UNIVERSITY of BRADFORD

Faculty of Engineering and Informatics
Department of Computer Science

Predicting Variation in Daily Martian Atmospheric Temperature Using Machine Learning Techniques.

A Big Data Project

Supervisor: Prof. Rami Qahwaji
Student: Gabriel Tosin Ayodele
UB Number: 21020767

A thesis submitted in part fulfillment of the degree of

MSc in Big Data Science and Technology 2023.

Author’s Statement of Originality

I certify that I am the sole author of all the material in this dissertation and that it is my original, independent work. I certify that all references used in the research were properly cited, and all quotations were given their proper due. This contribution in its entirety has never been submitted to another academic institution or with the goal of earning another degree or credential.

I am aware of how performing research may violate ethical standards, and I have taken all reasonable measures to assure the accuracy of the findings. I have followed the rigorous academic standards established by the concerned department and do not violate anyone's or anything's ownership rights.

I promise to preserve the values of academic honesty and integrity by submitting this dissertation, and I vouch for the work's accuracy and originality.

Student Name: Gabriel Tosin Ayodele

UB Number: 21020767

Student Signature

Dedication

I dedicate this project to my mother, Mrs Florence Tewogbola Ayodele, for her love, sacrifices and support throughout my life and education. To my sister Mrs. Titilayo A. Kuteyi, your selflessness, love, encouragement, and sacrifices have been my greatest strength.

Acknowledgement

I am grateful to God for His grace and blessings throughout this journey.

I would like to express my sincere gratitude to my supervisor, Professor Rami Qahwaji, for his superb leadership, inspiration, and constant support. I'm very thankful to him for being my mentor and giving me important insights, advice, and constructive criticism throughout the research.

I also want to convey my profound gratitude to the University of Bradford's academic and non-academic personnel, especially the College of Engineering and Informatics, for fostering an environment that supports academic performance. The educational experience would not have been complete without the dedication and help of the whole faculty, who went above and beyond to make sure we got the best education possible.

My appreciation also extends to my fellow students and friends, particularly Emma Emina, Ene Ogbe, Bankole Joshua, Ezekiel Sunday, Femi Omitoyin, and Ehionyen Ekoma, for the constant support and encouragement.

Finally, this journey would not have been possible without the support of my family and friends. Their love and support have been immeasurable throughout this journey.

Abstract

According to research and explorations conducted on mars. It was believed that the planet once supported life. This resulted from some evidence indicating the presence of water and a thick warm atmosphere on mars in the past. Due to this, man became curious to explore the planet as the red planet is believed to be similar to earth. Many technological tools have evolved with advancements in science and technology, making the entire exploration mission easy. Machine learning standout to be an essential technical tool today that is in use for mars exploration. Although the field is new, little work has been done in exploring mars' weather patterns for possible habitation of life. This work relied on the strength of machine learning techniques to analyse the mars data obtained from the EMM EMIRS instrument database. In this work, various techniques such as exploration, preprocessing, model development, and evaluations were performed on the dataset to predict the variation in daily atmospheric temperatures of mars present within the dataset. Seven machine learning models were developed in this work: Random Forest, Linear regression, SVM, ANN, CNN, Lasso Regression and Decision tree. The developed models were subjected to hyperparameters fine-tuning their accuracies. Finally, each model was evaluated using RMSE, MAE, and R2 value evaluation techniques. Following the evaluation, random forest outperformed all other six models used in this work with scores as follows: RMSE=1.5525, MSE=2.4105, MAE=1.1939 and R2=0.8283. Finally, the scores obtained were visualised.

Keywords: *Mars, space exploration, Models, Python, NASA, EMIRS, EMM, evaluation, pre-processing*

Contents

List of Figures

List of Tables

List of Abbreviation

CNN	convolutional neural networks
GRU	Gated recurrent units
LSTM	Long short-term memory networks
EMM	Emirate Mars mission
NASA	National Aeronautics and Space Administration
MSL	Mars science Laboratory
SCOTI	Scientific Captioning of Terrain Images
EMIRS	Emirate mars infrared Spectrometer
MSE	Mean squared error
MAE	Mean absolute error
RMSE	Root mean squared error
DTM	Digital Terrain Model
ANN	Artificial Neural Network
SVM	Support Vector Machine
TES	Thermal Emission Spectrometer
OTES	OSIRIS-REx Thermal Emission Spectrometer
FITS	Flexible image transport system

