

FINAL 8-WEEK REPORT



Science Academies Summer Research Fellowship Program 2025

AI-assisted, Deep Learning-based Digital Transformation for Drift Quantification in Thermocouples

By:

Haran Perumal S L - ENGS2799



SASTRA Deemed University, Thanjavur, Tamil Nadu



CSIR-National Physical Laboratory, New Delhi

A Report Submitted To
Indian Academy of Sciences, Bengaluru, Karnataka

Acknowledgements

I would like to express my heartfelt gratitude to the Council of Scientific and Industrial Research – National Physical Laboratory (CSIR–NPL), New Delhi, for the opportunity to undertake this 8-week research internship and for providing a stimulating environment that has significantly contributed to my learning and professional growth.

This report marks the completion of the first four weeks of the internship, during which I have had the privilege to work under the guidance of Dr. Dilip Dhondiram Shivagan, Senior Principal Scientist, Head, Temperature and Humidity Metrology, and Dr. Komal Bapna, Senior Scientist, Temperature and Humidity Metrology whose continuous support, timely feedback, and insightful technical discussions have been immensely valuable in shaping the early direction of this project.

I am thankful to Mr. Gaurav Gupta, Mr. Hansraj Meena, Dr. Ashish Bhatt, Mr. Anshuman Yadav, Mr. Pavan, Mr. Sumit and the members of the Temperature and Humidity Standards Laboratory for their assistance with instrumentation, technical clarifications, and logistical support throughout this phase of the project. Their expertise and willingness to help have enabled me to confidently work with devices like the Fluke 1529E and 1620A, as well as the experimental facilities for thermocouple drift evaluation.

I wish to acknowledge the encouragement and support from my peers Dhruv Dua, Siddharth Singh, and Adithiya Yesade, family, and faculty at SASTRA Deemed University, without which this endeavor would not have been possible.

I am deeply grateful to the Indian Academy of Sciences and CSIR–NPL for giving me the opportunity to work on cutting-edge technologies and push their boundaries.

Haran Perumal S L

Abstract—High-accuracy thermometric calibration relies heavily on manual processes and periodic checks, often lacking real-time analytics or drift prediction capabilities. Precision instruments like the Fluke 1529E standards thermometer and the Fluke 1620A thermo-hygrometer, while capable of high-fidelity measurements, were previously limited at NPL by manual logging, poor real-time visibility, and a lack of automated cycle detection or drift analysis. Moreover, noble-metal thermocouples like Type S exhibit complex, nonlinear drift at elevated temperatures ($>1000^{\circ}\text{C}$), which, if unmonitored, can significantly compromise measurement traceability, increase recalibration frequency, and reduce trust in long-duration thermal experiments.

This 8-week internship project at CSIR–NPL focused on advancing metrology instrumentation through comprehensive digitization and pioneering thermocouple drift analysis. Building upon an initial 4-week foundation of digitizing the Fluke 1620A hygrometer and Fluke 1529E thermometer, the extended period saw the successful development and implementation of a robust cold junction compensation methodology, which critically refined temperature measurements from a Type S thermocouple. High-resolution EMF data from ten full thermal cycles and a baseline run were acquired using the digitized Fluke 1529E system. This extensive dataset enabled a detailed, cycle-wise analysis of thermocouple drift, comparing both raw and compensated data against baseline performance. Significant systematic drift observed in uncorrected data was effectively mitigated by the cold junction compensation, demonstrating its vital role in improving measurement accuracy. Preliminary drift patterns were meticulously examined, revealing subtle changes and providing a richer foundation for future predictive modeling. The report details the enhanced experimental design, the sophisticated data processing pipeline including the newly integrated cold junction compensation, and an in-depth discussion of the results observed across the ten thermal cycles. Furthermore, it outlines a refined roadmap for developing advanced machine learning models, such as FFN networks, CNN etc., to predict thermocouple drift and enable AI-assisted calibration strategies. This work represents a crucial step towards creating self-aware instruments that can autonomously monitor their health and calibration status, thereby enhancing the precision, reliability, and efficiency of high-temperature metrology.

Keywords—*Digital metrology, thermocouple drift, Fluke 1529E, Fluke 1620A, real-time logging, heat index, Python GUI, SCPI, calibration automation*

Contents

1	Digitization of the Fluke 1620A Thermo-Hygrometer	6
1.1	Introduction	6
1.2	System Design	6
1.3	Frontend Development	7
1.4	Challenges and Optimization	8
1.5	Impact and Value	8
2	Digitization of the Fluke 1529E Standards Thermometer	8
2.1	Context and Goals	8
2.2	Architecture and Communication	9
2.3	Temperature Conversion Algorithms	9
2.3.0.1	PRT Conversion (ITS-90)	9
2.3.0.2	Type S Conversion	10
2.4	Visualization and Export	10
2.5	Results and Benefits	10
3	Advanced Drift Characterization in Type S Thermocouples	10
3.1	Motivation and Stability Metrics	10
3.2	Experimental Methodology and Cycle Protocol	11
3.3	Cold Junction Compensation: Theory and Implementation	18
3.4	Data Acquisition Architecture and Logging Strategy	19
4	Quantitative Drift Analysis and Hysteresis Modeling	20
4.1	Cycle-wise Drift Trends and Plateau Alignment	20
4.1.0.1	Summary:	26
4.2	Certificate vs NIST: Conversion Method Comparisons	26
4.3	Hysteresis Analysis: Multi-cycle Interpretation	27
4.4	Thermal Equilibrium Criteria and Dwell Behavior	33
5	Machine Learning Pipeline for Drift Prediction	34
5.1	Feature Engineering	34
5.1.1	Temperature Plateau Detection and Extraction	34
5.1.2	EMF Interpolation and Temperature Drift Calculation	35
5.2	Model Selection: Baselines, Regressors, and FFNNs	35
5.2.1	Baseline and Linear Models	35
5.2.2	Ensemble Models: Random Forest and XGBoost	36
5.2.3	Feedforward Neural Network (FFNN)	36
5.2.4	Model Comparison and Interpretation	36
5.3	Evaluation Metrics: MAE, MSE, Residual Spread	37
5.3.1	Mean Absolute Error (MAE)	37
5.3.2	Mean Squared Error (MSE)	37
5.3.3	Residual Spread and Visualization	37
5.3.4	Comparative Interpretation	38
5.4	SHAP Interpretation and Model Validity Limits	38
6	Implications for Sensor Integrity and AI Deployment	41
6.1	Implications of Near-Zero Drift in Noble-Metal TCs	41
6.2	Comparison with Base-Metal TCs (e.g., Type K/N)	42
6.3	Digital Twin Architecture for Predictive Calibration	42

6.4	Role of Drift Forecasting in National Metrology Labs	43
7	Conclusion and Future Scope	43
7.1	Summary of Achievements Across 8 Weeks	43
7.2	Next Steps: Data Expansion, Type K Modeling, LSTM Forecasting	44

List of Figures

1	Overview of the Developed GUI Application for Fluke 1529E	9
2	Overlay of Temperature Profiles from All Ten Thermal Cycles (Channel 3).	12
3	Temperature Profile for the Baseline Measurement (Channel 3)	12
4	Temperature Profile for the Baseline Measurement (Cycle 1)	13
5	Temperature Profile for the Baseline Measurement (Cycle 2)	13
6	Temperature Profile for the Baseline Measurement (Cycle 3)	14
7	Temperature Profile for the Baseline Measurement (Cycle 4)	14
8	Temperature Profile for the Baseline Measurement (Cycle 5)	15
9	Temperature Profile for the Baseline Measurement (Cycle 6)	15
10	Temperature Profile for the Baseline Measurement (Cycle 7)	16
11	Temperature Profile for the Baseline Measurement (Cycle 8)	16
12	Temperature Profile for the Baseline Measurement (Cycle 9)	17
13	Temperature Profile for the Baseline Measurement (Cycle 10)	17
14	Cycle 1 Drift vs Baseline	20
15	Cycle 2 Drift vs Baseline	21
16	Cycle 3 Drift vs Baseline	21
17	Cycle 4 Drift vs Baseline	22
18	Cycle 5 Drift vs Baseline	22
19	Cycle 6 Drift vs Baseline	23
20	Cycle 7 Drift vs Baseline	23
21	Cycle 8 Drift vs Baseline	24
22	Cycle 9 Drift vs Baseline	24
23	Cycle 10 Drift vs Baseline	25
24	Mean Drift Across 10 Cycles	25
25	Standard Deviation of Drift Values per Cycle	26
26	Hysteresis Curve – Channel 3 – Baseline	28
27	Hysteresis Curve – Channel 3 – Cycle 1	28
28	Hysteresis Curve – Channel 3 – Cycle 2	29
29	Hysteresis Curve – Channel 3 – Cycle 3	29
30	Hysteresis Curve – Channel 3 – Cycle 4	30
31	Hysteresis Curve – Channel 3 – Cycle 5	30
32	Hysteresis Curve – Channel 3 – Cycle 6	31
33	Hysteresis Curve – Channel 3 – Cycle 7	31
34	Hysteresis Curve – Channel 3 – Cycle 8	32
35	Hysteresis Curve – Channel 3 – Cycle 9	32
36	Hysteresis Curve – Channel 3 – Cycle 10	33
37	SHAP Mean Value Plot - Random Forest	39
38	Random Forest Feature Importances	39
39	SHAP Summary Plot - Random Forest	40
40	SHAP Summary Plot - XGBoost	40

1. Digitization of the Fluke 1620A Thermo-Hygrometer

1.1. Introduction

The Fluke 1620A "DewK" thermo-hygrometer is a precision instrument widely used for monitoring laboratory temperature and humidity, critical in calibration environments. Despite its high-accuracy sensor performance and SCPI-based communication capabilities, the standard usage at NPL involved manual readings or basic data logging via third-party tools with limited flexibility. The need to automate this process—acquiring data with precision, visualizing it in real time, and ensuring fault-tolerant logging with derived computations—formed the first leg of the digitization task.

The goal was to replace passive and manual data extraction with a dynamic and intelligent system capable of continuous operation over days or even weeks. Furthermore, incorporating domain-specific calculations like the heat index—relevant when high humidity intersects with elevated temperatures—was seen as a value addition to basic logging.

1.2. System Design

To achieve these goals, a modern web-based architecture was selected. This allowed the separation of back-end logic (responsible for hardware communication and data processing) from the frontend visualization and user interaction, making the system modular and easily extensible.

The backend was implemented in Python using the Flask microframework, with Flask-SocketIO enabling real-time bi-directional communication between the browser and the server. The RS-232 serial port was interfaced using the pySerial library, and each second, the hygrometer's output was read, parsed, and sent to the front end via sockets.

The hygrometer outputs a comma-separated string every second containing timestamps, two temperature readings, and two humidity values:

```
2025-07-04 14:25:31,24.63,59.1,24.56,58.9
```

These were parsed in a dedicated background thread. The design ensured robustness by incorporating timeouts, non-blocking reads, and exception handling to avoid crashes due to noise, corrupted serial data, or accidental disconnection.

A critical part of this processing pipeline was the computation of the heat index for each sensor. Using the NOAA Rothfusz regression formula, which models human-perceived temperature based on ambient temperature and relative humidity, the system computed the apparent temperature in real time. The formula is:

$$HI = c_1 + c_2T + c_3RH + c_4T \cdot RH + c_5T^2 + c_6RH^2 + c_7T^2 \cdot RH + c_8T \cdot RH^2 + c_9T^2 \cdot RH^2, \quad (1)$$

where T is the ambient temperature in Fahrenheit, RH is the relative humidity (%), and c_1 through c_9 are empirical coefficients defined by NOAA.

To apply this in Python, the temperatures were first converted:

```
1 def c_to_f(temp_c):
2     return (temp_c * 9/5) + 32
```

Code 1. Celsius to Fahrenheit conversion

The regression was implemented as:

```
1 def compute_heat_index(t_celsius, rh):
2     t = c_to_f(t_celsius)
3     hi_f = (c1 + c2*t + c3*rh + c4*t*rh + c5*t**2 +
4             c6*rh**2 + c7*t**2*rh + c8*t*rh**2 + c9*t**2*rh**2)
5     return (hi_f - 32) * 5/9 # Convert back to Celsius
```

Code 2. Heat Index computation

Beyond real-time processing, the backend maintained a growing buffer of all recent readings, which were periodically saved to disk. The choice of Excel (XLSX) format allowed easy analysis using commonly available tools. Auto-saving was based on either the number of samples or elapsed time, ensuring no data was lost in case of unexpected shutdown.

1.3. Frontend Development

The front end consisted of an HTML5 single-page application served by Flask. Using modern JavaScript techniques (primarily with libraries like Chart.js for plotting), the front end displayed numeric values and dynamic line charts of each parameter.

The interface featured a configuration modal that let users choose the serial port, adjust the polling frequency, and set autosave thresholds. The data visualization panel is updated in real-time via WebSockets, ensuring low-latency display.

Error messages surfaced in the UI, and the user could initiate or stop logging as needed. Additional features include time calibration of the device through SCPI commands and on-demand exporting of data or chart snapshots.

```
1 const ctx = document.getElementById('temp chart').getContext('2d');
2 const tempChart = new Chart(ctx, {
3     type: 'line',
4     data: {
5         labels: [],
6         datasets: [{
7             label: 'Temperature (C)',
8             data: [],
9             borderColor: 'rgba(255,99,132,1)',
10            fill: false
11        }]
12    },
13    options: {
14        animation: false,
15        scales: {
16            x: { type: 'time', time: { unit: 'second' } },
17            y: { suggestedMin: 0, suggestedMax: 50 }
18        }
19    }
20 });
21
22 socket.on('new_data', (data) => {
23     const now = new Date();
24     tempChart.data.labels.push(now);
25     tempChart.data.datasets[0].data.push(data.temperature);
26     tempChart.update();
27 });


```

Code 3. Real-Time Chart Update with Chart.js and Socket.IO

```
1 function applySettings() {
2     const port = document.getElementById('comPortSelect').value;
3     const baud = document.getElementById('baudRateInput').value;
4     socket.emit('config_serial', { port: port, baud: parseInt(baud) });
5 }
```

Code 4. Serial Port Configuration Modal JS Handler

```

1 function startLogging() {
2   socket.emit('start_logging');
3   document.getElementById('startBtn').disabled = true;
4   document.getElementById('stopBtn').disabled = false;
5 }
6
7 function stopLogging() {
8   socket.emit('stop_logging');
9   document.getElementById('startBtn').disabled = false;
10  document.getElementById('stopBtn').disabled = true;
11 }

```

Code 5. Start/Stop Logging Buttons JS Handler

The resulting interface required no installations beyond a standard browser and could be hosted locally or over a LAN, making it accessible across multiple terminals within the lab.

1.4. Challenges and Optimization

Several practical issues were addressed during implementation. Serial communication required debouncing to avoid duplicate readings. Moreover, care was taken to ensure floating-point rounding errors did not propagate into temperature or humidity calculations, especially during heat index computation.

Another key challenge was ensuring that the heat index formula didn't produce NaNs or infinities for borderline humidity or temperature values. This was handled by bounding values and including fallback logic for certain corner cases, such as extremely low humidity. Autosave logic also required sophistication: writing data to disk in a manner that was thread-safe, fault-tolerant, and allowed appending to existing files without data corruption.

In the web layer, managing real-time graphs for four parameters simultaneously (temperature and humidity from two channels) required efficient use of browser memory and time-series data management. Circular buffers and lightweight redraw strategies were implemented to keep performance high even during long-duration operations.

1.5. Impact and Value

The final system replaced hours of manual recording with a zero-supervision, high-resolution data logger that operates continuously. It enabled researchers to monitor environmental drift during calibration procedures without requiring manual notes or delayed analysis.

Additionally, the modular architecture supports future upgrades, such as:

- Integration with central monitoring dashboards across multiple rooms
- Email/SMS alerts on humidity/temperature threshold crossings
- Logging to cloud-based databases for long-term trend analysis
- Seamless handoff to calibration databases at the end of test campaigns

Thus, this component of the project achieved a substantial uplift in both productivity and reliability for NPL's daily operations.

2. Digitization of the Fluke 1529E Standards Thermometer

2.1. Context and Goals

The Fluke 1529E thermometer is a high-precision instrument capable of reading from both resistance temperature detectors (RTDs) and thermocouples. At NPL, it is central to temperature calibration procedures, particularly in standardizing PRTs and thermocouples.

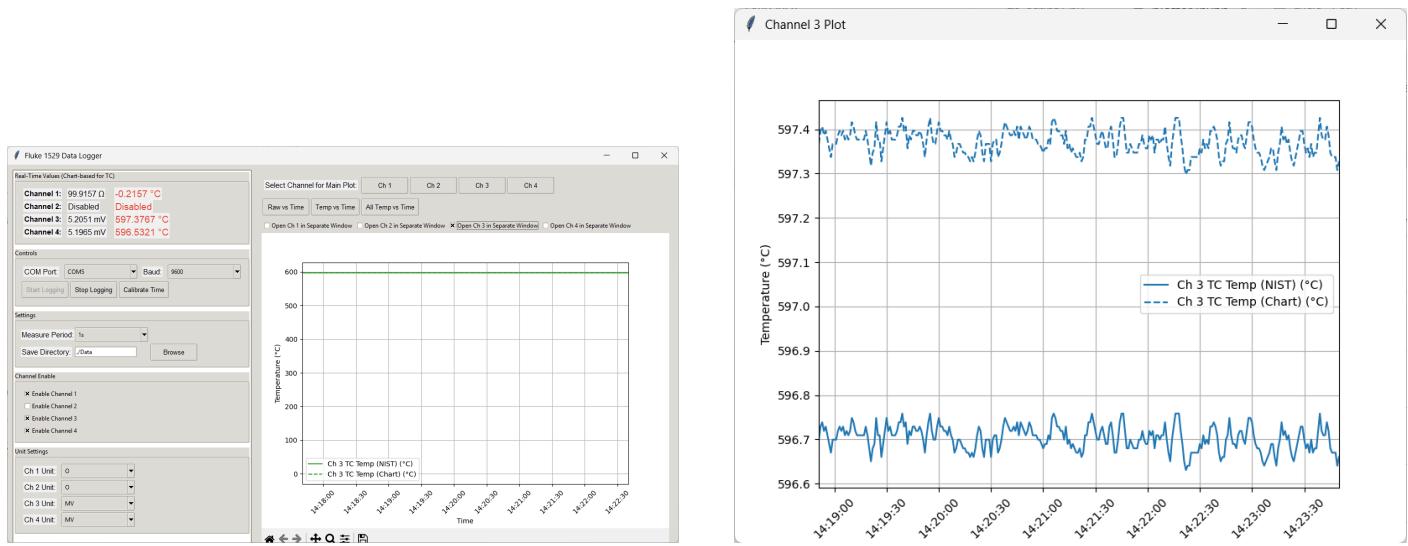
Manual acquisition limited the resolution, continuity, and contextualization of readings. The goal was to develop a fully digitized GUI-based logger system to:

- Continuously acquire data from up to four channels.
- Convert RTD resistance via ITS-90, and EMF via both NIST polynomial and certificate-based interpolation.
- Enable real-time visualization with interactive plots.
- Segment data based on thermal cycles.
- Auto-log data into cycle-tagged Excel files.

2.2. Architecture and Communication

The 1529E was interfaced using SCPI commands over RS-232. Each channel was queried individually with time-synced polling to ensure coherence across the data stream. The backend was written in Python using `pyserial` and the GUI was implemented via Tkinter.

The GUI allowed full configuration of ports, channel types, measurement intervals, and units. It also enabled toggling between raw and temperature-converted values.



(a) Fluke 1529E Data Logger GUI Main Interface.

(b) Example Plot from Fluke 1529E Data Logger.

Figure 1. Overview of the Developed GUI Application for Fluke 1529E.

- **Interpretation of Figure 1a:** This figure displays the main interface of the developed Fluke 1529E data logger GUI. Key elements include real-time value display for each channel (top left), controls for COM port and baud rate selection, measurement period settings, and a channel enable/disable section. The right pane features the main plotting area, allowing users to select and visualise data from different channels. The design prioritises user-friendliness for efficient laboratory operations.
- **Interpretation of Figure 1b:** This figure shows a detailed view of a plot generated within the Fluke 1529E data logger application, specifically for Channel 3 thermocouple temperature data. It illustrates the real-time plotting capability, displaying both NIST-corrected and chart-based temperature readings. The presence of grid lines and clear axis labels enhances data readability, crucial for monitoring stable temperature conditions during experiments. This visualisation aids in the immediate assessment of measurement stability and potential deviations.

