



## TEMPERATURE DATA ANALYSIS

<sup>1</sup>N.Bhargavi, <sup>2</sup>P.Roshan, <sup>3</sup>L.Bhuvan Chandra, <sup>4</sup>G.Sachin

<sup>1</sup>Assistant Professor, <sup>2,3,4</sup>Students

Department of CSE(Data Science)

Siddhartha institute of technology & sciences, narapally

[bhargavi.cse@siddhartha.co.in](mailto:bhargavi.cse@siddhartha.co.in), [23TQ1A6739@siddhartha.co.in](mailto:23TQ1A6739@siddhartha.co.in),  
[23TQ1A6722@siddhartha.co.in](mailto:23TQ1A6722@siddhartha.co.in), [23TQ1A6703@siddhartha.co.in](mailto:23TQ1A6703@siddhartha.co.in)

### ABSTRACT

Temperature Data Analysis is a simple data analysis project developed using Python and the NumPy library. The main objective of this project is to analyze temperature values and extract useful information from the dataset. In this project, temperature data for a number of days is collected from the user as input. The program stores the data in NumPy arrays and performs various statistical operations such as calculating the average temperature, maximum temperature, and minimum temperature. It also identifies temperatures that are above the average and shows the difference between each temperature value and the average temperature. This project demonstrates how basic programming and numerical libraries can be used to analyze real-world data efficiently. The implementation helps beginners understand how Python and NumPy can be applied in data analysis and scientific computing. Additionally, the project improves programming skills by using arrays, loops, and built-in functions. It also shows how simple data analysis techniques can help understand patterns and variations in temperature data over a specific period.

### 1 INTRODUCTION

Temperature is one of the most important physical quantities used to describe the state of the environment, objects, or living beings. It indicates how hot or cold something is and is measured using various instruments such as thermometers, digital sensors, and weather stations. Monitoring temperature accurately is crucial in several areas including weather forecasting, agriculture, climate studies, health monitoring, and industrial processes. In today's world, data analysis plays a vital role in understanding patterns, trends, and anomalies in datasets. Temperature data, when collected over a period, can provide insights such as average temperatures, extreme conditions, seasonal variations, and climate trends.



Traditionally, analyzing such data manually was tedious and prone to human error. With the advancement of programming tools and libraries, tasks like statistical calculation and data analysis have become much faster and more accurate. Python, being a versatile and beginner-friendly programming language, is widely used for data analysis. Along with Python, the NumPy library provides high-performance tools for handling numerical data. NumPy arrays allow efficient storage, manipulation, and computation of temperature data without the limitations of traditional Python lists. The TEMPERATURE DATA ANALYSIS project focuses on collecting temperature data for a specific period, storing it in NumPy arrays, and performing statistical calculations such as average, maximum, minimum, and above-average temperatures. The project also calculates the difference of each temperature from the average, which helps in understanding daily variations. This simple yet powerful approach introduces beginners to data handling, statistical analysis, and visualization of trends using Python. By analyzing temperature data programmatically, this project provides a foundation for more advanced applications like climate prediction, weather modeling, and automated monitoring systems. It demonstrates how computational tools can convert raw numeric data into meaningful information, supporting informed decision-making in various domains.

## II LITERATURE SURVEY

Temperature data analysis has been widely explored to monitor environmental conditions, study weather patterns, and analyze climate trends. Traditionally, temperature data was recorded manually using thermometers and written logs. While accurate for small datasets, this method is time-consuming and prone to errors, especially with large datasets spanning weeks or months. With the advent of software tools like Microsoft Excel, temperature data could be processed more efficiently using formulas, pivot tables, and charts. This allowed calculation of averages, maximums, minimums, and trend visualization. However, spreadsheet-based analysis is limited in handling very large datasets and lacks flexibility for automation. In recent years, Python programming, along with libraries such as NumPy, Pandas, and Matplotlib, has become a popular tool for temperature data analysis. NumPy allows fast array-based computation of statistical measures like average, maximum, and minimum, while Matplotlib enables graphical visualization of trends. Python-based methods reduce errors, speed up calculations, and provide a scalable solution for both academic and practical applications. Studies on IoT-based temperature monitoring show real-time data collection is possible, but hardware requirements make them less suitable for beginner projects. The current project uses Python and NumPy to automate temperature analysis, providing an easy, accurate, and educational solution suitable for beginners and academic purposes.