CHAPTER 1

Introduction

Mars is believed to be a planet that has supported life (Chang K., 2013) due to its similarity with Earth. Out of curiosity, humankind, over the years, has deployed different space exploration technologies to explore other planets for the possibility of life's existence. In quest of this exploration, planet mars are believed to be a planet with a likelihood for life. Planet mars can be viewed directly from Earth (Charles et al. *al. 2022*) due to its unique red colour, thus making it the Red Planet. The reddish colour of Mars is due to the presence of iron oxides on its surface.

Planet mar and planet earth has similarities in climate hence making it considered as a possible planet to be habitable. Although the size of earth is 11% bigger than mars and 50% behind closer to the sun when compared to mars. The climatic conditions of both planets have much similarities for instances the weather patterns, presence of polar caps and changes in season are both similar. Earth-based instruments have studied planet Mars since the 17th century, but close close-translation from within the Earth using space exploration techniques exploration began in the mid-1960s. Over the decades, orbital spacecraft and flyby have make data for mars exploration available using technologies available in the present age, while rovers and landers have been used to record the condition of mars atmosphere. On August 6, 2012, NASA's MSL Curiosity rover landed on the red planet successfully (National weather science, 2020). Its mission was to study the weather patterns of the red planet as this is essential for a successful human exploration of mars and also provide knowledge on the evolution of the planet mars. Curiosity has been active on mars for over 3640 earth days and has successfully documented data for different variations of mars seasons (Joaquin Rojas, 2022).

Despite success in landing rovers on mars and obtaining information about mars' weather conditions, the dynamic nature of its weather patterns remains an area that needs to be explored for better understanding. This research is aimed at developing a machine-learning model for predicting the daily variation in atmospheric temperatures on Mars. The model will be trained on data collected from the EMM EMIRS instrument, and its accuracy will be evaluated using a suitable evaluation metric. The results will provide valuable insight on Martian climate and contribute to existing knowledge about weather patterns on the red planet.

1.1 Problem Statement

The study of Mars and its atmospheric conditions has been a topic of great interest to scientists and researchers for many years. Mars is unique in that it has diverse weather patterns, which are crucial for future missions to the planet and for predicting potential impacts on human and robotic missions. Mars, like other planets, experiences unique weather patterns, including daily temperature variation. This variation, which occurs over a day, is an essential aspect of the Martian climate and can provide valuable insights into the forces driving this planet's weather. To get these insights, machine learning techniques have provided means to identify the variations. This can be achieved through the development of models.

Despite the extensive research that has been conducted on the Martian weather system, there is a need for more accurate and reliable models for predicting this variation in daily atmospheric temperature. The current models used to predict Martian weather patterns are based on empirical relationships and rely on limited data (Ishaani P. and Puri V,2021; Ali M,2021). As a result, these models become limited by their inability to account for the complex interactions between different atmospheric variables and the planet's surface.

1.2 Research Questions

- Is there a variation in the atmospheric temperature of mars within a day?
- Does a relationship exist among the various temperatures present in mars?
- Can a robust machine learning model be developed to predict the daily variation in the atmospheric temperature of mars?

1.3 Aims and Objectives

This research work will be aimed at solving the modelling issue by exploring the EMM (Emirates mars mission) EMIRS (Emirates Mars Infrared Spectrometer) instrument dataset to study the variation in atmospheric temperature and different temperatures present in mars weather data recorded by EMIRS instrument of EMM using various machine learning techniques. These techniques and algorithms will provide means for exploration of the dataset, performing pre-processing of the dataset and developing machine learning models that can be used to predict daily variations in Martian atmospheric temperature. Also, this work aims to contribute significantly to understanding mars' weather patterns as it impacts the planet. Based on such aim, the following are the study's objectives:

1. Identifying if there is a variation in the daily atmospheric temperature of planet mars.
2. Detecting if a correlation exists among various temperatures in planet mars
3. Develop a machine learning model to predict of daily mars' atmospheric temperature variations.
4. Contributing to the body of research on mars weather pattern exploration.