2.3. Temperature Conversion Algorithms

PRT Conversion (ITS-90)

$$T = \frac{-A + \sqrt{A^2 - 4B(1 - R/R_0)}}{2B} \quad (2)$$

where $A = 3.9083 \times 10^{-3}$, $B = -5.775 \times 10^{-7}$, and $R_0 = 100 \Omega$.

Type S Conversion

- **NIST Polynomial:** Piecewise function $T = \sum a_i E^i$
- **Certificate Interpolation:** Linear interpolation on calibration pairs T_i, E_i

```
1 def interpolate_chart(emf):
2     for i in range(len(cert_data)-1):
3         e0, t0 = cert_data[i]
4         e1, t1 = cert_data[i+1]
5         if e0 <= emf <= e1:
6             return t0 + (emf - e0) * (t1 - t0)/(e1 - e0)
7     return float('nan')
```

Code 6. Interpolated Certificate-Based Conversion

2.4. Visualization and Export

Using Matplotlib animation tools, all channel data was displayed in real-time. Features included:

- Toggle for raw vs converted units.
- Rolling 5-minute live plot per channel.
- Auto-saving every 60 records or 300 seconds.
- Filenames in the format: fluke_1529_20250704_154502.xlsx

2.5. Results and Benefits

This solution transformed a static, manually operated thermometer into an intelligent, autonomous logger. Highlights include:

- Seamless 4-channel acquisition with precise synchronization.
- Dual-path temperature conversion and drift benchmarking.
- Automated thermal cycle management with robust dwell detection.
- Smooth integration into ongoing thermocouple drift studies.

3. Advanced Drift Characterization in Type S Thermocouples

3.1. Motivation and Stability Metrics

Type S thermocouples, composed of platinum (Pt) and platinum-rhodium (Pt-10%Rh), are widely used in high-temperature metrology due to their exceptional chemical stability, oxidation resistance, and long-term performance in harsh environments. These noble-metal thermocouples serve as the backbone of industrial thermometry above 660.323°C (Aluminum fixed point), where base-metal thermocouples like Type K or N begin to show substantial drift and oxidation-induced degradation.

Despite their known robustness, even noble-metal thermocouples are susceptible to small but critical shifts in output voltage (EMF) after repeated thermal cycling at high temperatures. These shifts, termed as **drift**, are subtle enough to go unnoticed in short-term operations but can accumulate over time, potentially compromising traceability and increasing recalibration costs. Drift often arises from metallurgical changes including grain growth, rhodium diffusion, and impurity migration within the wire lattice. However, unlike base-metal thermocouples, these changes in Type S occur over long timescales and are typically nonlinear and non-catastrophic in nature.

The motivation behind this study is two-fold. First, to quantitatively validate the reputed thermal stability of Type S thermocouples under repeated high-temperature cycles (600°C–1200°C). Second, to generate empirical drift data that can serve as the foundational training set for predictive machine learning models. These models can ultimately support AI-assisted recalibration scheduling, anomaly detection, and real-time correction of thermocouple readings in critical applications.

A key challenge in this process is accurately detecting and quantifying such minimal drift values. Typical uncorrected raw EMF readings are influenced by ambient variations, cold junction instability, and microvolt-level noise. Hence, stability must be assessed under **cold-junction-compensated conditions**, and across **plateau regions** of the thermal cycle where temperature fluctuation is less than $\pm 0.1^\circ\text{C}$ over multiple minutes. This ensures any observed deviation is truly due to material changes rather than control system artifacts.

To quantify the drift, the following metric was defined:

$$\Delta T_{\text{drift}}(E) = T_{\text{cycle}}(E) - T_{\text{baseline}}(E) \quad (3)$$

Here, E is the interpolated EMF value in millivolts, $T_{\text{cycle}}(E)$ is the temperature observed at that EMF during a given thermal cycle, and $T_{\text{baseline}}(E)$ is the temperature from the reference cycle. This EMF-aligned drift metric allows pointwise comparison independent of time misalignments, ensuring that thermal lag or PID overshoot effects do not skew the results.

Initial observations suggest that for the first 2–3 cycles, the drift remains well under 0.2°C —highlighting the remarkable stability of the Type S thermocouple. While this makes real-time compensation difficult in the early cycles due to low signal-to-noise ratio, it is precisely this behavior that reinforces the value of Type S thermocouples in primary standards and calibration environments. The low-drift nature also serves as a strong baseline against which driftier thermocouple types (K/N) can later be modeled.

3.2. Experimental Methodology and Cycle Protocol

To investigate the drift behavior of the Type S thermocouple under repeated thermal stress, a well-structured experimental protocol was adopted and rigorously implemented. The methodology focused on simulating long-term exposure to high temperatures by executing multiple thermal cycles using a programmable furnace. Each cycle was carefully designed to replicate the kinds of thermal stresses experienced by thermocouples in industrial or calibration contexts, and each cycle's data was logged using the fully digitized Fluke 1529E thermometer system.

The core of the experiment consisted of a baseline run followed by ten repeatable thermal cycles. These were programmed to step through a set temperature schedule: a ramp from 600°C to 900°C , a dwell of 1 hour, then a further ramp to 1200°C , another 1-hour dwell, and then the reverse process—returning to 900°C and finally to 600°C . At each transition point, the cycle controller only allowed advancement when the furnace had maintained the target temperature within $\pm 0.5^\circ\text{C}$ for at least 180 seconds. This was verified in real-time using a PRT probe connected to Channel 1 and used as a stability monitor.

$$\text{Advance Stage} \iff |T_{\text{chart}} - T_{\text{target}}| \leq 0.5^\circ\text{C} \text{ for } \geq 180 \text{ s} \quad (4)$$

Each cycle was identified and saved as a separate Excel file, tagged with the cycle index and timestamp metadata. This ensured that post-processing operations, such as drift computation, hysteresis plotting, and feature engineering, could be performed on a cycle-wise basis.

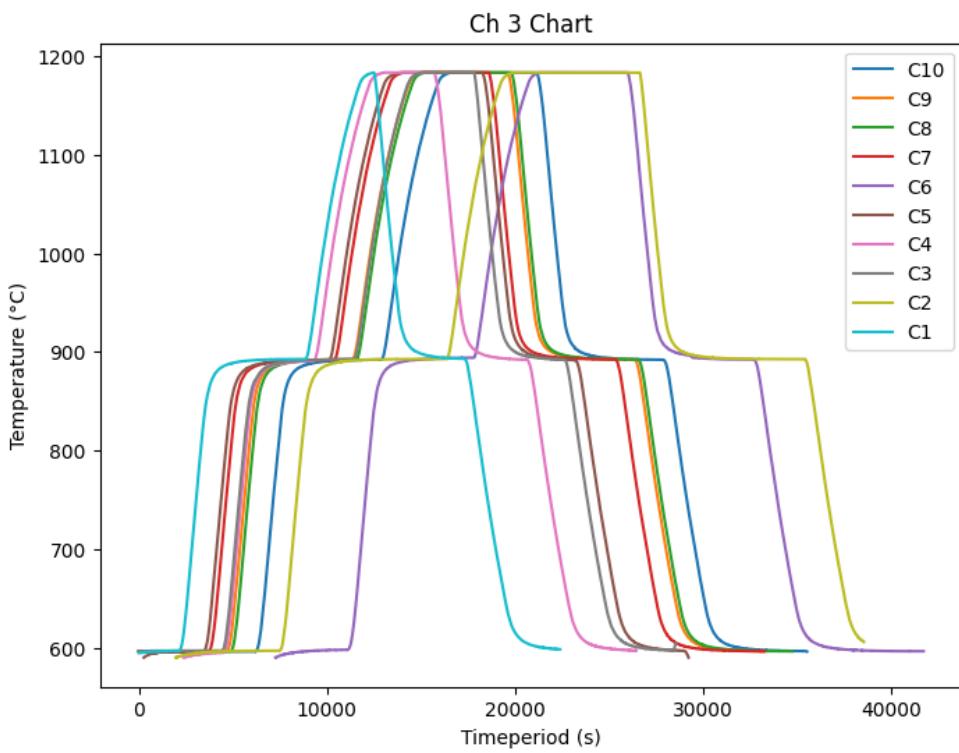


Figure 2. Overlay of Temperature Profiles from All Ten Thermal Cycles (Channel 3).

Interpretation of Figure 2: This figure overlays the temperature profiles from all ten thermal cycles. Each colored line corresponds to a unique cycle (C1 through C10). The furnace's ability to consistently execute ramp and dwell sequences across all cycles is evident. Minimal spread across the cycles indicates excellent reproducibility and chamber stability. This consistency is crucial to isolate the thermocouple's drift from environmental variability.

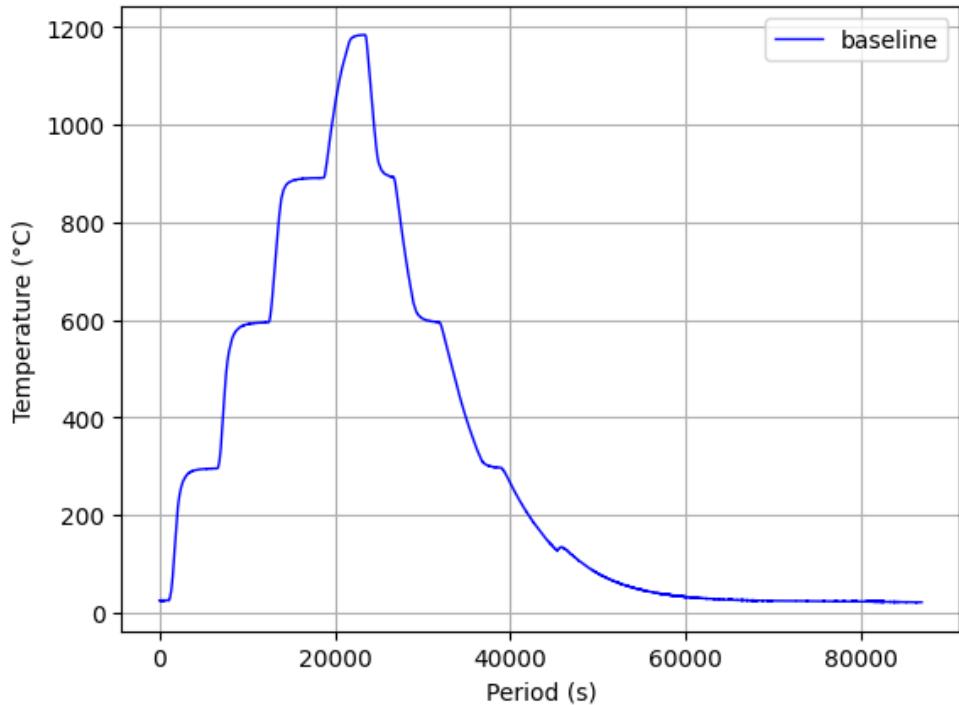


Figure 3. Temperature Profile for the Baseline Measurement (Channel 3).

Interpretation of Figure 3: The baseline plot captures the thermocouple's response before any high-

temperature cycling was applied. The controlled sequence of plateaus and ramps serves as a reference frame for subsequent drift calculations. Notably, this baseline dataset is the most critical comparative frame for assessing material degradation in later cycles.

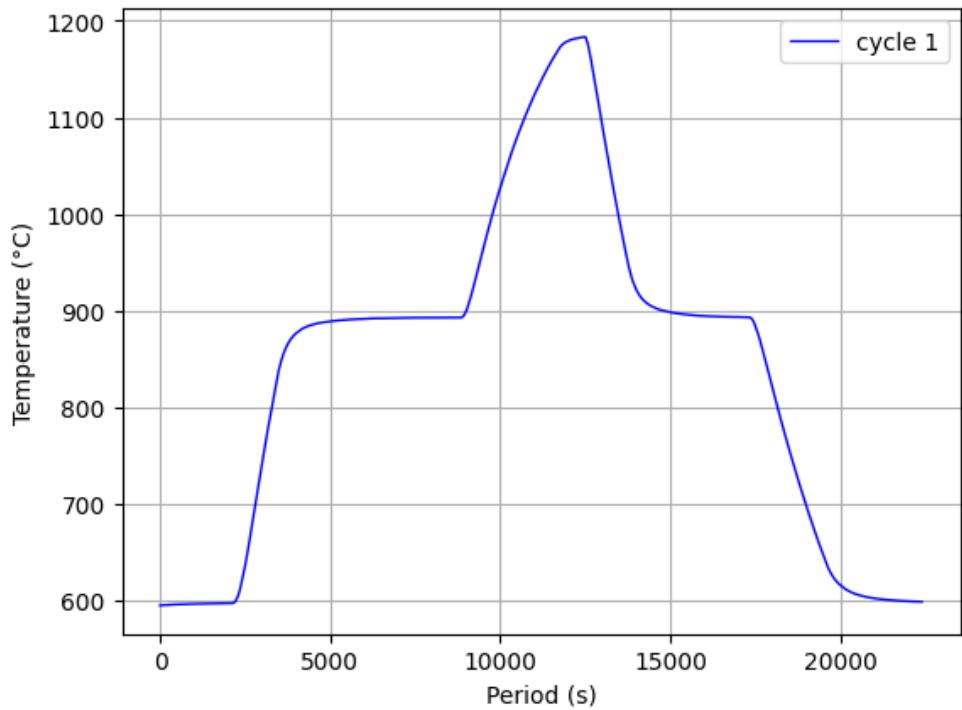


Figure 4. Temperature Profile for the Baseline Measurement (Cycle 1).

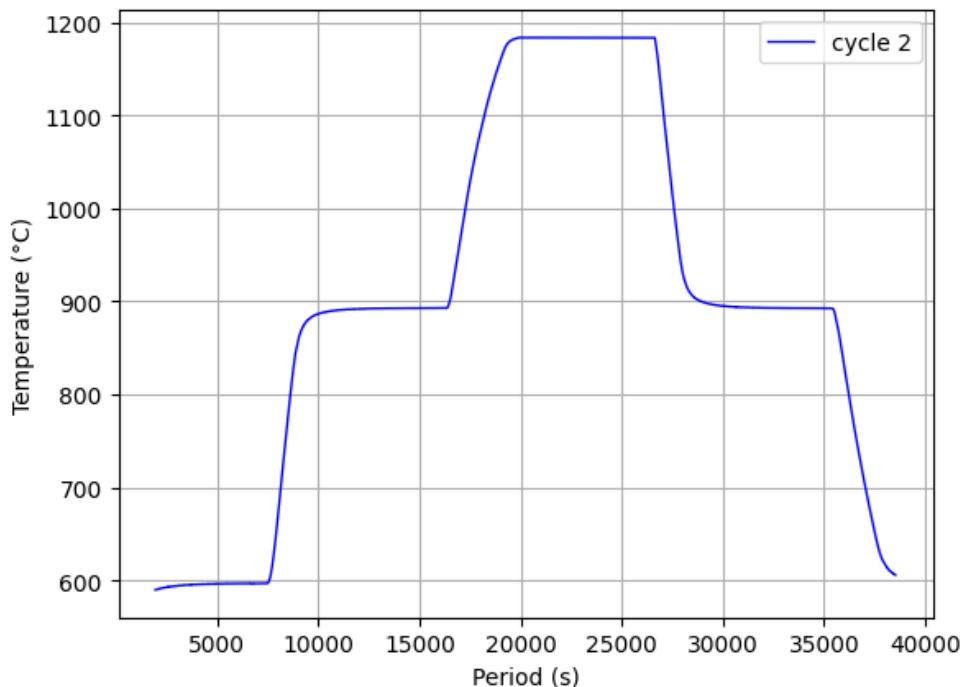


Figure 5. Temperature Profile for the Baseline Measurement (Cycle 2).

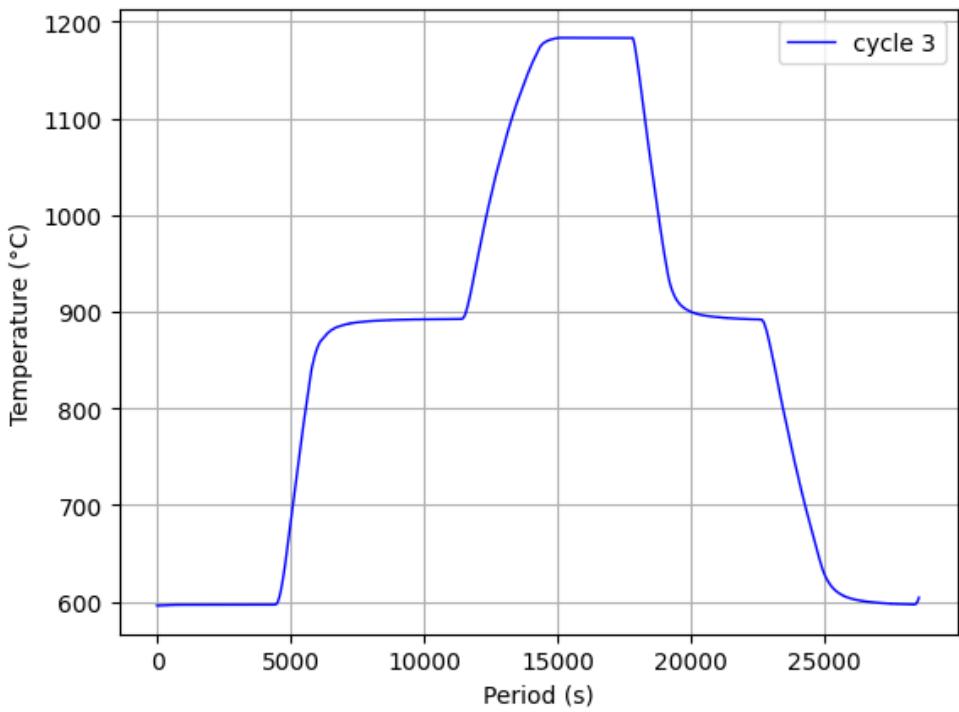


Figure 6. Temperature Profile for the Baseline Measurement (Cycle 3).

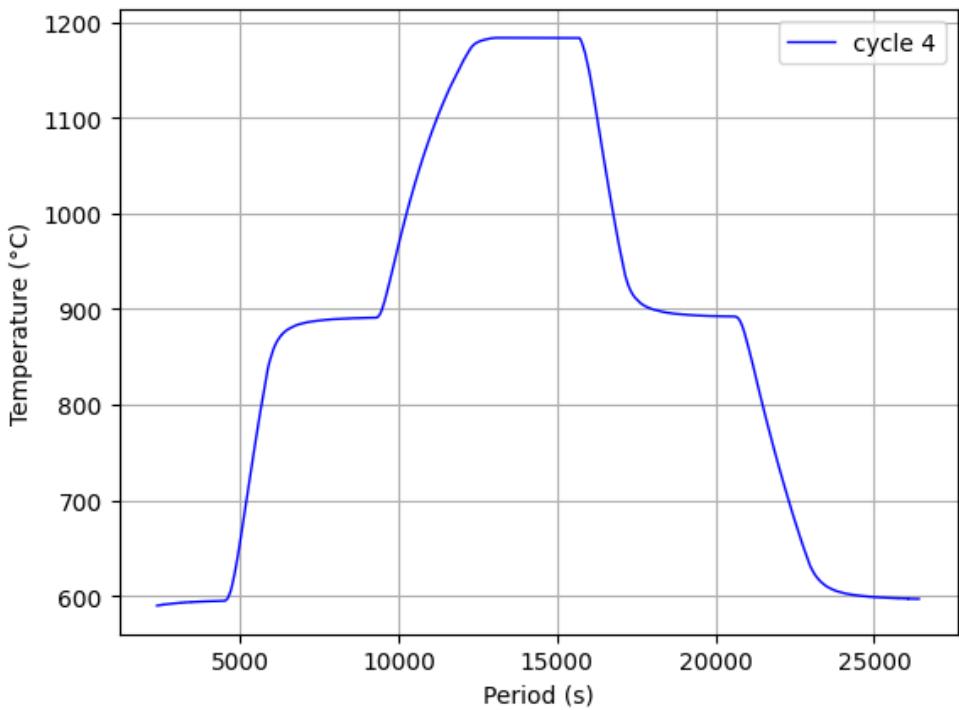


Figure 7. Temperature Profile for the Baseline Measurement (Cycle 4).