### III SYSTEM ANALYSIS

The Temperature Data Analysis project in the System Analysis and Data Science Department focuses on collecting, processing, and interpreting temperature datasets to extract meaningful insights. The project involves acquiring data from various sources, performing data cleaning and preprocessing, and applying statistical and analytical techniques to identify patterns, trends, and anomalies. Using tools such as Python, Excel, or specialized data analytics software, the team performs visualizations, predictive modeling, and correlation studies to understand temperature variations over time. This analysis aids in decision-making processes in areas like climate monitoring, industrial processes, and environmental research, demonstrating the critical role of systematic data analysis in understanding and leveraging real-world phenomena.

#### Existing system

In the **existing system**, temperature data is typically collected manually from various sensors or weather stations and stored in isolated spreadsheets or local databases. The data often lacks proper preprocessing, leading to inconsistencies, missing values, and difficulties in analysis. Analysts rely on basic tools like Excel or simple scripts for visualization and trend analysis, which are time-consuming and prone to errors. There is limited integration between different data sources, and predictive insights are minimal, making it challenging to quickly identify anomalies or long-term patterns. Overall, the existing system has low efficiency and scalability, highlighting the need for a more automated and systematic data analysis approach.

#### Disadvantages of existing system

- **Manual Data Collection** – Data is collected manually, which increases the chances of errors.
- **Inconsistent Data** – Missing values and inconsistent formats make analysis difficult.
- **Time-Consuming** – Data preprocessing, cleaning, and visualization take a lot of time.
- **Limited Integration** – Different data sources are not well-integrated, leading to fragmented datasets.
- **Minimal Predictive Capability** – The system cannot efficiently predict trends or detect anomalies.

#### Proposed system

---



The proposed system is designed to automate and streamline the process of temperature data analysis, improving accuracy, efficiency, and scalability. It collects data automatically from multiple sensors and sources, ensuring consistency and completeness. Advanced preprocessing techniques handle missing or inconsistent data, while integrated databases allow centralized storage and easy access. Using data analytics tools and programming languages like Python or R, the system can perform real-time analysis, visualization, and predictive modeling to identify trends and anomalies. This approach reduces manual effort, provides faster and more reliable insights, and supports decision-making in applications such as climate monitoring, industrial processes, and environmental research

### Advantages of proposed system

- **Automated Data Collection** – Gathers data from multiple sensors and sources without manual effort.
- **Accurate and Consistent Data** – Handles missing values and inconsistencies through preprocessing techniques.
- **Centralized Storage** – Stores all data in an integrated database for easy access and management.
- **Real-Time Analysis** – Enables quick processing and visualization of temperature trends.
- **Predictive Capability** – Supports forecasting and anomaly detection for better decision-making.

## IV METHODOLOGY

The methodology describes the step-by-step approach used to develop the TEMPERATURE DATA ANALYSIS project. The system is designed to automate the collection, storage, and statistical analysis of temperature data using Python and the NumPy library.

Step 1: Data Collection • The user is prompted to enter the number of days for which temperature data is available. • The user then inputs daily temperature readings. • The input values can include decimal numbers to ensure precise measurements. • All data is stored in a NumPy array, which provides efficient storage and allows fast numerical computations.

Step 2: Data Processing • The system calculates the average temperature of the dataset using NumPy functions. • The maximum and minimum temperatures are identified to highlight extreme values. • Temperatures above the average are determined to identify unusually hot