1.4 Justification of Research

Upon completion, this research work will improve existing works in the field of Machine learning applications on Mars weather patterns exploration. Prior to this work, few research papers have been written on the Application of machine learning techniques and algorithms on mars weather exploration. This research work will add to existing research papers on this topic of interest. It will also provide new dimensions and machine-learning models that can be used to explore temperature pattern patterns using the EMM EMIRS dataset. In addition, it will provide a basis for future research works that researchers can look upon and make additions to improve existing Machine models used for Mars weather patterns exploration. Space exploration heavily relies on Machine learning and Artificial Intelligence today as they provide several solutions and models (both developed and pre-trained models) that are useful for mars weather exploration.

1.5 Scope of Research

The proposed research will focus on exploring the EMM EMIRS instrument dataset to get insights into the temperature patterns on planet mars. It will also focus on developing Machine Learning algorithms to improve existing models used in previous research papers to predict Mars weather patterns. The models (Algorithms) to be developed will be used to predict the variation readings for daily atmospheric temperature presented within the dataset, which can be used subsequently for predictions on new data. This work will be based on textual datasets extracted from fit files presented by the EMM data repository, which provided data from February 2021 to November 2021.

1.6 Motivation

Mars, the fourth planet from the sun, has always been a subject of intense scientific and public interest due to its potential to support life. The current understanding of Mars's weather patterns is crucial in understanding the planet's climate and geology. This is why the prediction of Mars weather patterns has become a crucial area of research. Despite numerous missions to Mars by various space agencies, much of the planet's weather patterns still need to be better understood. The unpredictable nature of Mars weather and its potential impact on future manned missions to the planet predicts Mars weather patterns as a top priority.

1.7 Organization of the Project

This research paper is segmented into five sections: Chapter one presents an introduction to the research topic under review, chapter two presents a review of related works and machine learning techniques in the field of mars weather exploration, chapter three presents the methodologies deployed In this research paper, chapter four presents analysis and discussion of results obtained following the experiments conducted on mars dataset, chapter Five provides a conclusion and future work in the field of mars weather exploration.

CHAPTER 2

Literature Review

2.1 Related works

The fourth planet from the sun, Mars, has been a subject of ongoing scientific exploration and study (NASA, 2021). The increasing number of missions to the planet has increased the interest in understanding its weather patterns, which can significantly impact the habitability of Mars and the success of future missions. The use of machine learning techniques in studying weather patterns on planets, including Mars, has become prevalent in recent years (Ali M., 2019). These techniques can provide new insights into the Martian atmosphere's dynamics and improve weather pattern predictions (Michael D, 2022).

Mars has a complex atmosphere composed of several gases, including carbon dioxide, nitrogen, and argon, and a thin layer of water vapour (NASA, 2021). In addition, the planet experiences various weather phenomena, including dust storms, devils, and temperature fluctuations (Michael D,2022). These weather patterns can significantly impact the success of missions to Mars, and it is essential to have a better understanding of them.

Machine learning techniques have been used in several applications in the field of atmospheric science, including the prediction of weather patterns (Ishaani & Puri, 2021). These techniques can provide new insights into the dynamics of the Martian atmosphere and improve predictions of weather patterns, which can have important implications for future missions to Mars (Fendall et. al.,2014).

Several research have been done to explore the weather patterns on mars using machine learning techniques. These patterns include variation in dust storms, Ultraviolet radiation, temperature fluctuations etc. (Michael D. et. al., 2022). However, there needs to be more comprehensive research on the prediction of Mars weather patterns using machine learning, and more work is needed to understand the potential of these techniques in this field. This review aims at providing a comprehensive overview of the current state of research on the prediction of mars weather patterns using machine learning techniques. In addition, the review will cover the methodology and results of previous studies and provide a foundation for future research in this field.

In 2017, Daniele Bellutta performed two experiments on the tau data obtained from the jet propulsion Laboratory dataset from Mars Exploration Rovers mission for his project. First, he deployed Levenberg-Marquardt backpropagation and then cross-validating 15% of the Artificial Neural Network, which he deployed in his work. Several trials were performed, and the average Mean squared error for these trials was taken for the two experiments.
This work aimed to develop a model for predicting the opacity of the Martian atmosphere.