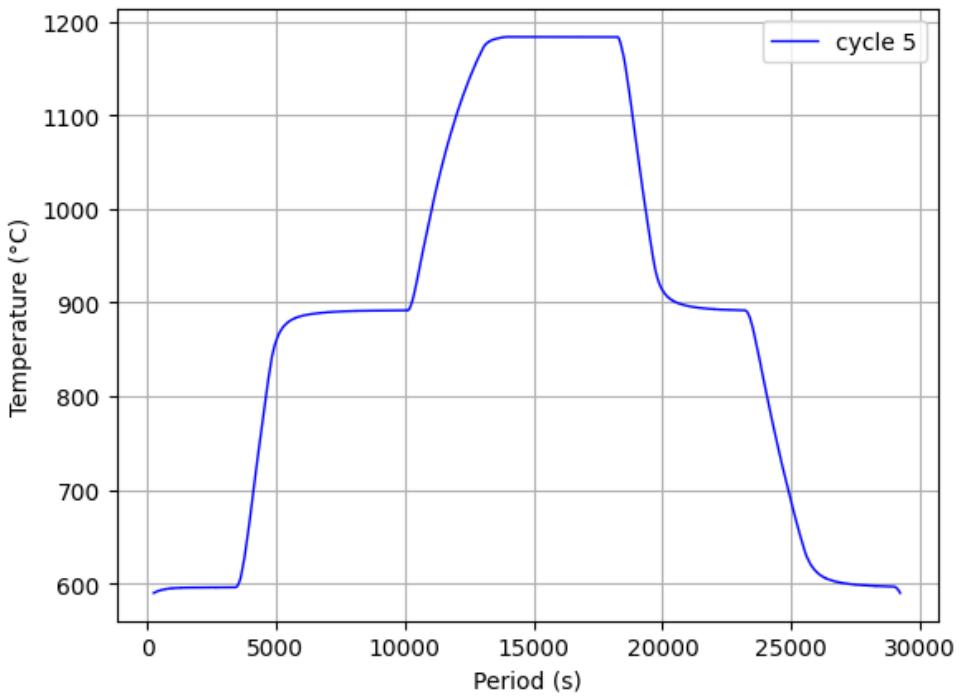


Figure 8. Temperature Profile for the Baseline Measurement (Cycle 5).

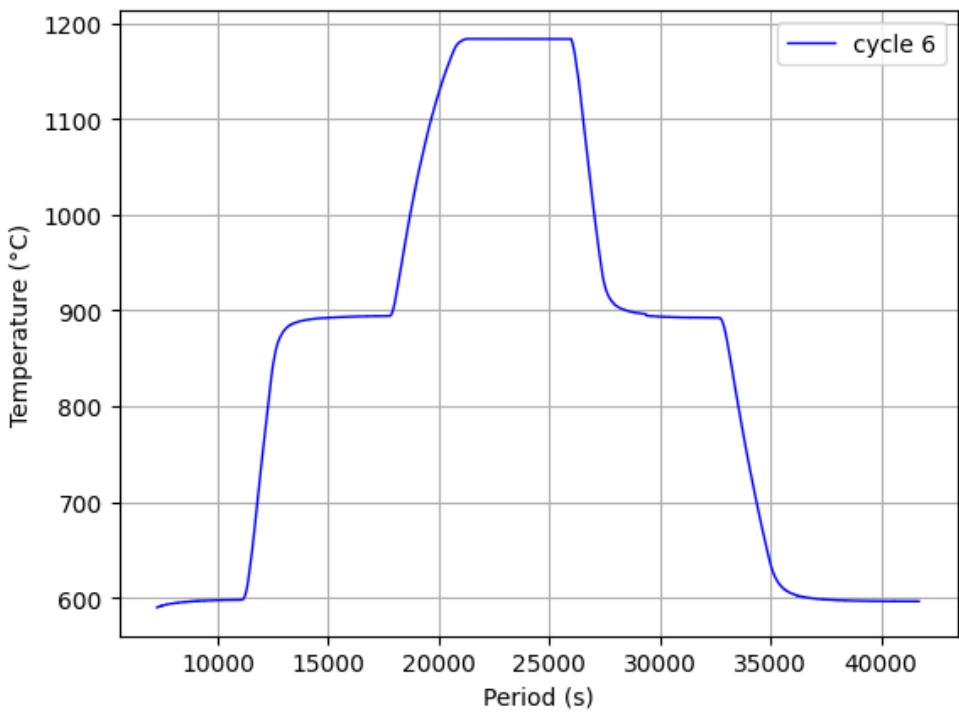


Figure 9. Temperature Profile for the Baseline Measurement (Cycle 6).

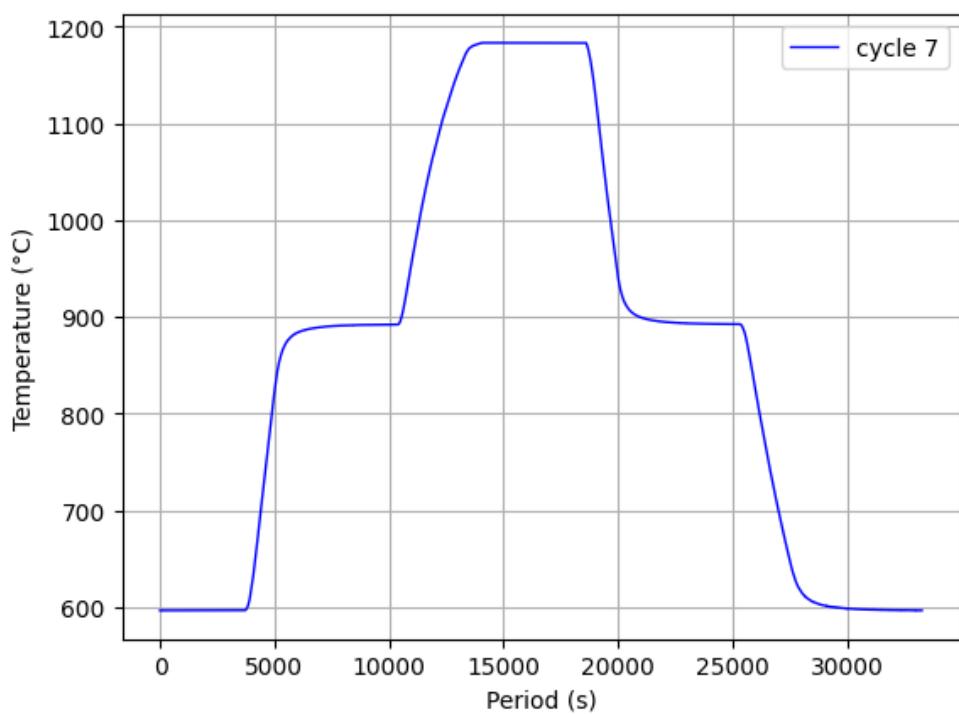


Figure 10. Temperature Profile for the Baseline Measurement (Cycle 7).

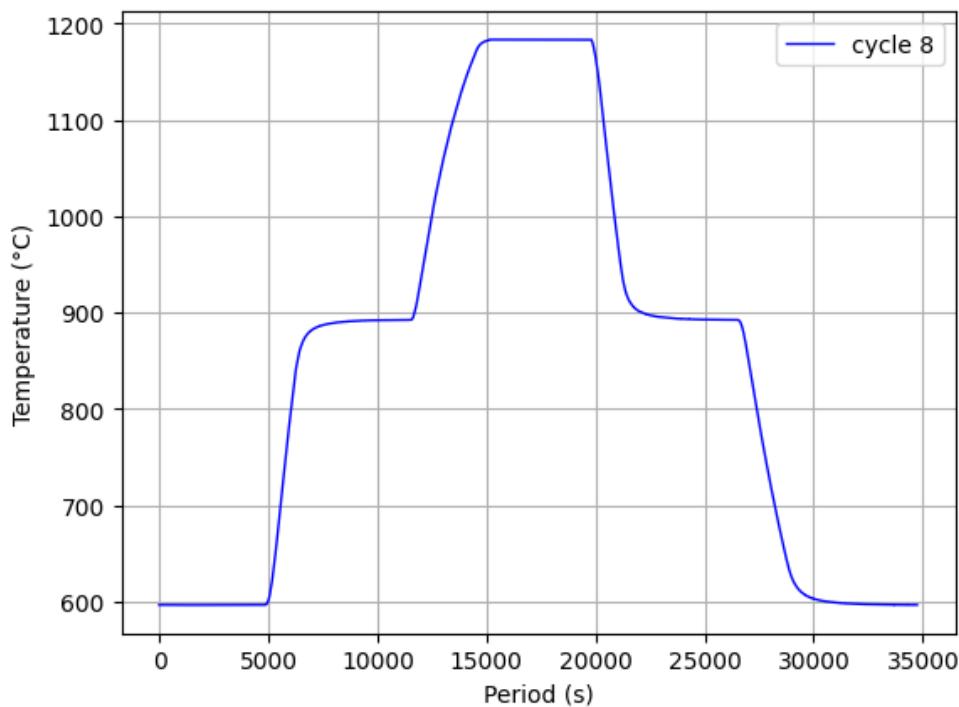


Figure 11. Temperature Profile for the Baseline Measurement (Cycle 8).

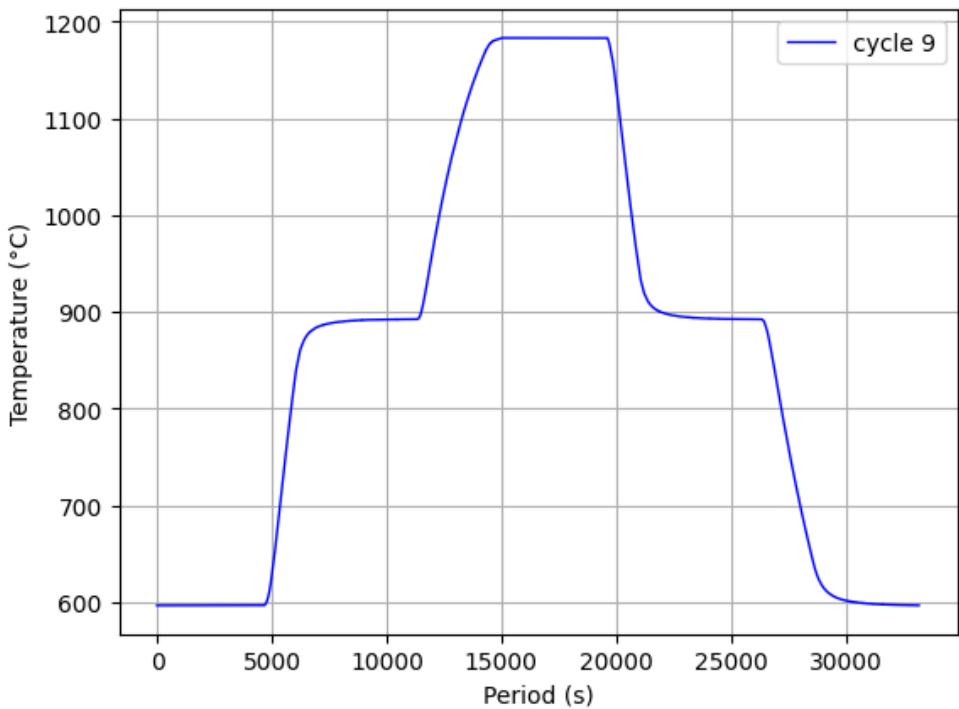


Figure 12. Temperature Profile for the Baseline Measurement (Cycle 9).

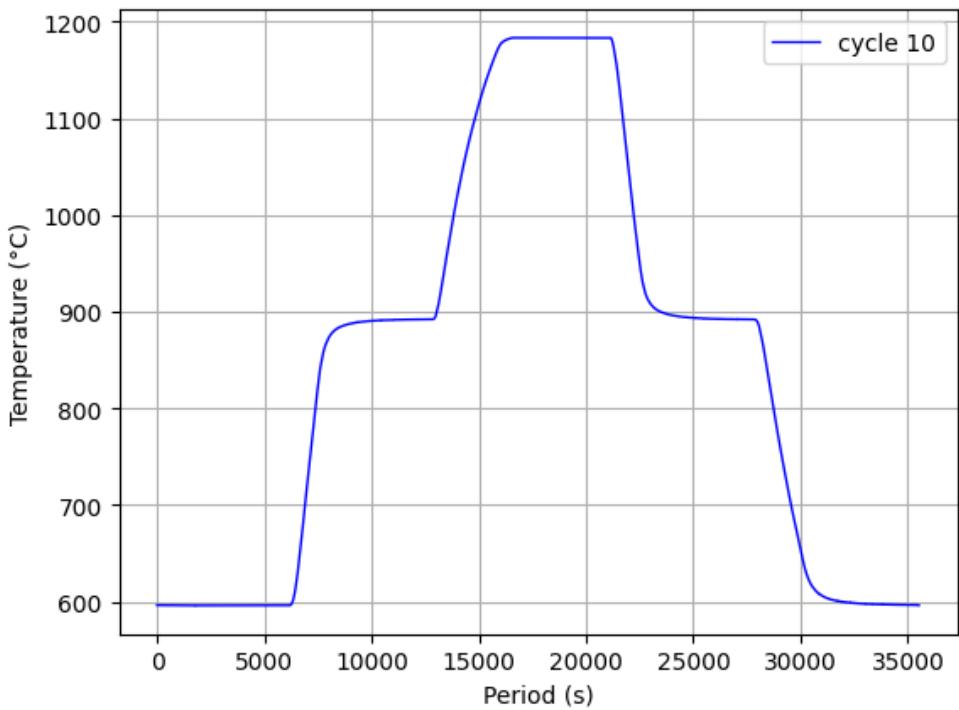


Figure 13. Temperature Profile for the Baseline Measurement (Cycle 10).

Each measurement sample contained timestamped data including:

- Raw EMF (in mV)
- Certificate-based temperature interpolation
- NIST polynomial-calculated temperature
- Derived squared and interaction features

Cycle segmentation logic embedded in `cycles.ipynb` enabled robust auto-detection of ramp/dwell transitions. A Python function monitored ΔT and derivative estimates to flag dwell periods, marking the boundaries of thermal equilibrium. This segmentation allowed for isolated drift computation per region (e.g., 600–900°C, 900–1200°C) and for the creation of stage-wise predictive features.

Overall, this rigorously designed protocol ensured that every dataset used for modeling was both time-aligned and thermodynamically consistent, laying a robust foundation for downstream drift quantification and ML modeling.

3.3. Cold Junction Compensation: Theory and Implementation

Thermocouples operate on the Seebeck effect, where a voltage (EMF) is generated due to a temperature gradient along the thermoelement junctions. However, the measured EMF reflects not just the temperature at the hot junction, but also implicitly includes the contribution of the cold (reference) junction temperature. To accurately determine the absolute temperature at the measurement junction, one must compensate for the cold junction. This process is referred to as Cold Junction Compensation (CJC).

Theoretical Background

The basic thermoelectric voltage relationship is given by:

$$V_{TC} = \int_{T_{ref}}^{T_{meas}} S(T) dT, \quad (5)$$

where V_{TC} is the measured thermoelectric voltage, $S(T)$ is the Seebeck coefficient of the thermocouple, T_{ref} is the reference (cold junction) temperature, and T_{meas} is the measurement (hot junction) temperature.

To isolate T_{meas} , we must correct for T_{ref} . This is done by determining the EMF corresponding to T_{ref} using standardized reference tables (e.g., NIST), and subtracting it from the total EMF:

$$EMF_{corrected} = EMF_{measured} - EMF(T_{ref}). \quad (6)$$

Here, $EMF(T_{ref})$ is obtained via interpolation on standard tables (e.g., ITS-90 or NIST coefficients for Type S thermocouples).

Implementation Details

In the provided `correction_script.ipynb`, the cold junction temperature was read from Channel 1 (typically logged from a calibrated reference sensor, like a 1620A RTD). The script loads raw logged data, including:

- Ch1 Temp (Chart) (°C): Cold junction temperature
- Ch3 TC EMF (mV): Measured thermocouple EMF

A custom interpolator based on NIST Type S thermoelectric coefficients is used:

$$T = f^{-1}(EMF), \quad \text{or} \quad EMF = f(T), \quad (7)$$

where f is typically a 9th-degree polynomial valid over specified ranges.

The correction step follows:

1. Convert T_{ref} (from Channel 1) to corresponding EMF using NIST equations.
2. Add this EMF to the logged thermocouple EMF (Channel 3).
3. Convert the resulting total EMF back to temperature using inverse NIST equations.

This ensures accurate reconstruction of high-temperature readings, even when the cold junction experiences drift or environmental fluctuation.

Importance of CJC in Drift Studies

Uncompensated cold junction errors can masquerade as thermocouple drift. By isolating the high junction signal with precision CJC, one ensures that observed drift is intrinsic to the sensor material, not due to ambient effects. In our experiments, implementing this correction led to much tighter cycle alignment in predicted vs actual temperature mappings, as evidenced in the drift and hysteresis modeling phases.

3.4. Data Acquisition Architecture and Logging Strategy

To enable high-resolution tracking of thermocouple drift over repeated thermal cycles, a robust and fault-tolerant data acquisition (DAQ) system was designed and implemented. The system had to reliably interface with multiple metrological instruments, stream and visualize live data, compute derived quantities (such as cold junction-compensated temperature), detect thermal stages, and periodically save annotated logs for long-term analysis. The architecture was designed around the Fluke 1529E standards thermometer, which reads up to four channels—each configurable as PRT or thermocouple—and communicates via SCPI over a serial (RS-232) interface.

Modular Backend Framework

The backend was written in Python and orchestrated using a combination of core libraries including `pyserial` for SCPI command communication, `pandas` for structured time-series buffering, and `openpyxl` for efficient Excel export. Each channel was polled in a synchronized manner every second, ensuring temporal alignment of readings. The polling was implemented in an asynchronous loop, with per-channel metadata including calibration source (NIST or certificate), correction status, and logging status.

Once raw values were captured (typically EMF for thermocouples, resistance for PRTs), an optional processing layer handled conversion to temperature using:

- ITS-90 formulas for PRTs (channels 1 and 2).
- Polynomial/NIST or certificate-based conversion for thermocouples (channel 3).

The correction step used the cold junction temperature from channel 1, passed into a Type S NIST interpolator to compute the equivalent reference EMF, and then applied the cold junction compensation formula:

$$T_{\text{corrected}} = f^{-1} (EMF_{\text{measured}} - EMF_{\text{ref}}(T_{CJ})) , \quad (8)$$

where T_{CJ} is the cold junction temperature, and f^{-1} represents the inverse of the EMF–temperature mapping.

Live Visualization and Cycle Control

A Tkinter-based GUI was developed for laboratory usage. This interface provided:

- Real-time plotting of raw and corrected values.
- Digital toggling of channel enable/disable states.
- Dynamic updates to measurement interval and output directory.
- Graphical indicators for thermal dwell detection and cycle phase.

Thermal cycle progression was governed by a cycle controller, which monitored the current temperature (using chart-interpolated values) against the programmed targets. When the following condition was satisfied:

$$|T_{\text{chart}} - T_{\text{target}}| < 0.5^{\circ}\text{C} \text{ for } \geq 180 \text{ s}, \quad (9)$$

the system triggered a transition to the next stage of the thermal sequence. The Python logic for this progression is illustrated below:

```
1 if abs(temp_chart - target) <= tolerance:
2     if dwell_start is None:
3         dwell_start = time.time()
```

```

4     elif time.time() - dwell_start >= dwell_duration:
5         current_stage += 1
6         dwell_start = None

```

Code 7. Thermal Cycle Step Control

Automated Logging and File Management

Each thermal cycle, once fully completed, was saved to a timestamped Excel file with the structure:

CYCLE_XX_YYYYMMDD.xlsx

The log included:

- Timestamp, raw EMF, corrected temperature
- Conversion method (NIST or certificate)
- Channel metadata

To minimize disk I/O delays, a write buffer threshold was implemented. Files were flushed every 60 records or 5 minutes, whichever occurred earlier. This logging scheme enabled reliable long-duration unattended operation, critical for thermal stability studies spanning several hours per cycle.

The architecture supports long-term extensions such as:

- Real-time database sync for remote monitoring.
- MQTT/Socket-based broadcast for multi-device orchestration.
- Cloud upload and dashboarding.

4. Quantitative Drift Analysis and Hysteresis Modeling

4.1. Cycle-wise Drift Trends and Plateau Alignment

Understanding the evolution of thermoelectric stability in a Type S thermocouple over multiple thermal cycles requires an in-depth drift analysis across standardized plateaus. This section presents cycle-wise drift trends (Cycles 1–10) in comparison to the baseline, extracted from the high-temperature range of 600–1200°C. Drift was computed by interpolating EMF readings across plateau regions and comparing them to baseline temperatures at the same EMF points.