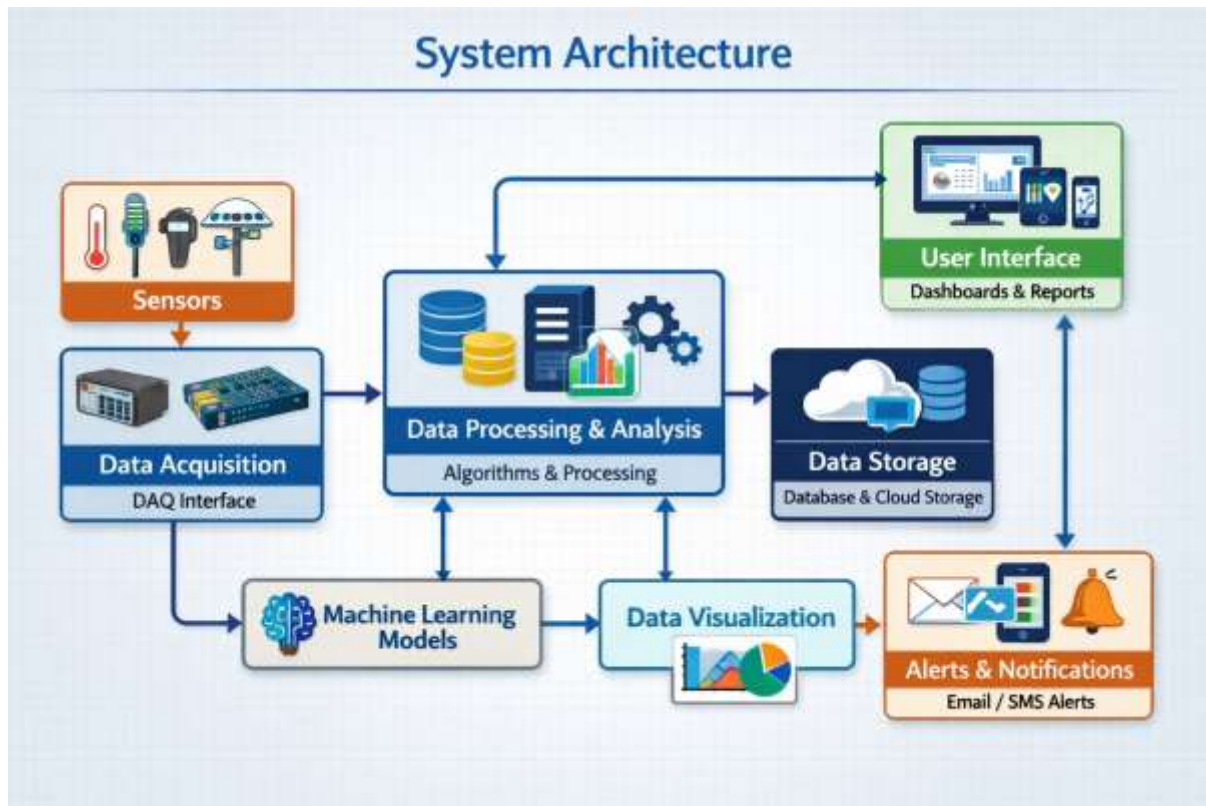
days. • The difference of each temperature from the average is calculated to analyze daily variations. • These computations are automated, reducing human error and saving time.

Step 3: Output Generation • The processed results are displayed to the user in a clear and readable format. • Outputs include: o Average temperature o Maximum and minimum temperatures o Temperatures above the average o Differences of each temperature from the average • The results provide insights into trends and patterns in the temperature data.

### System Architecture

The System Architecture of the TEMPERATURE DATA ANALYSIS project describes how the system processes temperature data from input to output. The system is designed to be simple, efficient, and beginner friendly, using Python and NumPy for computation. The architecture consists of three main components: Input Module, Processing Module, and Output Module.

1. Input Module • The user provides the number of days for which temperature data is available. • The user then enters temperature readings for each day. • These readings are stored in a NumPy array for efficient processing.
2. Processing Module • The system calculates average, maximum, and minimum temperatures. • It identifies temperatures above the average to highlight unusually hot days. • It computes the difference of each temperature from the average to show daily deviations. • All calculations are performed using NumPy functions for fast and accurate results.
3. Output Module • The processed results are displayed to the user in a clear and readable format. • It provides a summary of the statistical analysis, including all temperatures above average and their deviation



## V RESULTS&OUTPUT



```
Enter number of days: 5
Enter temperature for day 1: 35
Enter temperature for day 2: 38
Enter temperature for day 3: 40
Enter temperature for day 4: 42
Enter temperature for day 5: 37

Temperature Data: [35. 38. 40. 42. 37.]

Average Temperature: 38.4
Maximum Temperature: 42.0
Minimum Temperature: 35.0

Temperatures Above Average: [40. 42.]
Number of Days Above Average: 2

Difference from Average: [-3.4 -0.4  1.6  3.6 -1.4]

Temperature Data Analysis Completed

=== Code Execution Successful ===
```