Kass D. M. et. al. (2020) researched the wave activities of gravity in the lower atmosphere of Mars based on the Mars Climate Sounder observations. This was aimed at exploring the effect of gravity waves on Mar’s atmosphere. The study identified that intense and moderate activities of gravity can be observed all over tropical volcanoe and at middle to high latitudes for both hemispheres during fall and winter. Furthermore, it is also noticed at night in parts of the southern tropics during winter and fall.

Ishaani P. and Puri V. (2021) performed data analysis on martian data, which they obtained from Kaggle. Their work aimed to explore mars weather data using artificial intelligence skills (machine learning) and lay a foundation for future data analysis work on mars rover datasets. In their work, five machine-learning models were deployed on the used dataset. These models were convolutional neural networks (CNN), Gated recurrent units (GRU), Long short-term memory networks (LSTM), Stacked Long short-term memory networks (SLSTM) and CNN-LSTM. Evaluation techniques such as MSE, RMSE, MAE and R2 Value were used to evaluate the five models deployed. After applying this evaluation model, LSTM outperformed the other four models as it provided the highest R2 value and most negligible MSE value hence making it to be identified as the most robust model to make predictions on mars.

Alejandro de C. G. et al. (2019) studied Mars's Meteorological variables and Applied machine learning techniques. This research aimed to use a probabilistic approach to predict Martian weather patterns. This work used ANN to predict sol pressure and air temperature hourly. The model development process used variables such as sol, hour, air temperature, pressure, and orbit (Ls). The data were divided into 75% to 25% for training and testing. The data was used to train the network as the latter, which were unknown to the network, were used to validate the model's accuracy.

C. Charalambous et al. (2021) researched and identified observations of signals partitioning into seismic and environmental contributions owing to the Martian atmosphere. A temporal cross-frequency coupling was done across multiple bands, which produced noise as a result of fluctuations in wind and atmospheric pressure. The same was achieved using modulation and quantification of seismic motion, wind and pressure. The signal ratio to the amount of noise produced can be quantified concerning environmental independence.

Olsen K. F. et. al. (2021) performed studies on the geophysical and biological activities of Mars's atmosphere. They aim to identify possible gasses present in Mars's atmosphere. This was achieved through the detection of halogen gas which could have been released as a result of volcanic and gas-solid reactions. A global dust storm, which took place in the year 2018, led to an increase and release in halogen gas, which was later curtailed. It resulted in wide distribution of halogen gas, which was said to be 20 times more compared to the previously released.

Christopher Lee (2019) proposed an automated system for detecting craters and cataloguing them on mars. The model developed in this work was a digital terrain model (DTM) of mars. The DTM images developed use convolutional neural networks and UNET Architecture, and the location and size of craters features were found by matching circle algorithms. The crater detection algorithm was compared to existing crater datasets, and its performance was evaluated.

E Gramigna. In 2020, a calibration technique was developed to analyse the weather condition of mars and Venus. The aim of the work was to identify differences between Venus and Mars's atmosphere. This calibration technique was deployed so as to avoid biases when getting data for the atmosphere from both planets. From the results obtained on conducting the analysis, the atmospheric condition of Venus is more hostile than that of Mars as mars produced more thin and friendly weather compared to that of Venus.

P. N. Timothy et al. (2022) pinpointed the usefulness of machine learning on mars science. This research showed how machine learning knowledge and techniques have helped in the field of Martian science. They encouraged the movement of less human data imputation will help significantly in the field of Mars science and the application of weather patterns. More participation and involvement of machine learning skills, techniques and knowledge will significantly impact space weather exploration. As a result, more effort and money should be channeled towards developing more data. This will enhance the effectiveness of Machine learning on Martian science as ML depends largely on datasets.

Overall, these studies demonstrate the potential of machine learning for predicting mars' weather patterns. However, much work remains to be done in this area, and there is a need for further research to explore the various factors that affect the mars weather and to develop more sophisticated models for making accurate predictions.

2.2 Weather patterns on mars

Weather patterns refer to the recurring and predictable changes in atmospheric conditions that occur in a specific region over time. These changes can include temperature, pressure, wind, humidity, and precipitation and are driven by various meteorological processes such as air masses, fronts, and cyclones. The complex interplay of various meteorological factors, including solar radiation, atmospheric dynamics, topography, and ocean currents, determines a weather pattern. These factors work together to create recurring patterns of weather in specific regions, which can persist for periods ranging from a few days to several weeks or months.