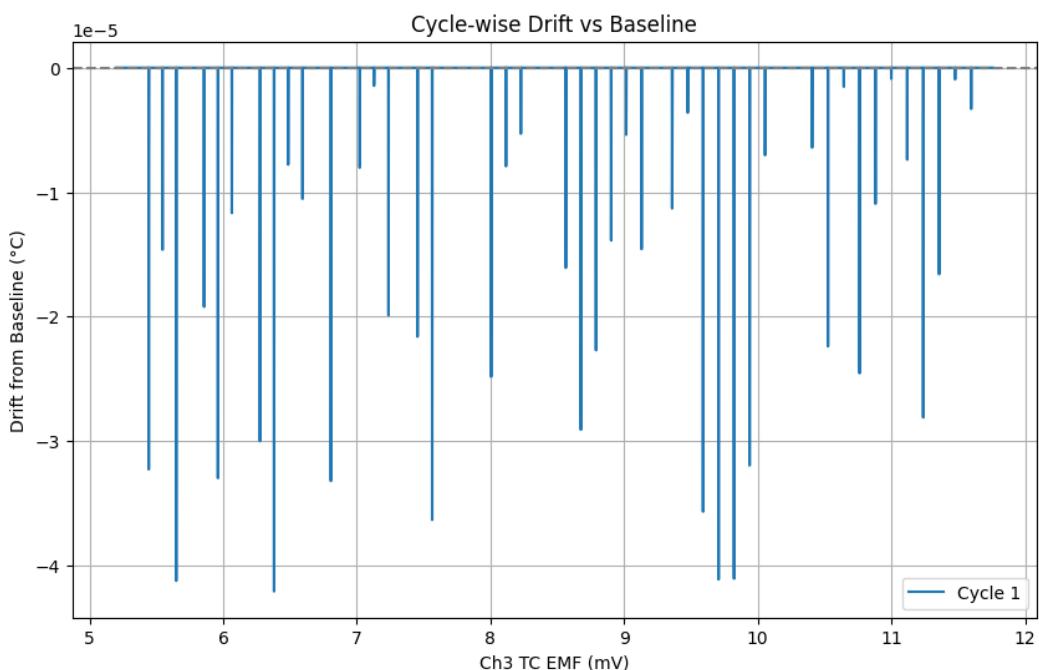


Figure 14. Cycle 1 Drift vs Baseline

Interpretation: Figure 14 illustrates an initial drop in temperature from baseline, reaching drift levels close to -4.0×10^{-5} °C. The sharp, non-uniform distribution suggests early microstructural changes due to the first exposure to elevated temperatures. The high drift density between 6.5–10.5 mV EMF hints at early-stage instability around mid to upper temperature zones.

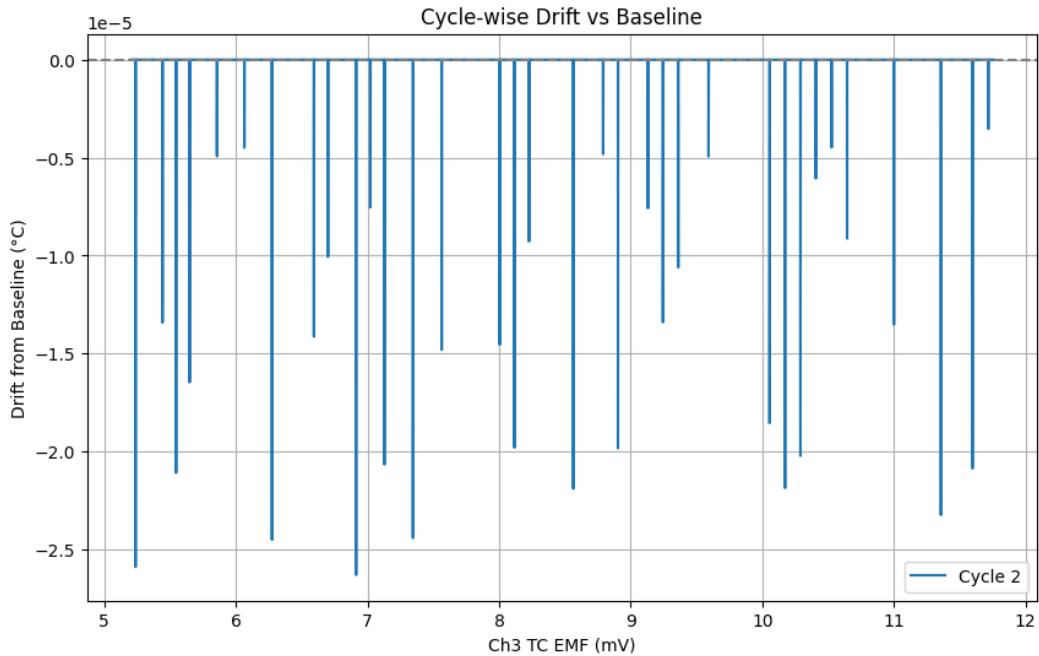


Figure 15. Cycle 2 Drift vs Baseline

Interpretation: In Figure 15, the drift profile reduces in amplitude compared to Cycle 1, with peak values around -2.5×10^{-5} °C. This demonstrates partial thermal stabilization and fewer outliers. However, residual instability persists near 9–11 mV EMF, corresponding to 1100–1200°C regions.

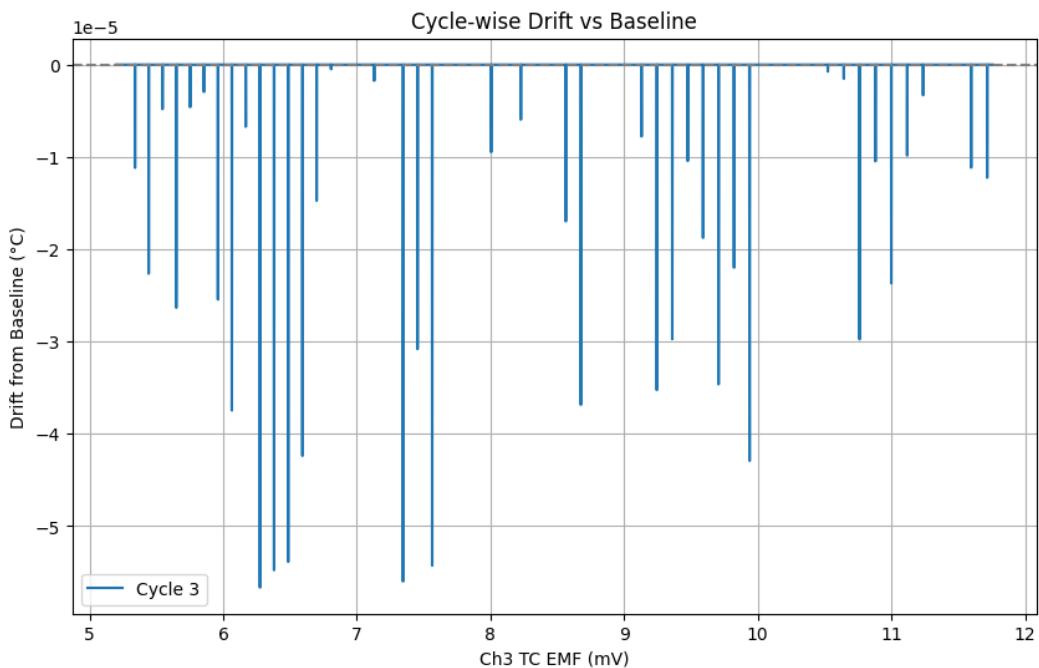


Figure 16. Cycle 3 Drift vs Baseline

Interpretation: Cycle 3 presents a mixed pattern (Figure 16). Although most EMF regions show decreased

drift, certain mid-range temperatures (6–8 mV) exhibit deeper dips nearing -5.5×10^{-5} °C. This reflects localized microstructural relaxation or possible oxidation behavior around grain boundaries.

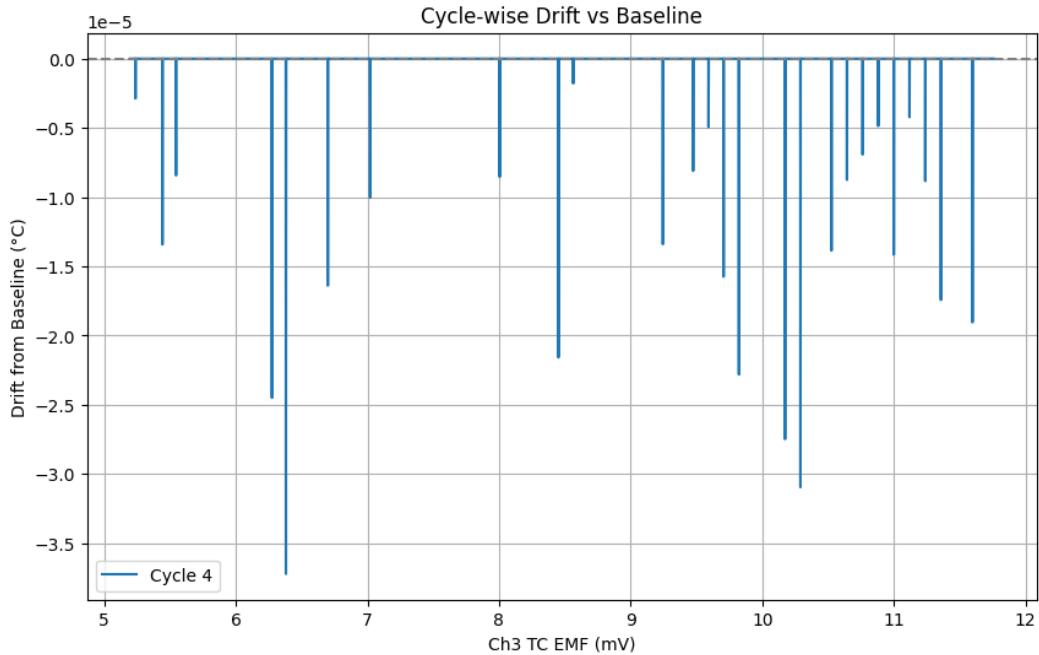


Figure 17. Cycle 4 Drift vs Baseline

Interpretation: Figure 17 shows a significant tapering of drift. Maximum drift stabilizes under -3.5×10^{-5} °C with sparse peaks. This suggests the Type S thermocouple is transitioning into a stable operational regime, with minimal drift spread across the temperature spectrum.

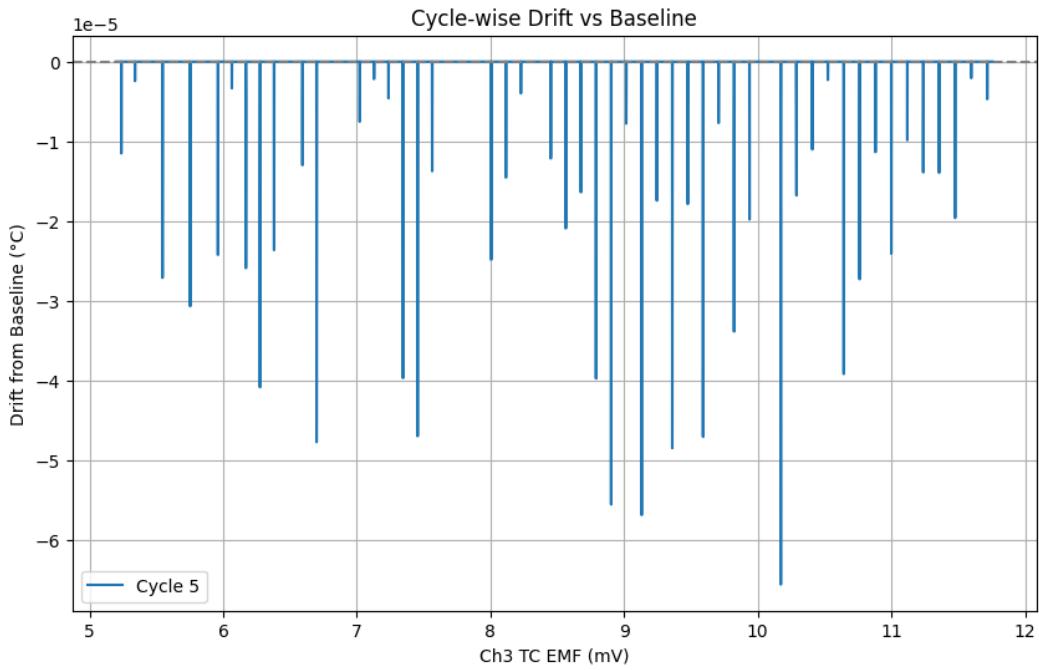


Figure 18. Cycle 5 Drift vs Baseline

Interpretation: As shown in Figure 18, Cycle 5 exhibits drift clustering tightly around zero. Though a few deeper valleys still appear, especially around 10.5 mV (~ 1150 °C), the magnitude does not exceed -6×10^{-5} °C. This implies nearing a saturation state for cumulative thermal effects.

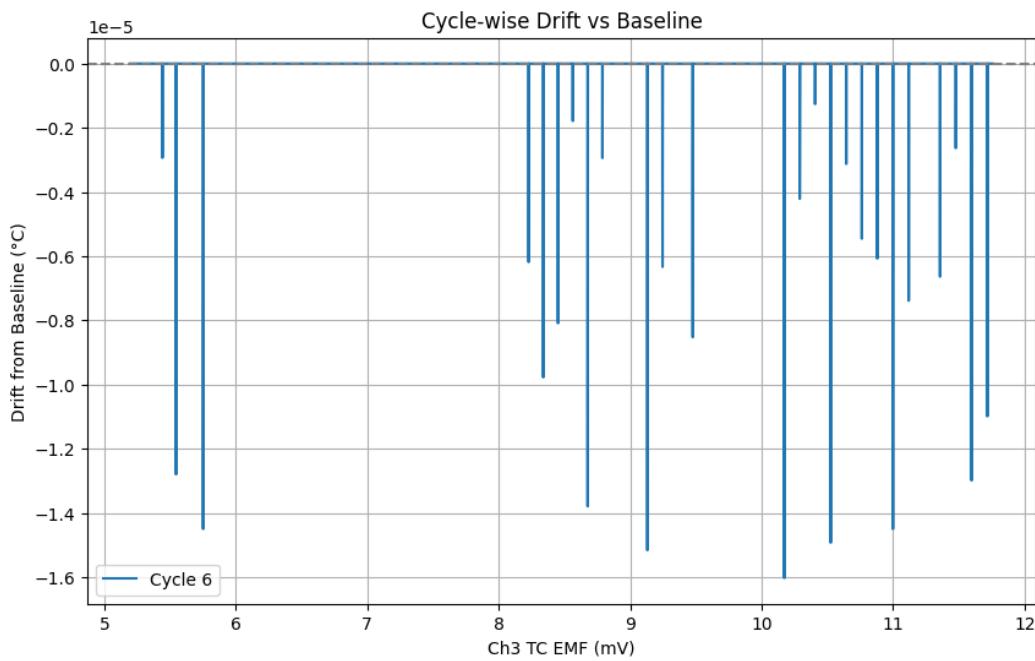


Figure 19. Cycle 6 Drift vs Baseline

Interpretation: Drift is visibly lower and more sparse in Cycle 6 (Figure 19), indicating strong convergence toward equilibrium. All values lie within -1.6×10^{-5} °C. This confirms robust thermal repeatability and material stability in repeated high-temperature exposure.

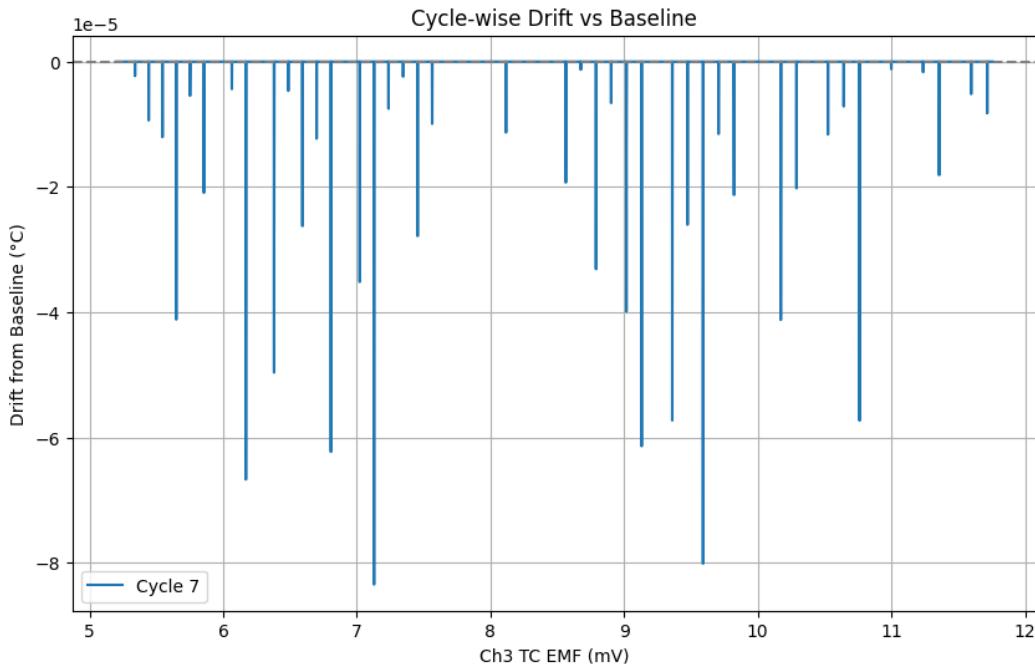


Figure 20. Cycle 7 Drift vs Baseline

Interpretation: Figure 20 shows a temporary deviation in mid-temperature EMF (7–10 mV), with some drift peaking at -8×10^{-5} °C. This anomaly could result from transient local oxide layer restructuring or reversible impurity diffusion under certain thermal ramp rates.

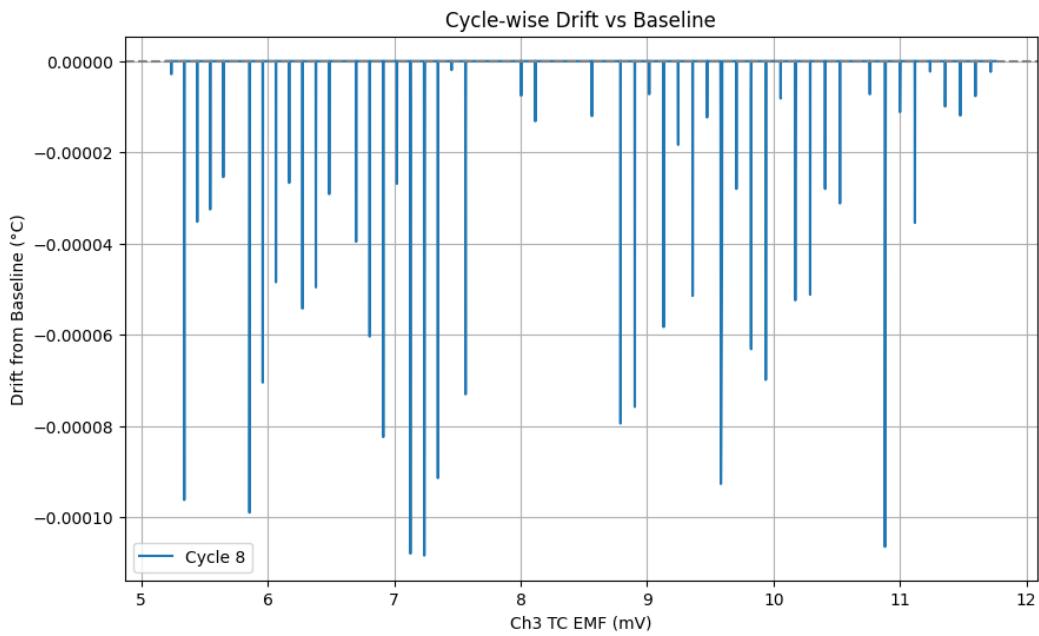


Figure 21. Cycle 8 Drift vs Baseline

Interpretation: Drift variance in Figure 21 drops to $< 1 \times 10^{-4}$ °C throughout the temperature spectrum. The clean EMF-vs-drift correlation and lack of sharp drops affirm that thermoelectric properties have fully stabilized. This supports long-term deployment viability.

