.001)
* Loss function: Mean Squared Error (MSE)
* Metrics: Mean Absolute Error (MAE)
* Validation split: 20%
* Early stopping: Patience of 5 epochs

summarizes a comprehensive study on MEMS IMU error modeling using Autoregressive-based Gauss-Markov processes and deep learning techniques for sensor calibration. The project follows the methodology established in the foundational paper "A Rigorous Temperature-Dependent Stochastic Modelling and Testing for MEMS-Based Inertial Sensor Errors" by El-Diasty and Pagiatakis (2009).

**Why AR-GM?**  
The AR-Based Gauss-Markov approach was selected because it:

1. **Preserves physical fidelity**: Maintains the statistical properties of real sensor errors
2. **Provides efficient parameter estimation**: Works with shorter datasets than traditional methods
3. **Enables temperature dependency modeling**: Naturally extends to varying environmental conditions
4. **Offers mathematical tractability**: Well-defined properties and efficient simulation algorithms
5. **Ensures compatibility with navigation filters**: Direct mapping to state-space models