Mars experiences a range of weather patterns, including dust storms, temperature variations, and wind patterns. These patterns are influenced by the planet's thin atmosphere, axial tilt, and surface features, such as the Valles Marineris canyons and Olympus Mons, the tallest volcano in the solar system (Fendall, 2014). One of the most unusual weather patterns on Mars is the frequent daily temperature variations, particularly in the equatorial regions, where the temperature can reach up to 70 degrees Fahrenheit during the day and drop to well below -100 degrees Fahrenheit at night (Dimitra A., 2023) The poles of Mars can reach as low as -195 degrees Fahrenheit, while only warming up to around -140 degrees Fahrenheit during the summer months.

Another mars predominant weather pattern is the intense dust storms that can engulf the entire planet. They occur as a result of the sun heating the surface of mars resulting to raise in surface temperature which causes hot air to rise and carry tiny particles of dust which varies in size and duration (Kuthunur S.,2023)

Mars also experiences a variety of wind patterns. For example, the planet's topography can influence wind patterns near the surface, with winds accelerating and changing direction as they move over mountains and valleys (Arizona state university,2022). At higher altitudes, the winds can be influenced by the planet's rotation, creating a pattern known as the Martian super-rotation.

2.3 Artificial Intelligence and Machine learning in Martian Science (Weather patterns analysis)

Over the years, a lot of interest and research have grown to explore and understand weather conditions and patterns within mars due to the possible habitation of this planet; hence, many technologies are put in place to explore this planet for the possible possession of life possibly. Machine learning, an aspect of Artificial Intelligence that uses collected data subject to training and then makes predictions based on the collected values, has paved the way for understanding weather patterns on mars. Machine learning helps systems reason and behave in similar fashions as humans. Machine learning works by using various algorithms, training them based on data provided to it and making predictions based on those data. The ability of Machine learning Models (Algorithms) to capture, analyze and describe complex data Patterns has made this field an essential field in many industrial and scientific sectors today (Issam El Naqa et al.,2015). These techniques have provided means for researchers to perform the study of space Missions by humans remotely here on earth without going to the planet mars themselves (Which is a harsh planet for the existence of life). Machine learning techniques have been successfully applied to spacecraft and to analyze space-collected data (Ali M., 2021). According to Ali M. (2021), the application of machine learning skills to mars exploration has provided a means to overcome the communication gap which exists when rovers are used on planet mars, as it will take a longer time to transmit data from mars to earth.

However, due to the existence of machine learning techniques, this communication gap is bridged as it provides data within a short time. It also provides researchers with visualization and images of weather patterns on mars from here on earth within a short while. Machine learning techniques have helped researchers explore planet mars better and have sufficient information concerning the planet, and put in place any measure required to enhance possible human missions on planet mars. Ishaani P. and Puri V. (2021) performed data analysis and machine learning techniques on the mars weather dataset provided by REMS, which was obtained from Kaggle by deploying various machine learning techniques on the martian dataset obtained. These models were convolutional neural networks (CNN), Gated recurrent units (GRU), SLTM, Stacked Long short term memory networks (LSTM) and CNN-LSTM. Evaluation techniques such as MSE, RMSE, MAE and R2 Value were used to evaluate the five models deployed. After applying this evaluation model, LSTM outperformed the other four models as it provided the highest R2 value of 0.8640 and the most negligible MSE value of 0.0294 hence making it the most robust model to make predictions on the mars weather pattern.

Kristin R. et al., 2021, used an unsupervised machine learning algorithm built by k-means clustering called the NMF model to classify data obtained by the Curiosity's Chemistry and Camera (ChemCam) instrument of NASA from the first 2,756 sols. The algorithm arranged the data into 6 different groups of chemical compositions, which include low and high SiO_2, felsic,low and high Ca_2O, and high total FeO_2. The results obtained showed a transition between the different regions along Curiosity's traverse since leaving the landing site. The results were obtained using only LIBS data, and no training data were required. To perform further classification using random forest model, the six groups were used for training the model. First, the random forest model was developed on a dataset split in the ratio of 80:20. The NMF scores and the corresponding cluster of 11,772 spectra were used for model training after which the NMF scores of 2,943 spectra were used for testing the model. Experiment was repeated 50 times by changing the training and test data randomly but keeping their sizes constant. During evaluation, the lowest score obtained from the 50 experiments was 0.998. 14 of the experiments produced an accuracy value of 1which allows the model to be used for predictions on new data unknown to the models.