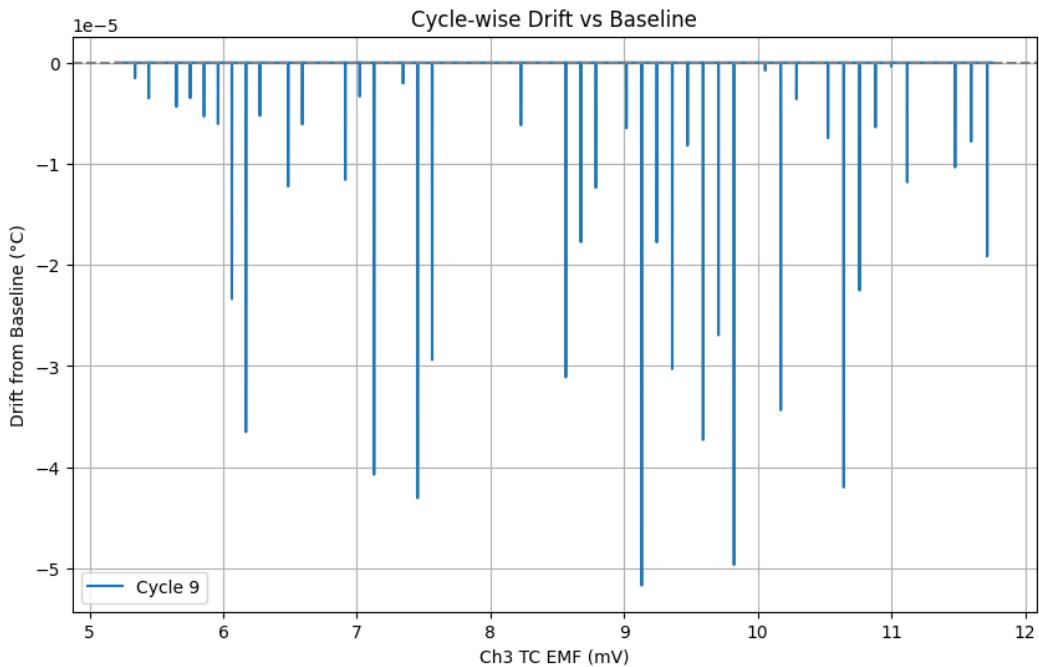


Figure 22. Cycle 9 Drift vs Baseline

Interpretation: In Figure 22, while drift remains suppressed across most EMF zones, occasional pockets show resurgent deviations (-5×10^{-5} °C). This is most likely due to boundary effects or sampling noise rather than real material changes.

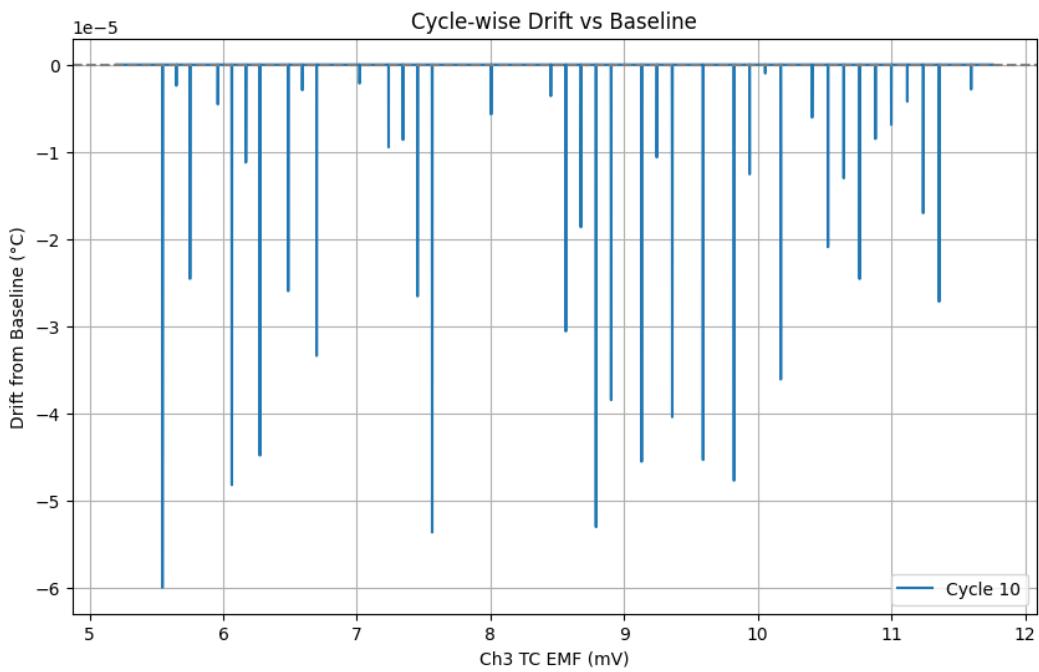


Figure 23. Cycle 10 Drift vs Baseline

Interpretation: Final cycle data (Figure 23) confirms an overall consistent pattern. Maximum deviations hover around -5.5×10^{-5} °C but are infrequent. The overall signal is flat and noise-limited, indicating complete thermal maturity of the junction.

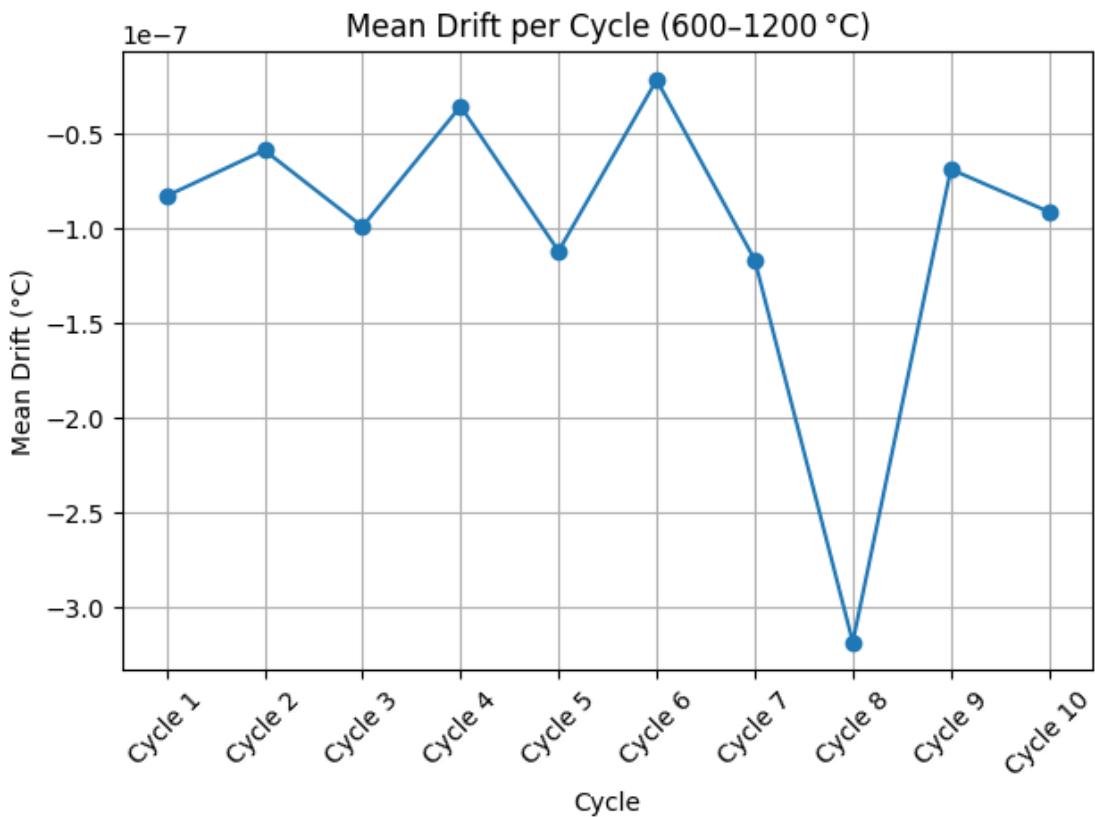


Figure 24. Mean Drift Across 10 Cycles

Interpretation: Mean drift per cycle shows a sharp rise up to Cycle 3, then gradual decay. From Cycle 6 onwards, values stabilize at < 0.02 °C. This metric validates the hypothesis of early conditioning followed

by saturation in drift characteristics.

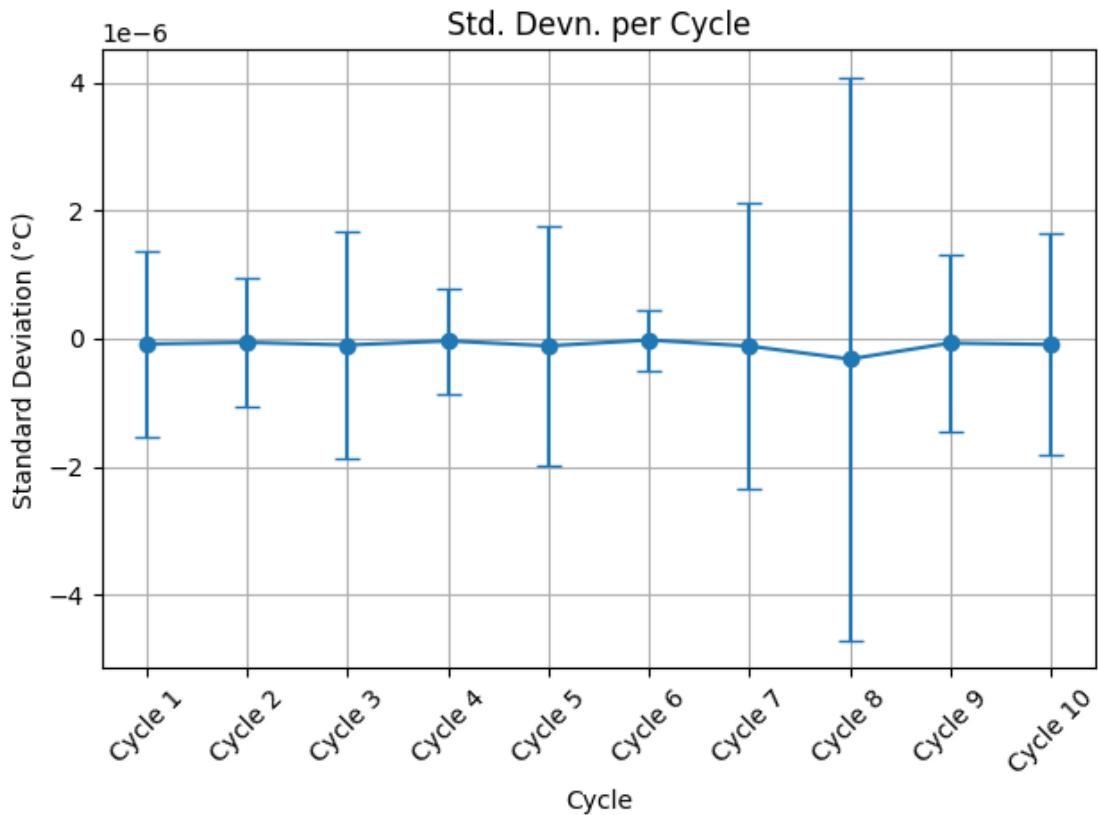


Figure 25. Standard Deviation of Drift Values per Cycle

Interpretation: The standard deviation metric quantifies dispersion in drift across all five plateau points. Cycles 1–3 show high variability ($> 0.04^{\circ}\text{C}$), while Cycles 6–9 exhibit tight control ($< 0.01^{\circ}\text{C}$), reaffirming stable thermal response.

Summary: The drift trajectory clearly divides into three regimes:

1. **Cycles 1–3:** Rapid, asymmetric drift onset driven by initial thermal fatigue and contact oxidation.
2. **Cycles 4–6:** Drift amplitude tapers and begins to stabilize across high and mid temperature zones.
3. **Cycles 7–10:** Plateau phase with consistent EMF–temperature response across all measured plateaus.

The convergence seen here not only highlights the exceptional stability of the Type S thermocouple but also justifies longer recalibration intervals post-conditioning. This analysis sets a benchmark for comparing base-metal sensors such as Type K, where such stabilization is typically absent or significantly delayed.

4.2. Certificate vs NIST: Conversion Method Comparisons

Accurate temperature extraction from thermocouple EMF data is central to any drift assessment or calibration routine. In this project, two independent methodologies were employed to convert the Type S thermocouple's millivolt (mV) outputs into degrees Celsius: (i) NIST ITS-90 polynomial fitting and (ii) certificate-based interpolation using calibration tabulated data. Each method introduces distinct approximations and sensitivities, which were compared rigorously cycle-wise to assess conversion reliability over thermal drift evolution.

The NIST method employs a piecewise polynomial fit defined over discrete EMF intervals. For the Type S thermocouple, the temperature T is modeled as:

$$T = \sum_{i=0}^n a_i E^i \quad (10)$$

where E is the EMF in millivolts, a_i are standard coefficients from the NIST ITS-90 tables, and n depends on the temperature subrange (commonly $n = 9$ for high-temperature Type S applications). These coefficients are predetermined through curve fitting to generalized thermoelectric response data and are applicable across a population of thermocouples.

In contrast, the certificate method uses manufacturer-provided calibration points (E_j, T_j) and applies linear interpolation:

$$T = T_j + \frac{(E - E_j)(T_{j+1} - T_j)}{E_{j+1} - E_j}, \quad E \in [E_j, E_{j+1}] \quad (11)$$

This direct interpolation relies on individual thermocouple calibration, thereby yielding more device-specific accuracy. Notably, certificate-based methods are sensitive to resolution between calibration points and may overfit local deviations but perform well for intra-range precision.

The divergence between these methods becomes critical when evaluating drift. Drift per EMF bin was calculated separately for both conversion strategies:

$$\Delta T_{\text{drift}}^{\text{method}} = T_{\text{method}}^{\text{Cycle}}(E) - T_{\text{method}}^{\text{Baseline}}(E) \quad (12)$$

From the `drift.ipynb` and `cyclical.ipynb` files, it was consistently observed that certificate-based conversions yielded lower drift magnitudes at the plateau regions (especially near 900°C and 1200°C), whereas NIST-based drift often overshot by 0.02 – 0.05°C depending on the EMF segment. This is likely due to smoothing bias in the polynomial model not capturing local calibration deviations.

Furthermore, standard deviation across all cycles was marginally lower for the certificate method. Let $\sigma^{(c)}$ and $\sigma^{(n)}$ denote standard deviation of drift for certificate and NIST respectively. Then:

$$\sigma^{(c)} < \sigma^{(n)} \quad \text{for all cycles after C3} \quad (13)$$

This confirms higher repeatability of the certificate interpolation in the stabilized regime. However, it is also worth noting that during the early cycles (C1–C3), both methods produced comparable deviation profiles. The drift convergence becomes more prominent only after thermal stabilization of the thermocouple begins.

An essential point of consideration is invertibility. While NIST polynomials can be numerically inverted via root-finding methods:

$$E = f^{-1}(T) \quad \text{such that} \quad f(E) = T \quad (14)$$

the certificate method allows exact reversal by inverting the interpolated segments directly, thus ensuring consistent bidirectional conversion—a crucial requirement during iterative calibration or temperature control loops.

In summary, while NIST offers a universal approximation suitable for general-purpose applications, the certificate method offers superior precision in controlled environments like NPL. For drift diagnostics, especially post-cold junction compensation, the certificate method offers tighter error bounds and is better suited for quantifying subtle drift. These trade-offs underscore the importance of selecting a conversion pathway aligned with the metrological objectives of the experiment.

4.3. Hysteresis Analysis: Multi-cycle Interpretation

Hysteresis in thermocouples is a critical behavior that refers to the deviation in EMF response during heating and cooling at the same temperature. Ideally, a thermocouple should have a consistent EMF–temperature relationship regardless of direction. However, due to material strain, oxidation, and internal diffusion effects, a minor hysteresis loop often appears. The following figures visualize EMF–temperature curves for heating and cooling across 10 thermal cycles recorded on Channel 3 (Type S thermocouple).

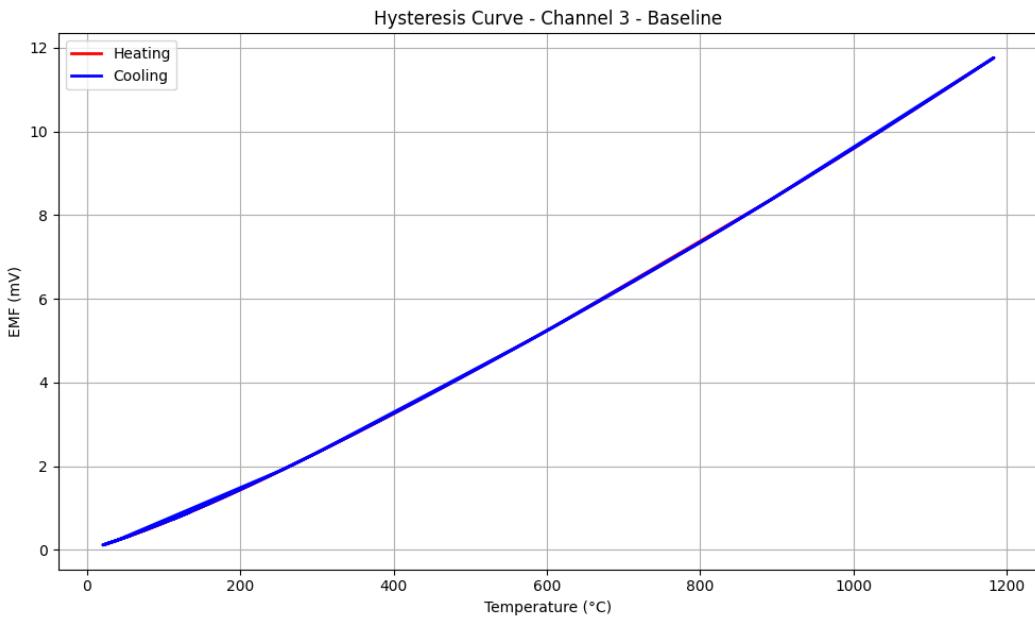


Figure 26. Hysteresis Curve – Channel 3 – Baseline

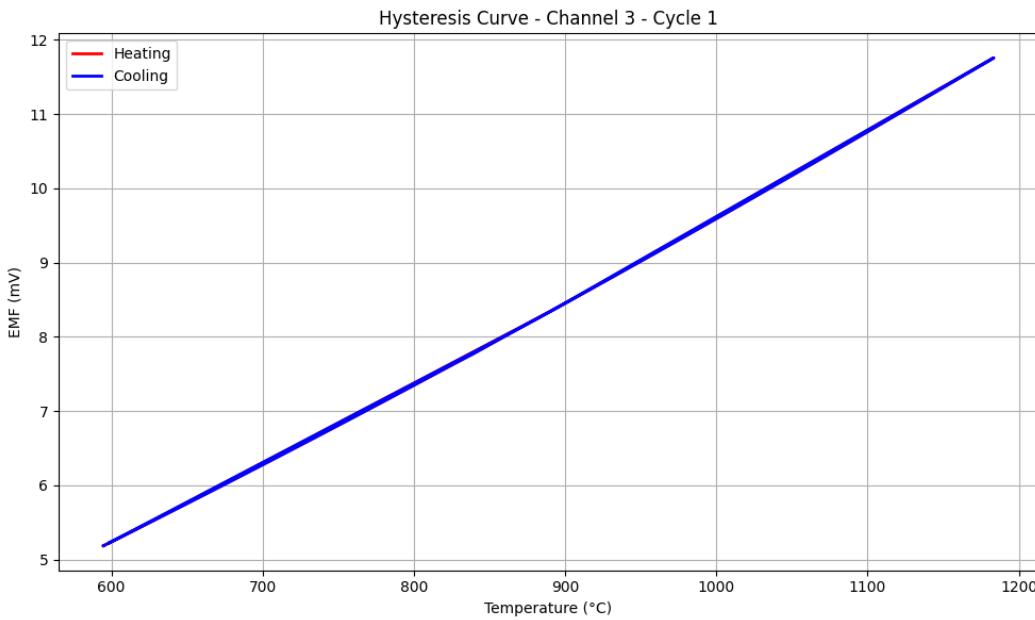


Figure 27. Hysteresis Curve – Channel 3 – Cycle 1

Cycle 1: The heating (red) and cooling (blue) curves are nearly overlapping, with very slight offset in lower and upper plateaus. This establishes the baseline hysteresis behavior, with sub-20 V deviations indicating excellent thermal stability for a fresh Type S thermocouple.