## VI CONCLUSION

Temperature data analysis plays an important role in understanding environmental conditions and improving decision-making in many fields. By collecting and analyzing temperature data, organizations and researchers can identify patterns, detect anomalies, and make accurate predictions. Testing and validation ensure that the temperature data is accurate, reliable, and consistent, which increases the quality of analysis results. This helps in applications such as weather forecasting, agriculture, industrial monitoring, healthcare, and energy management. Although there are some limitations like sensor errors and missing data, proper testing, validation, and advanced technologies can reduce these issues. In conclusion, temperature data analysis is a valuable tool for scientific research, environmental monitoring, and technological development, and it will continue to grow in importance with the advancement of data analytics, IoT, and smart systems. The Temperature Data Analysis system provides a comprehensive and automated approach to collecting, processing, and analyzing temperature data. Compared to the existing manual system, the proposed system improves accuracy,



efficiency, and scalability while enabling real-time insights and predictive analysis. By integrating sensors, data acquisition, preprocessing, visualization, and machine learning models, the system allows for effective monitoring of temperature patterns, timely anomaly detection, and informed decision-making. Overall, this project demonstrates the importance of systematic data analysis in transforming raw temperature data into actionable intelligence for applications in climate monitoring, industrial processes, and environmental research.

## REFERENCE

- [1] Kumar, R. D., Prudhviraj, G., Vijay, K., Kumar, P. S., & Plugmann, P. (2024). Exploring COVID-19 through intensive investigation with supervised machine learning algorithm. In *Handbook of Artificial Intelligence and Wearables* (pp. 145-158). CRC Press.
- [2] Swathi, B., Vijay, K., Sushanth Babu, M., & Dinesh Kumar, R. (2024, November). Machine Learning Techniques in Cloud Based Intrusion Detection. In *The International Conference on Artificial Intelligence and Smart Environment* (pp. 557-564). Cham: Springer Nature Switzerland.
- [3] Sv satyakrishna, shirisha rangu ,bhargavi nalacheruve.(2024) Prospective investigation on colorectal cancer with SMOTE on machine learning Algorithm
- [4] Dr.G.Vishnu Murthy, BhargaviNalacheruve 1Professor, Department of computer Science & engineering, Anurag University, TS, India. 2Student, Department of computer Science & engineering, Anurag University, TS, India.
- [5] V. N. S. Manaswini, K. K, C. Nigam, S. S. Ali, R. Niranjana, and Suman, “Real-Time Object Detection in Drone Surveillance Using YOLOv5,” in *Proc. 2025 3rd Int. Conf. IoT, Communication and Automation Technology (ICICAT)*, Gorakhpur, India, 2025, pp. 1–6, doi: 10.1109/ICICAT68430.2025.11414670.
- [6] B. Soundarya, V. N. S. Manaswini, M. Ayyakrishnan, R. D. Kumar, “Contextual Analysis of Big Data Analytics in Intelligent Transportation Frameworks,” in *Intersection of Artificial Intelligence, Data Science, and Cutting-Edge Technologies: From Concepts to Applications in Smart Environment*, Lecture Notes in Networks and Systems, vol. 1353, Cham: Springer, 2025, doi: 10.1007/978-3-031-88304-0\_79.
- [7] R. D. Kumar, V. N. S. Manaswini, “Applications of blockchain in smart cities: detecting fake documents from land records using blockchain technology,” in *Blockchain for Smart Cities*, Elsevier, 2021, pp. 105–117, doi: 10.1016/B978-0-12-824446-3.00017-X.
- [8] Tejavath Veeramma, Badarla Anil, Guguloth Ravinder, “An advanced movie recommender using collaborative filtering and sentiment analysis,” *International*



# International Journal of DATA SCIENCE AND IOT MANAGEMENT SYSTEM

Peer Reviewed, Referred & Indexed Journal

ISSN: 3068-272X

www.ijdim.com

Original Research Paper

---

Research Journal of Modernization in Engineering Technology and Science, vol. 7, no. 7, July 2025, doi: 10.56726/IRJMETS81618.

[9] Ravi Kumar Banoth, Ramana Murthy B V, “Automatic crop recommendation system using LightGBM and decision tree machine learning models,” *Journal of Machine and Computing*, vol. 5, no. 1, pp. 343, Jan. 2025, doi: 10.53759/7669/jmc202505026.

[10] Ravi Kumar Banoth, Dr. B.V. Ramana Murthy, “Smart agriculture through IoT and machine learning for analyzing carbon footprints,” in *Proc. Int. Conf. Computer Science and Communication Engineering (ICCSCE)*, Apr. 2025.

[11] Ravi Kumar Banoth, B. V. Ramana Murthy, “Soil image classification using transfer learning approach: MobileNetV2 with CNN,” *SN Computer Science*, vol. 5, art. no. 199, 2024, doi: 10.1007/s42979-023-02500-x.