Alejandro de C. G. et al. (2019) developed a machine learning algorithm using an Artificial neural network (ANN) to explore weather patterns from the mars dataset available. This created ANN model was used to predict planet mars' hourly pressure and temperature. The model development process used variables such as hour, sol, pressure, air temperature and orbit (Ls). The data were divided into 75% to 25% for the purpose of training the model as well as testing it. First, the data was used to train the network after which it the test set was used to validate the model's accuracy. The model produced a correlation coefficient score of 0.9995. Upon performing second validation, the correlation coefficient score obtained was 0.99035.

Despite this success, machine learning model development remains a challenge in the field of Martian science (Joanna Welden, 2022). This results from fluctuations in atmospheric parameters; hence, new research is to be conducted over time, and models are needed to improve upon to make the knowledge required for mars exploration available. Therefore, this research paper will explore datasets extracted from the EMM EMIR spectrometer instrument to explore underlying patterns within the extracted dataset and develop machine learning models that can be relied upon in the future to make predictions on mars' temperature value. This will be achieved by deploying various machine-learning algorithms. This report will discuss the algorithms deployed in the next section below.

2.4 Artificial Intelligence Algorithms

The Following a review conducted on existing research work to explore weather patterns on mars presented in sections 2.1 and section 2.2, various machine learning techniques have been applied to develop models for making predictions on mars' atmospheric conditions. However, below are some models which will be deployed in this research paper for the development of machine learning model for atmospheric temperature prediction:

1. **Random Forest**: Random Forests are an ensemble(combination) of decision trees. It is a Supervised Learning algorithm used for classification and regression. This technique uses a subset of training samples as well as a subset of features to build multiple split trees. Numerous decision trees are built to fit each training set. The distribution of samples/features is typically implemented in a random mode. The input data is passed through multiple decision trees. It executes by constructing a different number of decision trees at training time and outputting the class, that is, the mode of the classes (for classification) or mean prediction (for regression) of the individual trees. Random forests mostly handle fitting issues arising from an imbalanced dataset.

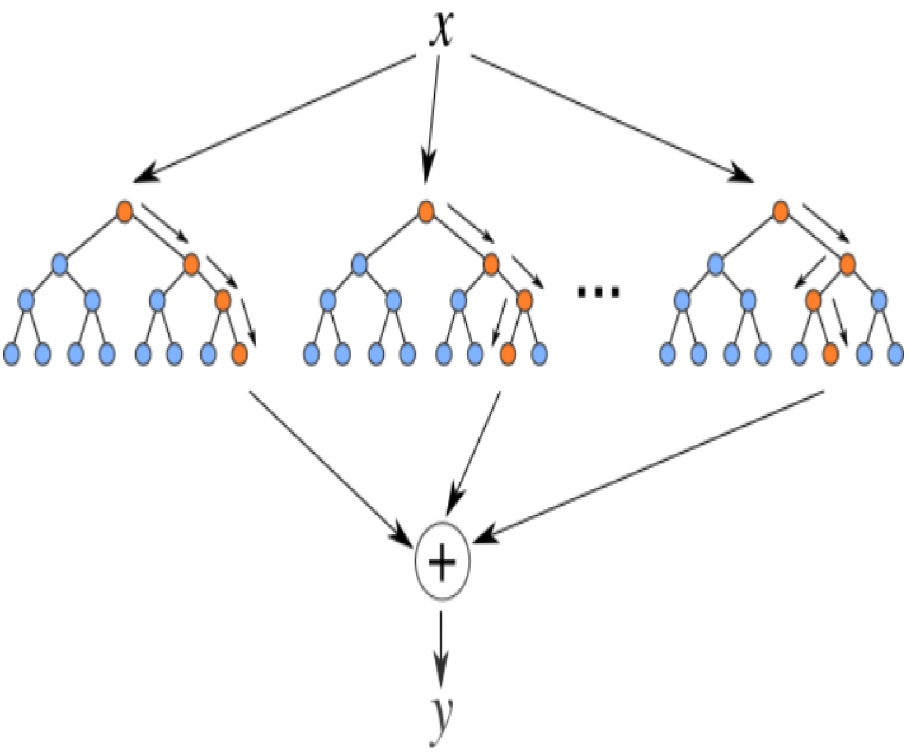


Figure 1: Random Forest (Image obtained from: link)