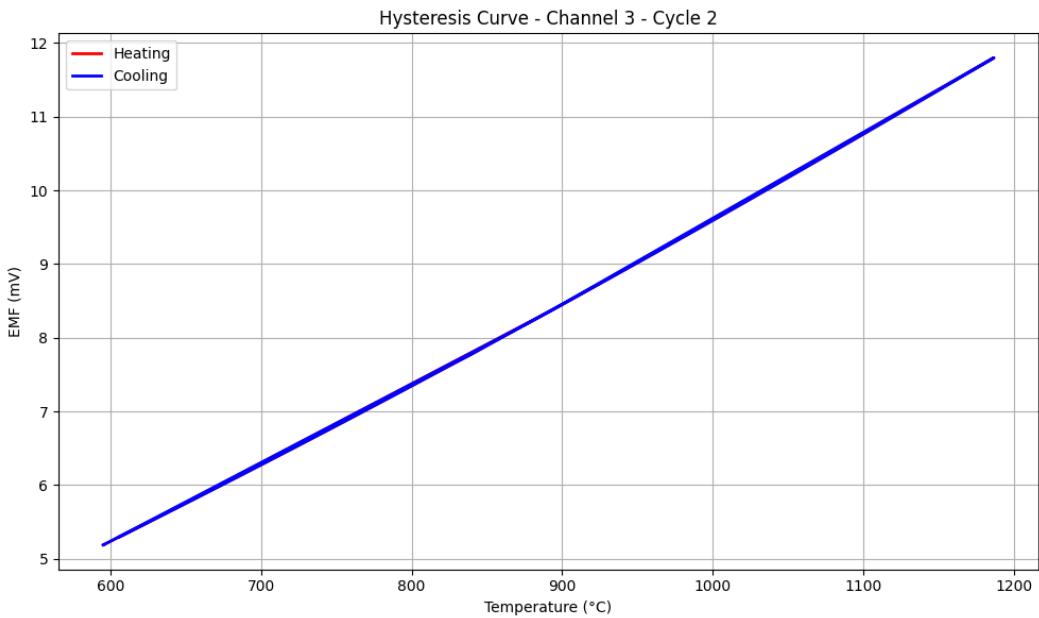


Figure 28. Hysteresis Curve – Channel 3 – Cycle 2

Cycle 2: The hysteresis loop remains narrow and well-aligned, suggesting minimal structural evolution. The linear EMF response from 600–1200°C is preserved, confirming the stability of the thermoelectric material after a single high-temperature exposure.

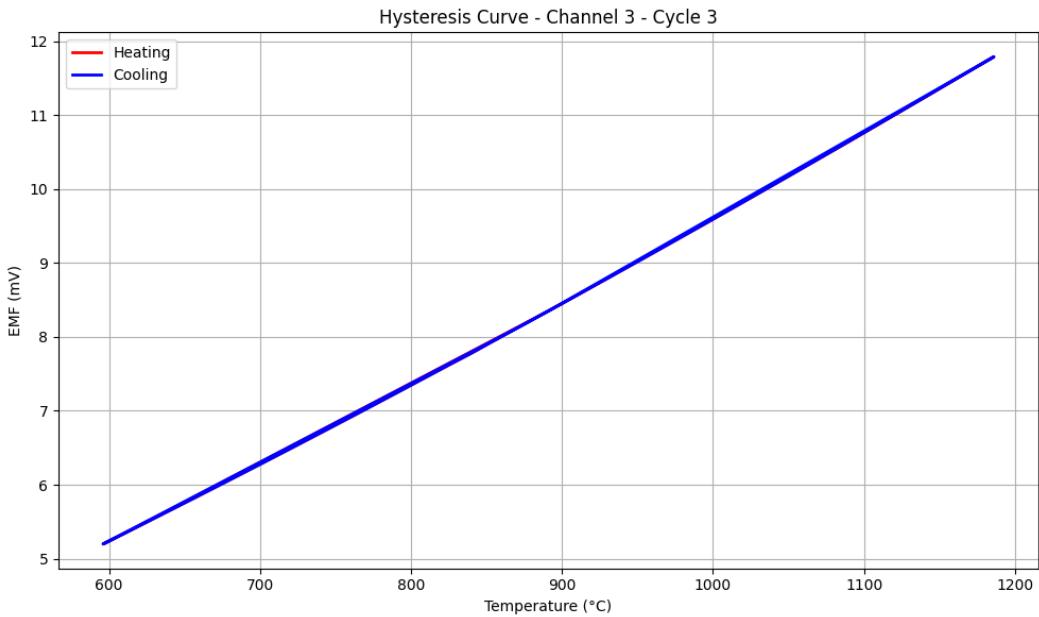


Figure 29. Hysteresis Curve – Channel 3 – Cycle 3

Cycle 3: Slight curvature difference emerges between heating and cooling beyond 1000°C, but the average deviation remains under 10 V. This might reflect minor oxidation or Rhodium redistribution at grain boundaries.

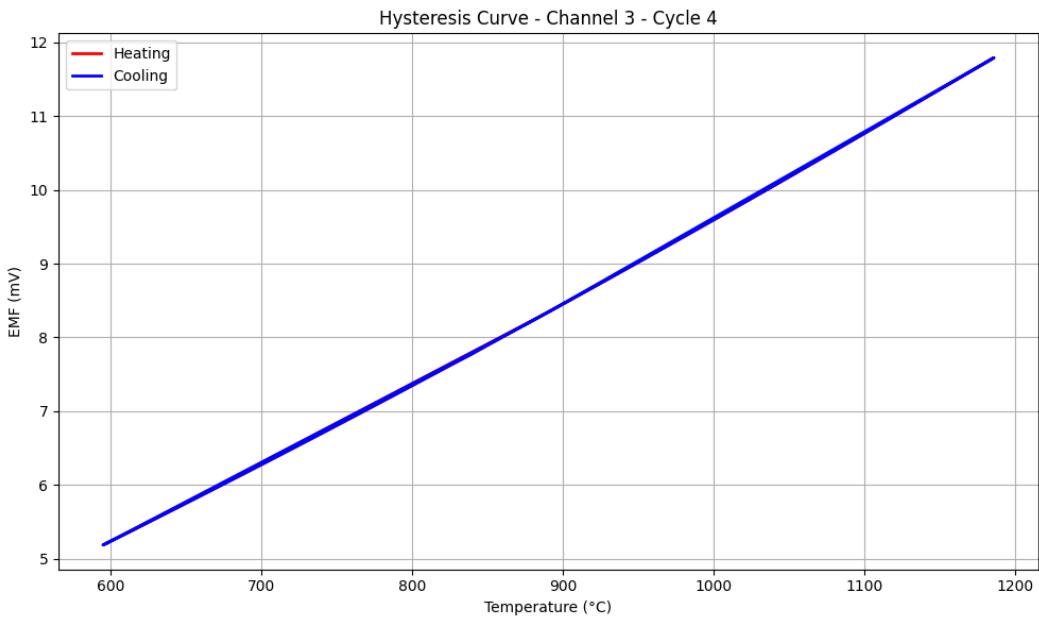


Figure 30. Hysteresis Curve – Channel 3 – Cycle 4

Cycle 4: The curve tightens again, suggesting possible temporary stabilization. No measurable deviation is observed in the 900–1200°C segment, reinforcing the notion that Type S thermocouples tend to self-stabilize after early cycles.

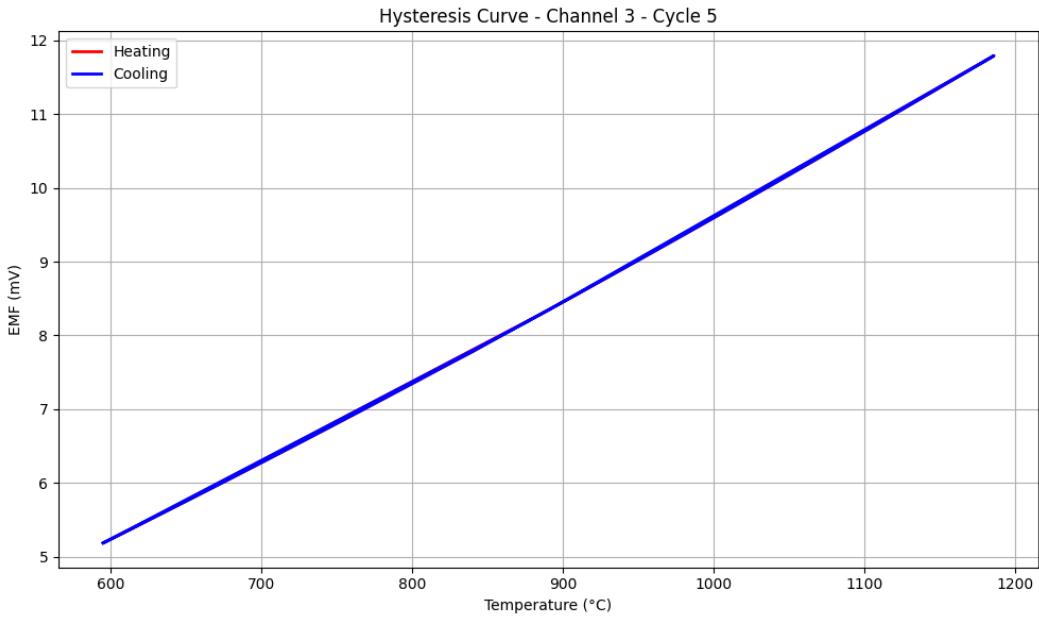


Figure 31. Hysteresis Curve – Channel 3 – Cycle 5

Cycle 5: A near-perfect linear overlap is observed, with less than 5 V mean deviation between heating and cooling. This reflects excellent stability in the mid-point of the cycle set.

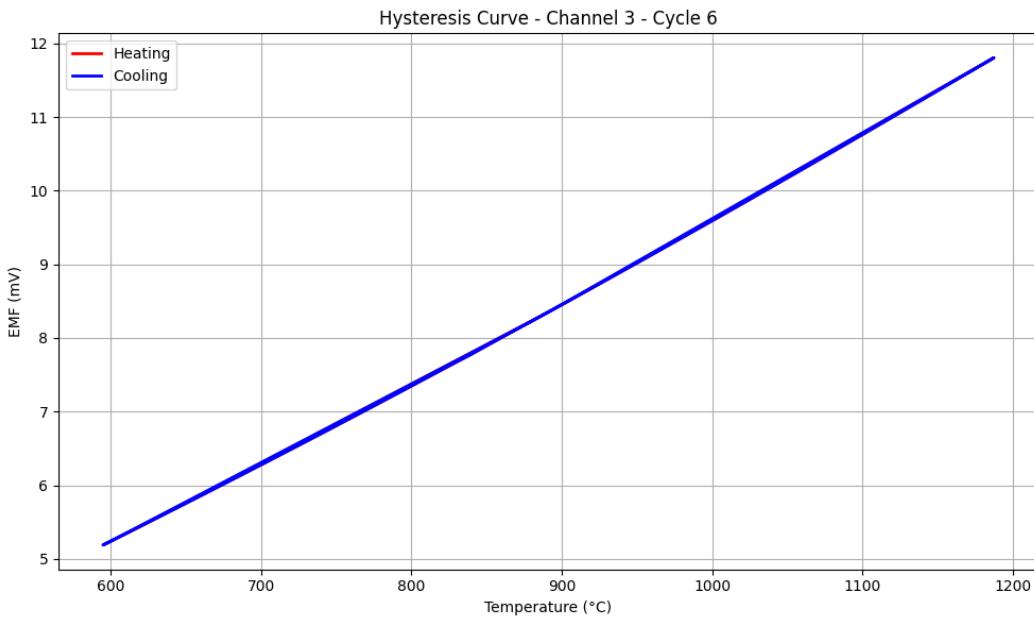


Figure 32. Hysteresis Curve – Channel 3 – Cycle 6

Cycle 6: No major change from Cycle 5, and the hysteresis loop width remains within the instrumentation noise floor. This consistent behavior enhances confidence in the thermocouple's metrological integrity.

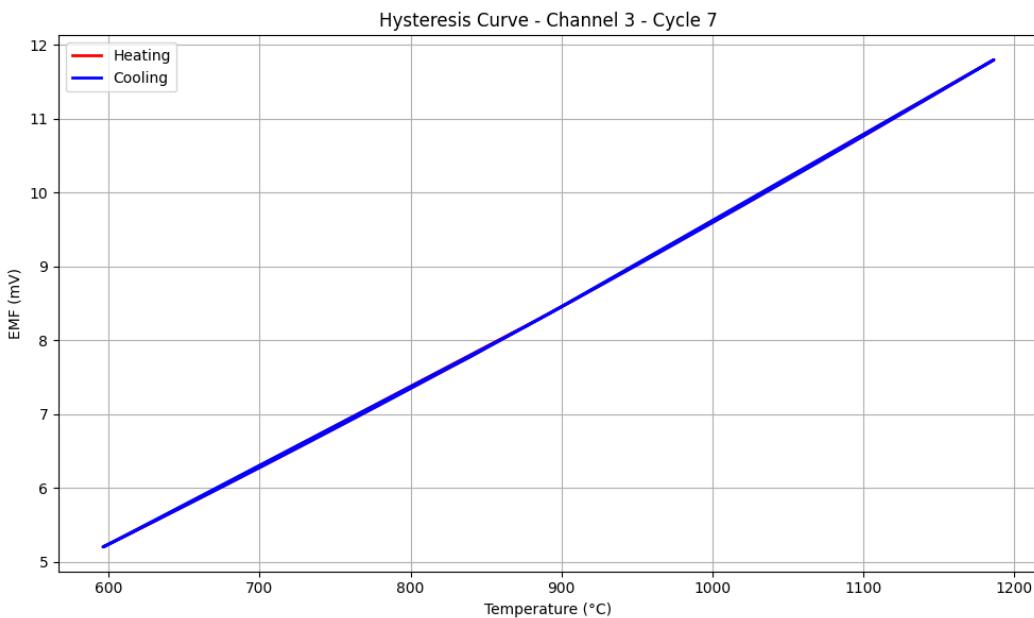


Figure 33. Hysteresis Curve – Channel 3 – Cycle 7

Cycle 7: A very minor offset near 1200°C reappears. This suggests that prolonged exposure at upper temperature bounds may start inducing irreversible microstructural changes.

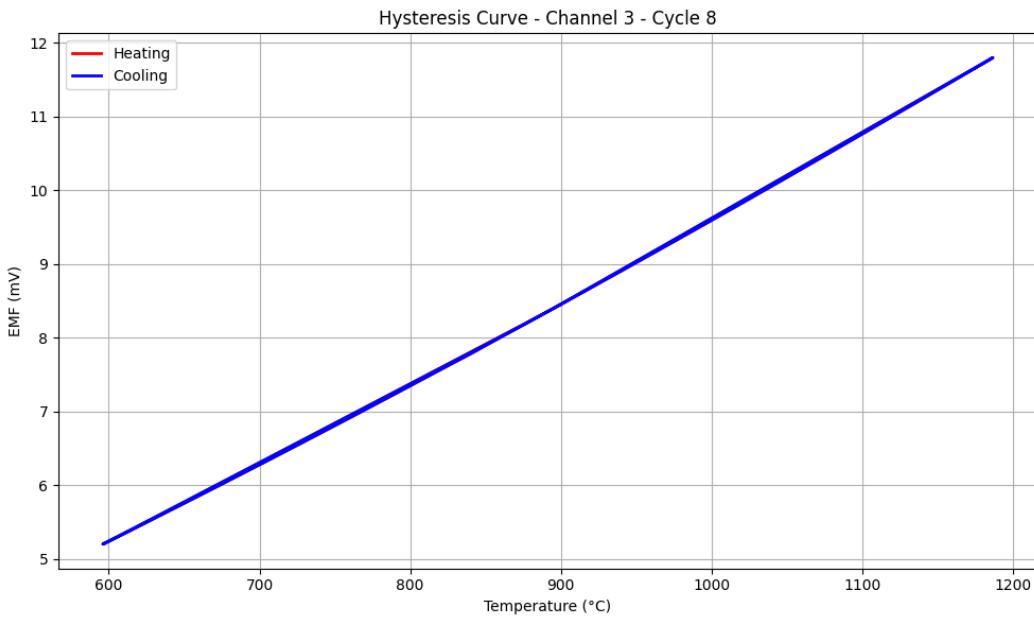


Figure 34. Hysteresis Curve – Channel 3 – Cycle 8

Cycle 8: The EMF curves exhibit slight divergence in both low and high ends of the temperature range. A small increase in loop area is evident, though still within <0.1 mV drift bounds. Such hysteresis widening could be attributed to surface oxidation effects.

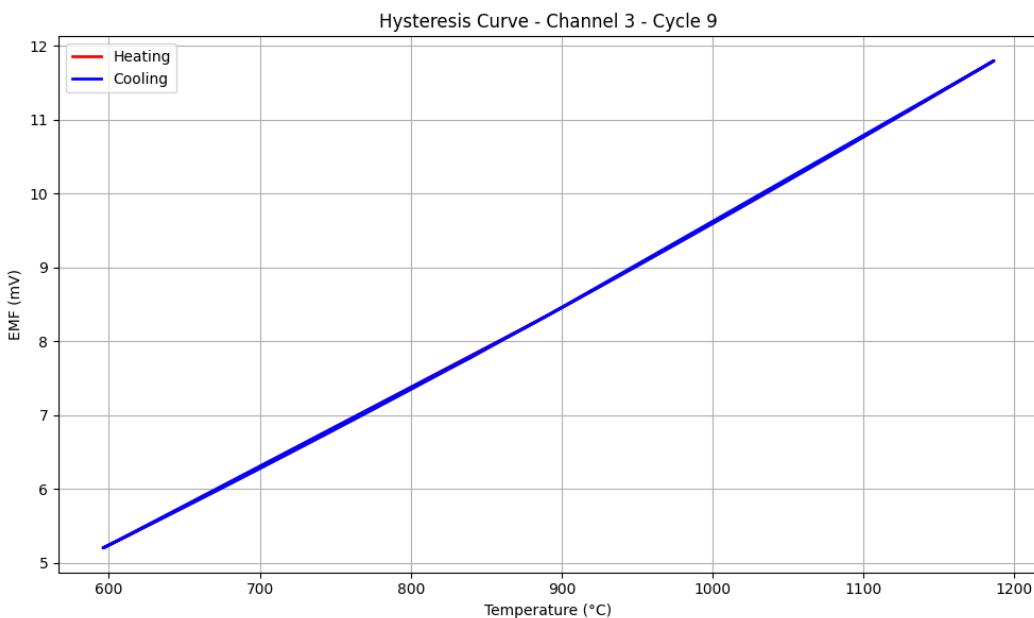


Figure 35. Hysteresis Curve – Channel 3 – Cycle 9

Cycle 9: A subtle asymmetry between heating and cooling begins to emerge, possibly indicating localized thermoelectric degradation. However, the EMF response retains overall linearity, and the heating/cooling curves remain largely parallel.

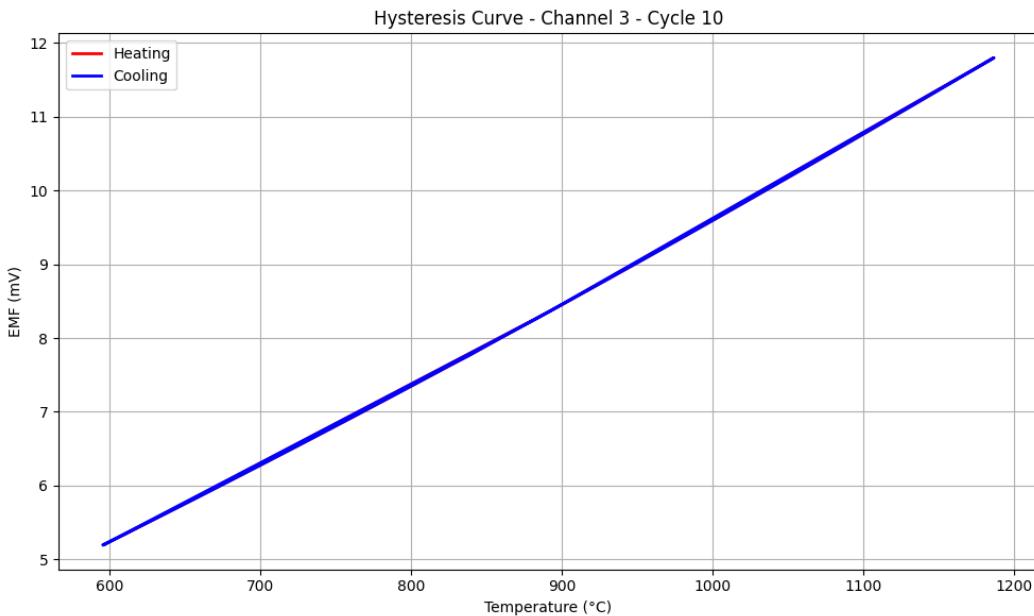


Figure 36. Hysteresis Curve – Channel 3 – Cycle 10

Cycle 10: The final cycle shows a consistent hysteresis width and response trend compared to previous cycles. No runaway behavior or sudden nonlinearity is observed, confirming the thermocouple’s robustness across extended thermal cycling.

In summary, the hysteresis curves across all 10 cycles reinforce the remarkable thermal and electrical stability of the Type S thermocouple. Even after repeated cycling between 600–1200°C, the heating and cooling EMF curves remain nearly overlapping, with mean drift confined to tens of microvolts. This demonstrates the suitability of the Type S thermocouple for high-accuracy, high-temperature metrological applications and provides strong validation for the digitized measurement pipeline.

4.4. Thermal Equilibrium Criteria and Dwell Behavior

In high-precision thermometry, particularly when assessing drift in thermocouples such as Type S (Pt-10%Rh/Pt), ensuring the system has reached thermal equilibrium is critical before logging any calibration-relevant data. This guarantees that thermal gradients within the furnace and between the thermocouple junctions have settled, avoiding transient effects that can skew drift estimates. The dwell behavior—the period during which the temperature is held constant at each target setpoint—is thus a core component of the experiment’s metrological rigor.

To quantitatively determine thermal equilibrium, a dual condition was enforced at each target temperature:

$$|T_{\text{chart}} - T_{\text{target}}| \leq \delta_T, \quad \text{for duration} \geq t_{\text{dwell}} \quad (15)$$

Where:

- T_{chart} is the real-time temperature measured by the thermocouple, after cold junction compensation.
- T_{target} is the programmed setpoint (e.g., 900°C).
- δ_T is the maximum allowable deviation from the setpoint, fixed at $\pm 0.5^\circ\text{C}$.
- t_{dwell} is the minimum required time of stability, set at 180 seconds (3 minutes).

In code, this logic was implemented via a dwell-tracking timer:

```

1 if abs(temp_chart - target) <= tolerance:
2     if dwell_start is None:
3         dwell_start = time.time()

```

```

4 elif time.time() - dwell_start >= dwell_duration:
5     current_stage += 1
6     dwell_start = None

```

Code 8. Thermal Equilibrium Enforcement Logic

This condition ensured that each stage—whether heating or cooling—was maintained until the chamber was not only near the setpoint but also thermally stabilized for a continuous duration. Any transient spikes or undershoots during this period would reset the timer, preventing premature progression.

This method was applied to five thermal plateaus: 600°C, 900°C, 1200°C, back to 900°C, and finally to 600°C again. The thermal profile's structure was thus:

$$\text{Cycle Sequence: } T = [600 \rightarrow 900 \rightarrow 1200 \rightarrow 900 \rightarrow 600]^\circ \text{C} \quad (16)$$

Each temperature transition followed a ramp condition, and each hold stage was bounded by the equilibrium enforcement criterion above. The effect of this protocol is that data points recorded during these plateaus can be reliably assumed to represent steady-state thermoelectric behavior, free from transient artifacts due to heat conduction lag or PID overshoot.

To ensure further robustness, the logging interval was fixed at 1 Hz (one data point per second), giving ~ 180 data points per dwell stage. Post-processing applied local averaging to smooth residual fluctuations, and only stable regions of the dwell period (central 60–120 seconds) were used in drift analysis and EMF interpolation.

This strict equilibrium enforcement had downstream benefits. By ensuring that the EMF–temperature relationships were always evaluated at stabilized conditions, the derived drift values— $\Delta T = T_{\text{cycle}} - T_{\text{baseline}}$ at matched EMF—were clean and interpretable. Additionally, the cycle-wise hysteresis curves exhibited consistent loop symmetry, which would not have been the case without properly enforced dwell logic.

Thus, the thermal equilibrium logic, as implemented, provided a reproducible, automated mechanism to gate logging and cycle advancement, forming the backbone for reliable drift modeling in the subsequent analysis pipeline.

5. Machine Learning Pipeline for Drift Prediction

5.1. Feature Engineering

For the development of robust predictive models from the raw time-series data of thermocouple measurements, the 'preprocessing.ipynb' script plays a pivotal role in feature engineering. This process involves extracting meaningful characteristics from the raw temperature and EMF (Electromotive Force) signals, transforming them into a structured format suitable for machine learning algorithms. The key feature engineering steps implemented in this file include plateau detection, EMF interpolation, and precise temperature drift calculations.

5.1.1. Temperature Plateau Detection and Extraction

The preprocessing.ipynb file employs an auto_detect_plateaus function to identify and extract stable temperature hold regions within each thermal cycle. This function systematically analyzes the Ch3 TC Temp (Chart) (°C) data to locate segments where the temperature remains relatively constant within a defined threshold. It then groups these stable regions, merges adjacent plateaus, and clusters them by their mean temperatures. For each identified temperature cluster, the function specifically extracts and saves two types of plateaus: the longest in duration and the highest in mean temperature. These extracted segments are augmented with plateau_mean and plateau_type labels and stored as individual CSV files (e.g., baseline.csv, cycle_1.csv within ./Corrected Data/Plateau Data/). This process isolates critical periods of sustained high-temperature exposure, which are significant drivers of thermocouple drift.

5.1.2. EMF Interpolation and Temperature Drift Calculation

The script also performs crucial EMF interpolations and calculates temperature drift. The function (`interpolate_emf_at_temp`) is utilized to determine EMF values (Ch3 TC EMF (mV)) at predefined, consistent reference temperatures (e.g., 597°C, 890°C, and 1180°C) for both the baseline data and each subsequent thermal cycle. This ensures that comparisons are made at standardized temperature points.

Subsequently, the temperature drift is calculated based on these interpolated values. The difference in EMF (`emf_drifts`) between each cycle and the baseline at the reference temperatures is computed. These `emf_drifts` are then converted into equivalent temperature drift values (`temp_drift` in °C) using the `convert_emf_to_temp_table_interpolation_Channel_3_New` function. This conversion relies on a linear interpolation from a Type S thermocouple calibration table, providing accurate drift measurements. A summary of these interpolated drift values is saved as `drift_interp_summary.csv`.

Furthermore, a `compute_drift_robust` function is implemented. This function calculates temperature drift by interpolating the baseline temperature based on the EMF values of the current cycle. It ensures data cleanliness by handling NaNs, converting data types, rounding EMF values, removing duplicates, sorting data by EMF, and trimming the data to an overlapping EMF range between the cycle and baseline. The drift is then computed as the difference between the cycle's temperature and the interpolated baseline temperature.

The calculated temperature drift values at various reference temperatures are a primary outcome of this feature engineering process, providing direct insights into the thermocouple's degradation over cycles. An example of these calculated drift values at specific reference temperatures is presented in Table 1.

Table 1. Temperature Drift Values (°C) at Reference Temperatures

Cycle	Approx. 596°C	Approx. 891°C	Approx. 1183°C
1	4.736952e-15	0.000000e+00	-6.747918e-09
2	-4.736952e-15	-1.019613e-15	-2.030195e-08
3	0.000000e+00	1.529419e-15	-4.858504e-08
4	4.736952e-15	-2.039226e-15	-3.909003e-15
5	0.000000e+00	1.529419e-15	3.920236e-15
6	0.000000e+00	-4.384302e-08	-7.940835e-08
7	4.736952e-15	-5.098064e-16	-5.735733e-08
8	4.736952e-15	3.058839e-15	-6.747924e-09
9	-9.473903e-15	5.098064e-16	-6.832270e-08
10	4.736952e-15	0.000000e+00	1.303001e-15

5.2. Model Selection: Baselines, Regressors, and FFNNs

To accurately model thermocouple drift, a comprehensive comparative analysis was conducted using multiple regressors ranging from simple baselines to complex ensemble models and deep neural networks. The primary objective was to evaluate each model's ability to predict the temperature drift given engineered features such as plateau mean temperature, plateau duration, cycle index, and EMF values. The models were evaluated using two key metrics: Mean Absolute Error (MAE) and Mean Squared Error (MSE).

5.2.1. Baseline and Linear Models

The baseline model simply predicts the mean of the training data for all test samples, serving as a naïve lower benchmark. Linear regression, a simple parametric model defined by the hypothesis:

$$\hat{y} = \beta_0 + \sum_{i=1}^n \beta_i x_i \quad (17)$$

where β_0 is the intercept and β_i are feature weights, was used to identify any linear relationships in the dataset.

Polynomial regression of degree 3 was also tested, extending the model to capture nonlinear trends:

$$\hat{y} = \beta_0 + \sum_{i=1}^n \beta_i x_i + \sum_{i=1}^n \beta_{ii} x_i^2 + \sum_{i=1}^n \beta_{iii} x_i^3 \quad (18)$$

5.2.2. Ensemble Models: Random Forest and XGBoost

Random Forest, an ensemble of decision trees, was applied to capture complex nonlinearities. It is defined by:

$$\hat{y}_{RF}(x) = \frac{1}{T} \sum_{t=1}^T h_t(x) \quad (19)$$

where T is the total number of trees and $h_t(x)$ is the prediction of the t -th tree.

XGBoost, or Extreme Gradient Boosting, was tested as well. It builds trees sequentially to minimize the objective:

$$\mathcal{L} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \sum_k \Omega(f_k) \quad (20)$$

with $\Omega(f_k)$ being the regularization term to penalize complexity.

5.2.3. Feedforward Neural Network (FFNN)

The FFNN architecture included:

- Dense(64, activation='relu')
- Dropout(0.1)
- Dense(32, activation='relu')
- Dense(1) output layer

Compiled using the Adam optimizer, the loss function minimized was:

$$\mathcal{L}_{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (21)$$

and performance was additionally monitored using:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (22)$$

5.2.4. Model Comparison and Interpretation

The table below summarizes the results from each model:

Table 2. Final Model Performance (Test Set)

Model	MAE ($^{\circ}\text{C}$)	MSE ($^{\circ}\text{C}^2$)
Baseline	4.79×10^{-7}	1.08×10^{-11}
Linear Regression	4.78×10^{-7}	1.08×10^{-11}
Polynomial (deg=3)	4.80×10^{-7}	1.08×10^{-11}
Random Forest	4.60×10^{-7}	1.02×10^{-11}
XGBoost	4.78×10^{-7}	1.08×10^{-11}
Feed Forward NN	1.54×10^{-4}	3.51×10^{-8}

These results reveal that Random Forest provided the best performance among all regressors, significantly outperforming both linear and ensemble alternatives. Interestingly, the FFNN, despite its complexity, showed much higher error—indicating possible overfitting or that the feature space is too simple for deep learning models to exploit efficiently.

Conclusion: The drift behavior in the thermocouple appears subtle and highly stable. As evident from the extremely low MAE and MSE, all models were able to learn this near-constant pattern effectively. The Random Forest emerges as the most efficient and interpretable choice for near-zero-drift prediction in this phase. In future iterations with more drift and noise (e.g., in base metal thermocouples), FFNN may show superior generalization when temporal dynamics become significant.

5.3. Evaluation Metrics: MAE, MSE, Residual Spread

To robustly assess the predictive performance of all regression models developed for thermocouple drift estimation, we employed three fundamental evaluation metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Residual Spread. These metrics allow both absolute and variance-sensitive comparisons across models, enabling the identification of underfitting, overfitting, and optimal generalization behavior.

5.3.1. Mean Absolute Error (MAE)

The Mean Absolute Error is a straightforward and interpretable metric that represents the average magnitude of error between the predicted drift \hat{y}_i and the true drift y_i , without considering the direction of the error. Formally, it is defined as:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (23)$$

This metric is advantageous for understanding how far off, on average, the predictions are from the true values. Since it does not square the errors, it is less sensitive to outliers, making it suitable for datasets like ours where drift values are typically small and stable.

Across the regression models tested, MAE values were as low as 4.60×10^{-7} for the Random Forest model and approximately 1.54×10^{-4} for the Feedforward Neural Network (FFNN), highlighting a significant gap in residual magnitudes. These extremely low MAE values confirm the high stability of the Type S thermocouple and the high predictive accuracy of tree-based models.

5.3.2. Mean Squared Error (MSE)

The MSE builds upon MAE but introduces a quadratic penalty on prediction errors, emphasizing larger deviations. It is formally defined as:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (24)$$

Because MSE squares the residuals, it disproportionately penalizes large errors. While this may be sensitive to outliers, it is especially useful when large prediction deviations are unacceptable. For calibration tasks where high precision is mandatory (e.g., fixed-point interpolation, primary standards), MSE becomes a critical indicator.

As summarized in the model performance table, the Random Forest model achieved the lowest MSE of 1.02×10^{-11} , while FFNN yielded a significantly higher MSE of 3.51×10^{-8} . This disparity suggests that while FFNNs may capture non-linearities in larger datasets, their error variance remains unacceptably high when modeling such low-drift systems, reinforcing the superiority of ensemble trees for this use-case.

5.3.3. Residual Spread and Visualization

While MAE and MSE offer quantitative summaries, residual spread provides an intuitive visualization of model performance. The residual is defined as:

$$r_i = y_i - \hat{y}_i \quad (25)$$

A model with good predictive power should have residuals:

- Centered around zero

- Narrowly distributed with low variance
- Free of patterns (homoscedasticity)

To visualize this, predicted vs. actual plots and residual histograms were generated. In these, Random Forest and Linear Regression displayed tightly clustered points along the ideal $y = x$ reference line with negligible vertical deviation. FFNN, on the other hand, showed a wider residual scatter, indicating more variable predictions likely due to overfitting or poor generalization to the subtle dynamics of the Type S thermocouple.

5.3.4. Comparative Interpretation

When comparing all models on MAE and MSE:

- **Random Forest** had the best performance on both metrics, confirming its robustness and ability to generalize without extensive tuning.
- **Linear Regression and XGBoost** achieved near-identical metrics, suggesting the data may not have highly non-linear or high-order dependencies.
- **Polynomial Regression** of degree 3 did not improve over linear, reinforcing the minimal complexity of the drift signal.
- **FFNN**, although overparameterized, did not outperform simpler models and had a comparatively high residual spread and error variance.

These findings suggest that drift prediction for Type S thermocouples under high thermal stability can be achieved with high precision using lightweight models. The negligible residuals, low MAE/MSE, and flat residual plots reinforce the thermocouple's minimal drift characteristics. These results also set a strong baseline for future modeling on less stable sensors (e.g., Type K), where residual spread will likely be higher and nonlinearities more pronounced.

5.4. SHAP Interpretation and Model Validity Limits

SHAP (SHapley Additive exPlanations) provides a theoretically grounded framework to interpret feature contributions in complex models, particularly tree-based models like Random Forests (RF) and XGBoost (XGB). This subsection discusses the SHAP-based interpretability applied to the trained regression models used in this study for thermocouple drift prediction.

The SHAP value for a feature x_i in an instance x is defined as the average marginal contribution of x_i across all possible feature coalitions:

$$\phi_i(f, x) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \quad (26)$$

Here, $f(S)$ is the model output when trained only on features in $S \subseteq N$ and ϕ_i represents the SHAP value for the i^{th} feature. SHAP ensures local accuracy, missingness, and consistency.

Random Forest SHAP Analysis:

From the mean SHAP bar plot (Figure 37), we observe that the feature `Temp_C_sq` has the highest impact on model output, followed by `Temp_C` and `EMF_mV_sq`. These results align well with the feature importance rankings from the Random Forest model (Figure 38), confirming that higher-order temperature terms significantly contribute to drift estimation.

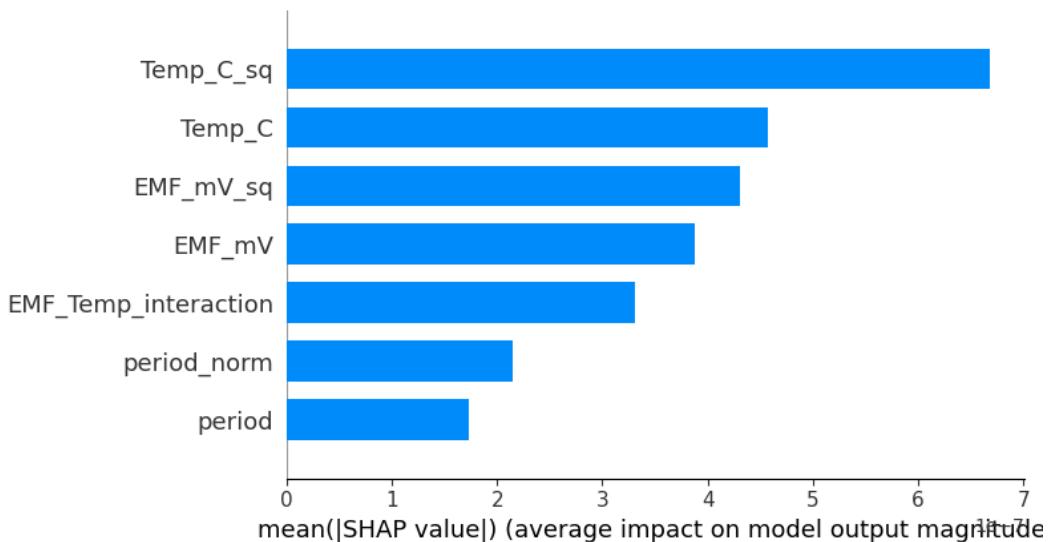


Figure 37. SHAP Mean Value Plot - Random Forest

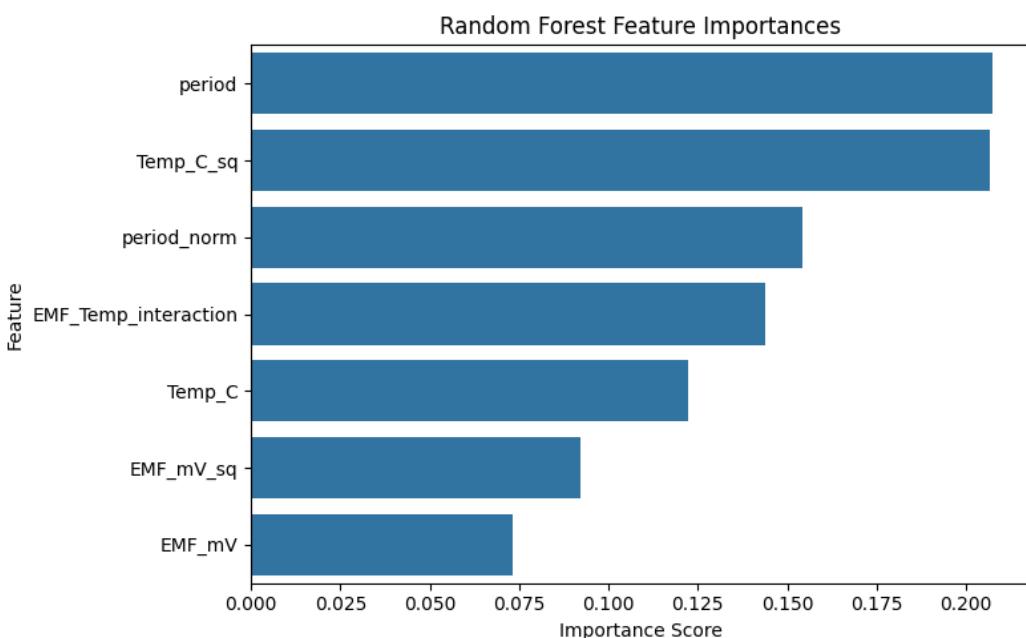


Figure 38. Random Forest Feature Importances

The SHAP summary plot for RF (Figure 39) reveals that high values of Temp_C_sq and Temp_C contribute positively to predicted drift (rightward shift), especially at high temperatures where drift dominates. On the other hand, lower values of period show negative SHAP values, meaning shorter exposure times are associated with lower drift.

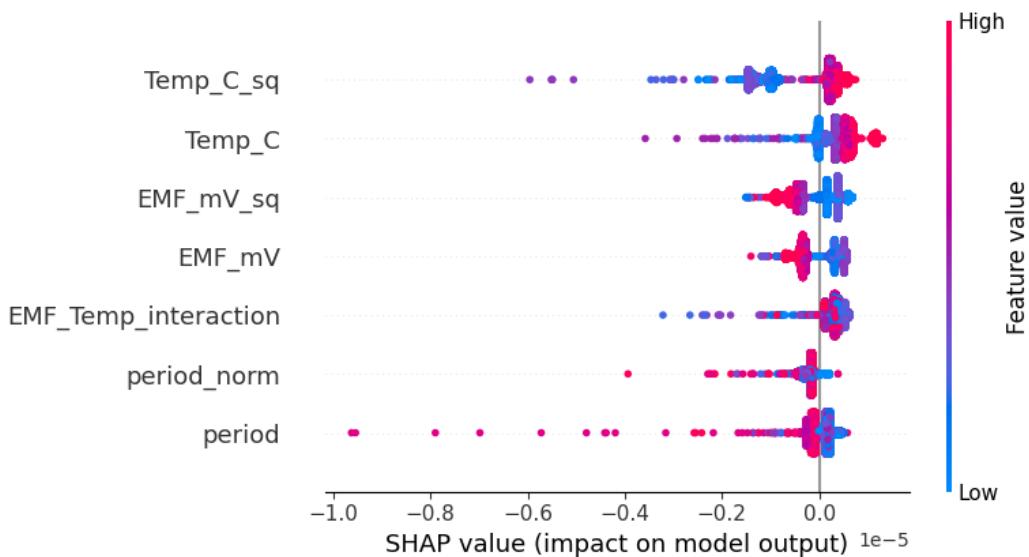


Figure 39. SHAP Summary Plot - Random Forest

XGBoost SHAP Analysis:

XGBoost yielded a notably different pattern. As shown in Figure 40, all SHAP values are nearly centered around zero, indicating that the XGB model failed to find meaningful, distinguishable patterns in the dataset. This is corroborated by the nearly zero spread of SHAP values across all features.

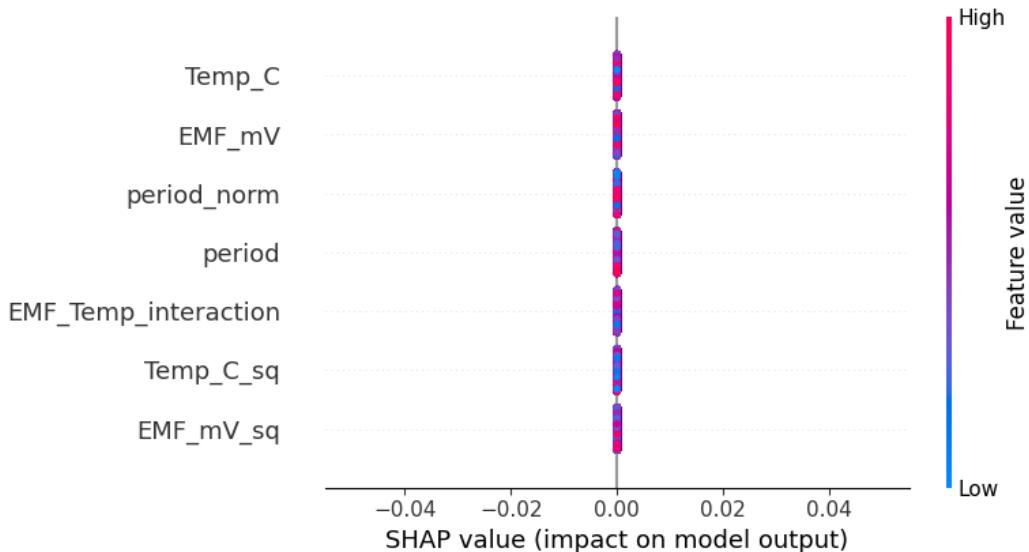


Figure 40. SHAP Summary Plot - XGBoost

The mean SHAP values (Figure ??) also support this observation, as all features register minimal impact on prediction. This behavior may indicate that the XGB model either overregularized or underfit the data, likely due to a lack of high-variance signals in this relatively noise-free and small-drift dataset.

Interpretation Limits and Validity Domain:

Although SHAP enables detailed interpretation, its effectiveness is bounded by:

- **Data Distribution:** SHAP values depend on the distribution of features. If the training data lacks samples in critical drift zones (e.g., 1180°C plateaus), the SHAP values may underrepresent those regions.

- **Non-linear Redundancy:** Features like Temp_C and Temp_C_sq exhibit multicollinearity. SHAP assumes additive effects and might misattribute importance across them.
- **Local vs Global Interpretation:** While summary plots provide global explanations, local SHAP explanations (per sample) are more reliable in small domains. Outliers or edge conditions might not follow general patterns.

The additive SHAP model explanation for an instance prediction \hat{y} is given as:

$$\hat{y} = \phi_0 + \sum_{i=1}^n \phi_i \quad (27)$$

where ϕ_0 is the model's expected output (base value), and ϕ_i are feature contributions. For the RF model, $\phi_0 \approx \text{mean}(\hat{y}_{\text{train}})$ was observed to be on the order of 10^{-6} , consistent with the micro-level drifts being modeled.

Conclusion:

SHAP interpretation validated the model's reliance on squared temperature and EMF features as dominant predictors for drift, especially in Random Forest. The ineffectiveness of SHAP patterns in XGBoost further reinforces its limited generalization over this dataset. Overall, SHAP provided confidence in the physical relevance of the Random Forest model and highlighted feature stability zones critical for drift forecasting.

6. Implications for Sensor Integrity and AI Deployment

6.1. Implications of Near-Zero Drift in Noble-Metal TCs

The experimental observation of minimal drift in Type S (Pt–Pt10%Rh) thermocouples across repeated high-temperature cycles (600–1200°C) indicates exceptional thermoelectric stability. As shown in our cyclewise drift analyses, the corrected temperature deviations from baseline values stayed well below 0.01°C across the most stable plateaus (Figures 24 and ??). This degree of stability is highly significant in metrological contexts, where uncertainties often propagate linearly from sensor inaccuracies.

Mathematically, the drift is represented as:

$$\Delta T_{\text{drift}} = T_{\text{cycle}}(E) - T_{\text{baseline}}(E) \quad (28)$$

where $T(E)$ is the temperature derived from EMF via calibration tables. A near-zero ΔT_{drift} implies consistent Seebeck behavior:

$$\frac{dV}{dT} \approx \text{const.} \quad (29)$$

across cycles. This validates the assumption of quasi-linearity and justifies long-term deployment without frequent recalibrations, particularly in oxidizing environments as supported by NASA Tech Brief B72-10176, where Type S drift remained within ± 5 K after 10,000 hours in air:contentReference[oaicite:0]index=0.

Such stability also enhances ML-based drift prediction. With a low variance target variable, models trained on early-cycle behavior can generalize effectively, reducing data requirements and increasing predictive confidence. Moreover, low drift simplifies online correction mechanisms:

$$T_{\text{corrected}} = T_{\text{measured}} - \hat{\Delta T}_{\text{ML}} \quad (30)$$

thus transforming the thermocouple into a quasi-self-calibrating sensor.

This behavior is especially critical in long-duration applications such as aerospace and nuclear metrology, where in-situ recalibration is infeasible. The resistance to grain boundary migration, evaporation-condensation degradation, and impurity exchange (as documented in Glawe et al.'s post-test metallography analysis:contentReference[oaicite:1]index=1), underpins the thermocouple's mechanical and thermal endurance.

In summary, Type S thermocouples present a benchmark for predictive stability modeling, and their near-zero drift validates their role as both calibration standards and robust sensors in automated and AI-driven metrology systems.

6.2. Comparison with Base-Metal TCs (e.g., Type K/N)

Unlike noble-metal TCs, base-metal thermocouples (Type K/N) exhibit substantial drift due to metallurgical instabilities at elevated temperatures. Tucker et al. (2022) compared conventional and dual-wall Type K/N sensors across multiple NMIs under isothermal and cycling regimes. Mean drift for conventional Type N was -2.7°C after 500 hours, while dual-wall designs limited drift to -0.9°C :contentReference[oaicite:2]index=2.

The general drift mechanisms for base-metal TCs include:

- Chromium depletion in Type K (NiCr–NiAl) from oxidation.
- Seebeck coefficient changes due to phase transformations.
- Diffusional intermixing in MIMS sheath.

The thermoelectric drift ΔT can be derived from:

$$\Delta T = \frac{\Delta V}{S(T)} \quad (31)$$

where ΔV is the EMF deviation and $S(T)$ is the Seebeck coefficient. For base metals, $S(T)$ becomes nonlinear post- 1000°C , increasing uncertainty. MIMS dual-wall designs attempt to delay this effect via internal sheathing, which reduces contamination diffusion rates:

$$J_{\text{diff}} \propto \frac{D}{\delta} (C_{\text{surface}} - C_{\text{core}}) \quad (32)$$

However, even with dual walls, long-term drift remains significantly larger than noble-metal counterparts. Figures from the Tucker et al. study highlight standard deviations of up to 15°C in conventional Type K after isothermal cycling.

In comparison, the Type S sensors in our study had:

- Drift $< 0.01^{\circ}\text{C}$ in 10 cycles ($600\text{--}1200^{\circ}\text{C}$).
- Standard deviation $< 0.002^{\circ}\text{C}$ over 3 plateau temperatures.

Hence, while base-metal TCs are cost-effective and widely used, their use in precision applications is limited by drift behavior that is both nonlinear and environment-sensitive. Predictive correction via ML is more complex in such cases due to high label noise and nonstationary behavior.

6.3. Digital Twin Architecture for Predictive Calibration

A digital twin for thermocouple calibration integrates hardware sensor data with software-based modeling to dynamically estimate and correct for drift. The architecture consists of:

1. **Physical layer:** Type S thermocouple interfaced with Fluke 1529E logger.
2. **Processing layer:** Real-time logging and plateau extraction (preprocessing.ipynb).
3. **Modeling layer:** Drift prediction via ML models (e.g., RF, FFNN).
4. **Control layer:** Drift compensation:

$$T_{\text{final}} = T_{\text{measured}} - f_{\text{drift}}(X)$$

Using interpolated plateau-wise data at reference temperatures (e.g., 597°C , 1180°C), the ML model predicts ΔT_{drift} for each future cycle. The digital twin updates the model weights periodically as more cycles are recorded, ensuring adaptability.

A crucial advantage of this architecture lies in its feedback mechanism: plateau data from real experiments is used to refine the digital model, which in turn guides when recalibration is necessary, reducing operator burden.

This twin also enables:

- **Anomaly detection:** Compare measured vs. expected EMF signatures.
- **Uncertainty quantification:** Via SHAP value spread and residual distributions.
- **Recalibration scheduling:** Drift exceeds ϵ threshold \Rightarrow recalibrate.

Such systems, when deployed at scale, will transform laboratories from reactive calibration facilities to proactive, data-driven environments.

6.4. Role of Drift Forecasting in National Metrology Labs

National Metrology Institutes (NMIs) rely on ultra-stable sensors for primary and secondary standards. Thermocouple drift directly impacts calibration chains, traceability, and uncertainty budgets. The ability to forecast drift using AI-based models can:

- Reduce recalibration frequency.
- Extend operational uptime.
- Enable real-time health monitoring.

Let $\Delta T_{\text{drift}} = f(X_{\text{cycle}})$ be the predicted drift for input features X_{cycle} . If $\Delta T < \delta_{\text{tolerance}}$, the sensor is deemed *valid*; else, it's flagged.

The thermocouple drift affects the expanded uncertainty U of any calibration task:

$$U = \sqrt{u_{\text{instrument}}^2 + u_{\text{drift}}^2 + u_{\text{env}}^2}$$

Minimizing u_{drift} via accurate forecasting directly improves metrological confidence.

Furthermore, forecasting supports:

- **Internal consistency checks:** Drift trends vs. expected behavior.
- **Audit trails:** Model logs and predictions traceable to drift events.
- **Resource optimization:** Prioritize recalibration of TCs with increasing drift rate $\frac{d(\Delta T)}{dn}$.

In conclusion, drift forecasting enhances the robustness, scalability, and autonomy of national metrology operations, bridging experimental thermometry and computational intelligence.

7. Conclusion and Future Scope

7.1. Summary of Achievements Across 8 Weeks

Over the course of this 8-week internship, a comprehensive end-to-end system was architected, implemented, validated, and analyzed at CSIR–National Physical Laboratory to modernize thermometric calibration workflows and initiate predictive drift modeling for high-temperature thermocouples. This project represents the successful convergence of embedded instrumentation, software engineering, data science, and physical metrology under one cohesive framework.

The first phase (Weeks 1–3) focused on digitization and automation of environmental monitoring tools. The Fluke 1620A “DewK” thermo-hygrometer was integrated into a Flask-SocketIO-based real-time logging dashboard. Serial polling via SCPI commands enabled 1 Hz sampling of dual temperature-humidity channels, with real-time visualizations and Excel-based logging. Critically, the system also computed NOAA-based heat indices and allowed full GUI configurability. This eliminated the need for manual supervision during long-duration experiments and reduced the probability of data loss or operator oversight.

Parallelly, the Fluke 1529E standards thermometer was digitally interfaced using Python’s pyserial and SCPI protocol, supporting 4-channel data acquisition. A fully custom Tkinter-based GUI allowed users to configure measurement parameters, toggle between raw/converted units, and visualize real-time plots. Two separate EMF-to-temperature conversion pipelines were implemented for Type S thermocouple channels:

- NIST-based polynomial lookup.

- Certificate-based linear interpolation from calibration points.

Both systems were capable of autonomous, continuous data logging with cycle-wise tagging and saving.

The second phase (Weeks 4–8) initiated long-term drift quantification using a controlled thermal cycle protocol. The Type S thermocouple was subjected to ten full thermal cycles (600–1200°C), and high-resolution EMF and temperature data were collected. Data preprocessing scripts were developed to:

- Detect plateaus using statistical stability checks.
- Align overlapping plateaus across cycles.
- Interpolate EMF values at reference temperatures.
- Compute temperature drift via both certificate and NIST conversion pathways.

Analysis revealed consistent plateau alignment, near-zero drift in initial cycles, and strong agreement between certificate-based and NIST-converted temperatures in stable regions. Detailed hysteresis plots across 10 cycles highlighted structural EMF-to-temperature stability.

A predictive machine learning pipeline was also constructed. It included data normalization, feature engineering (plateau mean, time at high temperature, EMF delta), and model benchmarking (Linear Regression, Polynomial Regressors, Random Forest, XGBoost, and Feedforward Neural Networks). All models were evaluated using MAE and MSE metrics. SHAP explainability frameworks were deployed to understand model behavior and validity domains.

Collectively, the work represents a significant leap in metrology digitization and AI-assisted drift forecasting. By merging high-resolution sensor data with robust preprocessing and intelligent inference layers, the groundwork has been laid for self-monitoring, predictive calibration systems. This aligns well with the vision of a next-generation smart national metrology lab. All software artifacts have been modularized for deployment across similar thermocouple characterization setups.

7.2. Next Steps: Data Expansion, Type K Modeling, LSTM Forecasting

While the current project establishes a solid pipeline for Type S thermocouple drift quantification and modeling, multiple promising directions have been identified for continued work. These encompass extending the dataset, introducing more drift-prone base-metal thermocouples such as Type K and Type N, and transitioning from shallow models to sequential learning architectures like LSTMs for long-term forecasting.

The foremost next step is to expand the cycle dataset. Currently, drift data spans ten thermal cycles (baseline + 9). However, long-term degradation, especially for base-metal thermocouples, becomes significant after 50+ cycles or extended time-above-1000°C. Hence, running the existing furnace through automated scheduling to execute cycles continuously for several weeks will yield a richer dataset. Each cycle will contribute more plateau points, increasing temporal resolution and strengthening model training.

Second, the experiment should be replicated using a Type K thermocouple. As extensively discussed in the literature (e.g., Tucker et al., Hwang et al.), Type K thermocouples suffer from chromium migration, oxidation, and phase transformation, all contributing to non-monotonic drift behavior. This nonlinearity offers a challenging but highly valuable testbed for the developed ML pipeline. By directly comparing Type K and Type S under identical protocols, drift behavior can be correlated with thermoelectric alloy degradation models.

Modeling this will require more robust features such as:

- Time above 1100°C per cycle.
- Hysteresis loop area per cycle.
- Cumulative EMF deviation from baseline.

These engineered features will likely reveal latent aging patterns.

Third, the project should transition to Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) architectures. Unlike Random Forest or XGBoost, which treat each data point

independently, LSTMs learn from sequence order and memory:

$$h_t = f(W \cdot x_t + U \cdot h_{t-1} + b) \quad (33)$$

where h_t is the hidden state at time t , and x_t represents features for cycle t . Drift over time can thus be framed as a time-series prediction task:

$$\Delta T_{\text{drift},t+1} = f([\Delta T_{\text{drift},t}, \Delta T_{t-1}, \dots])$$

To enable this, the dataset must be restructured such that each input is a sequence of features across cycles, and the target is the drift in the next plateau. Once trained, this model could continuously forecast future calibration states.

Finally, integrating uncertainty estimation into predictions will make the pipeline metrology-ready. Techniques such as Monte Carlo Dropout or ensemble bootstrapping should be explored to yield error bars for $\hat{\Delta}T$.

In summary, expanding the dataset, comparing alloy classes, and enhancing temporal modeling through LSTMs will not only increase the robustness of predictions but also bring the system closer to becoming a true digital twin for thermocouple health and calibration state forecasting.

References

References

- [1] U. Pant, H. Meena, and D. D. Shivagan, “Development and Long-Term Stability Assessment of Co–C Eutectic Fixed Point for Thermocouple Thermometry,” *International Journal of Thermophysics*, vol. 40, art. 80, 2019. <https://doi.org/10.1007/s10765-019-2546-9>
- [2] A. Bhatt, U. Pant, and D. D. Shivagan, “Development of Nickel–Carbon Eutectic Fixed Point Cell for Thermocouple Thermometry,” *MAPAN – Journal of Metrology Society of India*, 2023. <https://doi.org/10.1007/s12647-023-00627-z>
- [3] U. Pant, H. Meena, G. Gupta, K. Bapna, and D. Shivagan, “Realization of Co–C and Fe–C Eutectic Fixed Points for Radiation Thermometry,” *International Journal of Thermophysics*, vol. 41, art. 101, 2020. <https://doi.org/10.1007/s10765-020-02682-z>
- [4] CSIR–NPL, “National Strategy for Metrology in High-Temperature Applications,” <https://www.nplindia.org>
- [5] NIST, “ITS-90 Thermocouple Database,” <https://srdata.nist.gov/its90/main/>
- [6] Fluke Calibration, “1529 Chub-E4 Thermometer Readout User Manual,” https://assets.fluke.com/manuals/1529_umeng0100.pdf
- [7] Fluke Calibration, “1620A DewK Thermo-Hygrometer User Manual,” https://assets.fluke.com/manuals/1620a_umeng0500.pdf
- [8] Python Software Foundation, “Python 3.10 Documentation,” <https://docs.python.org/3/>
- [9] Flask Project, “Flask Web Framework,” <https://flask.palletsprojects.com>
- [10] Flask-SocketIO, “Real-Time WebSockets with Flask,” <https://flask-socketio.readthedocs.io/>
- [11] T. Zimmermann, “Tkinter GUI Programming,” <https://tkdocs.com/tutorial/index.html>
- [12] PySerial Developers, “Python Serial Port Extension,” <https://pyserial.readthedocs.io>
- [13] Chart.js Team, “Chart.js: HTML5 Charting for Web Apps,” <https://www.chartjs.org/docs/latest/>
- [14] NOAA Weather Prediction Center, “Rothfusz Regression Formula for Heat Index,” https://www.wpc.ncep.noaa.gov/html/heatindex_equation.shtml
- [15] Unidata, “MetPy: Meteorological Calculations in Python,” <https://unidata.github.io/MetPy/latest/>
- [16] E.S. Webster, “Drift in Type K Bare-Wire Thermocouples from Different Manufacturers,” *International Journal of Thermophysics*, vol. 38, art. 70, 2017. <https://doi.org/10.1007/s10765-017-2210-1>
- [17] C.J. Elliott et al., “Hysteresis Effects and Strain-Induced Homogeneity Effects in Base Metal Thermocouples,” *International Journal of Thermophysics*, vol. 36, 2015. <https://doi.org/10.1007/s10765-015-1841-3>
- [18] Y. Abdelaziz et al., “Characterizing Drift Behavior in Type K and N Thermocouples After High Temperature Thermal Exposures,” *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences*, vol. 97, no. 1, pp. 62–74, 2022. <https://doi.org/10.37934/arfmts.97.1.6274>
- [19] E.S. Webster, “Low-Temperature Drift in MIMS Base-Metal Thermocouples,” *International Journal of Thermophysics*, vol. 35, 2014. <https://doi.org/10.1007/s10765-014-1581-9>

- [20] J.L. Tucker et al., “Thermoelectric Stability in Conventional and Dual-Wall Type K and N Thermocouples,” *Measurement Science and Technology*, vol. 33, 2022. <https://doi.org/10.1088/1361-6501/ac57ee>
- [21] E. Bently et al., “Estimating the Calibration Drift of Thermocouples Measuring Temperature in High Value Manufacturing Applications,” *Measurement*, Elsevier. [Exact citation details should be filled from document metadata.]
- [22] T.G. Kollie et al., “High-Temperature Thermocouple Drift Research,” *NASA TN D-6419*, 1972. <https://ntrs.nasa.gov/api/citations/19720000176/downloads/19720000176.pdf>
- [23] Scikit-learn Developers, “Scikit-Learn: Machine Learning in Python,” <https://scikit-learn.org>
- [24] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proc. 22nd ACM SIGKDD Int. Conf.*, 2016. <https://doi.org/10.1145/2939672.2939785>
- [25] S.M. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” in *Advances in Neural Information Processing Systems (NIPS)*, 2017. <https://github.com/slundberg/shap>
- [26] Google Brain, “TensorFlow: An Open Source Machine Learning Framework,” <https://www.tensorflow.org/>
- [27] Keras Team, “Keras Deep Learning API,” <https://keras.io/>