2. **Lasso Regression:** LASSO stands for Least Absolute Selection Shrinkage Operator. Shrinkage is defined as a constraint on attributes or parameters. The algorithm operates by finding and applying a constraint on the model attributes that cause regression coefficients for some variables to shrink toward a zero. lasso regression analysis is a shrinkage and variable selection method and helps to determine which predictors are most important.

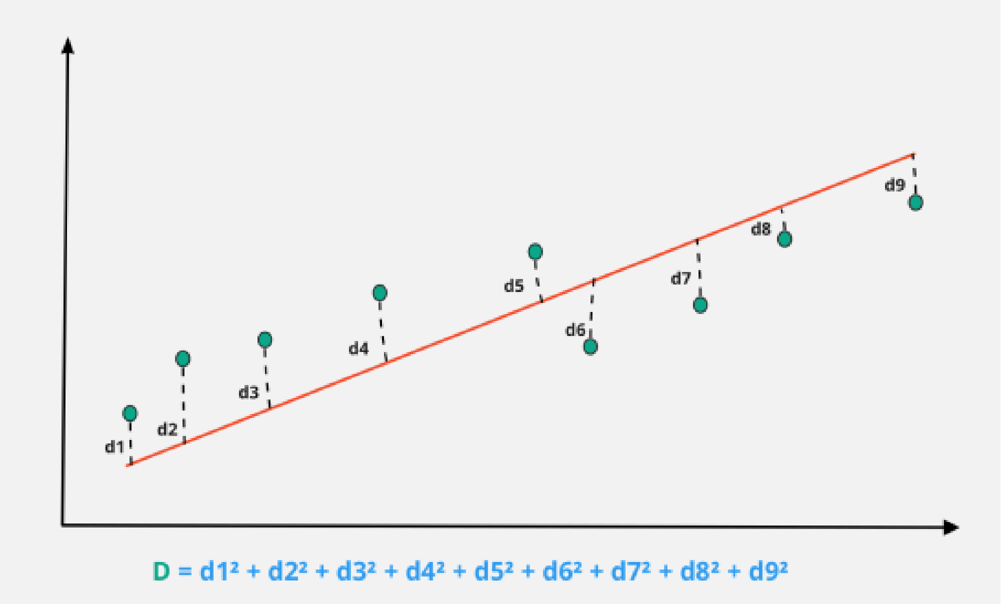


Fig. 2: LASSO Regression (Image obtained from: link)

3. **Decision Tree:** A decision tree is a machine-learning model representing data from nodes. The decision tree models can be applied to data which contains numerical features and categorical features. Decision trees are good at capturing non-linear interaction between the features and the target variable. Decision trees match human-level thinking, so it is intuitive to understand the data presented to them. The figure below shows how the decision tree works:

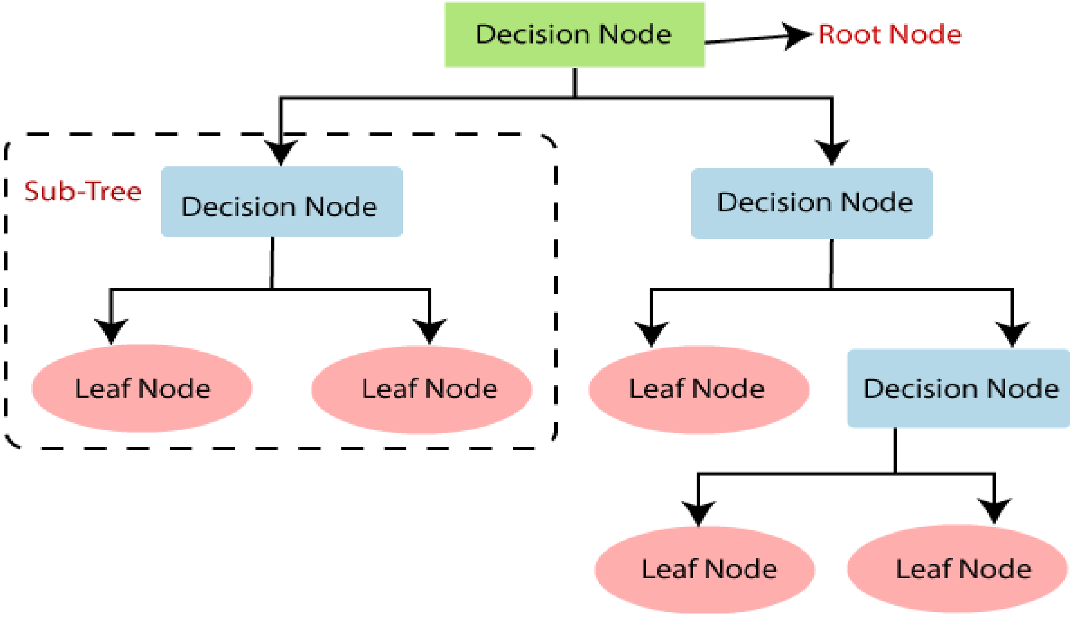


Figure. 3: Decision Tree (Image obtained from: link)

4. **Artificial Neural Network:** Artificial Neural Networks (ANN) are a special type of machine learning algorithms that are modelled after the human brain. ANN is able to learn from the data and provide responses in the form of predictions or classifications. ANNs are nonlinear statistical models which display a complex relationship between the inputs and outputs to discover a new pattern. An important advantage of ANN is the fact that it learns from the example data sets. ANN is also capable of taking sample data rather than the entire dataset to provide the output result.

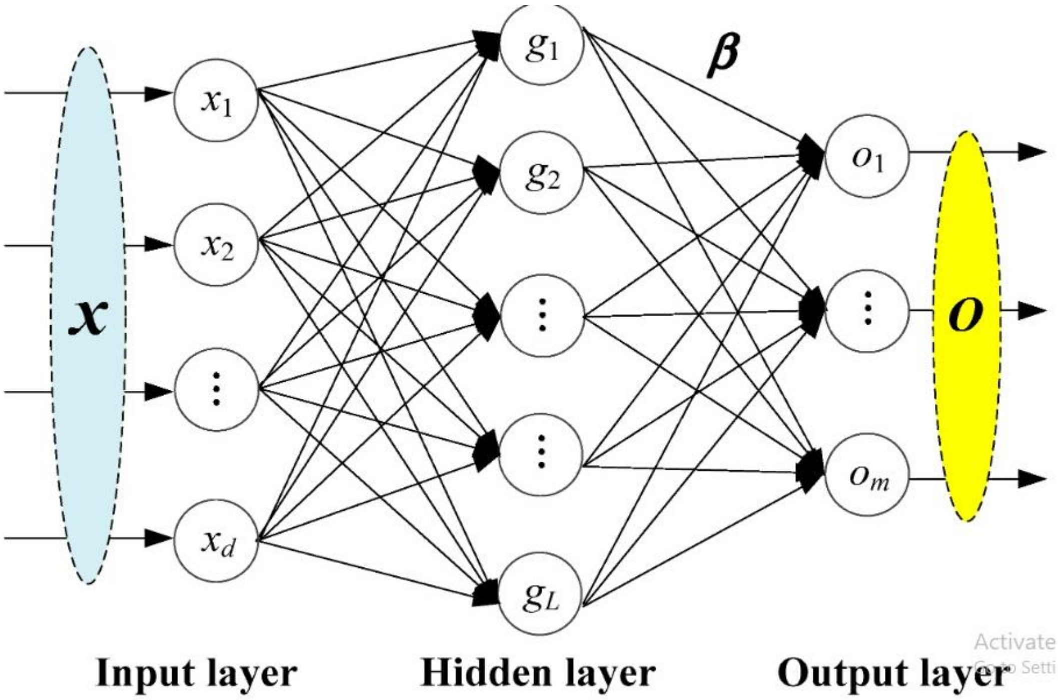


Fig. 4: Artificial Neural Network (Image obtained from: link)

5. **Support Vector Machine (SVM):** Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane.

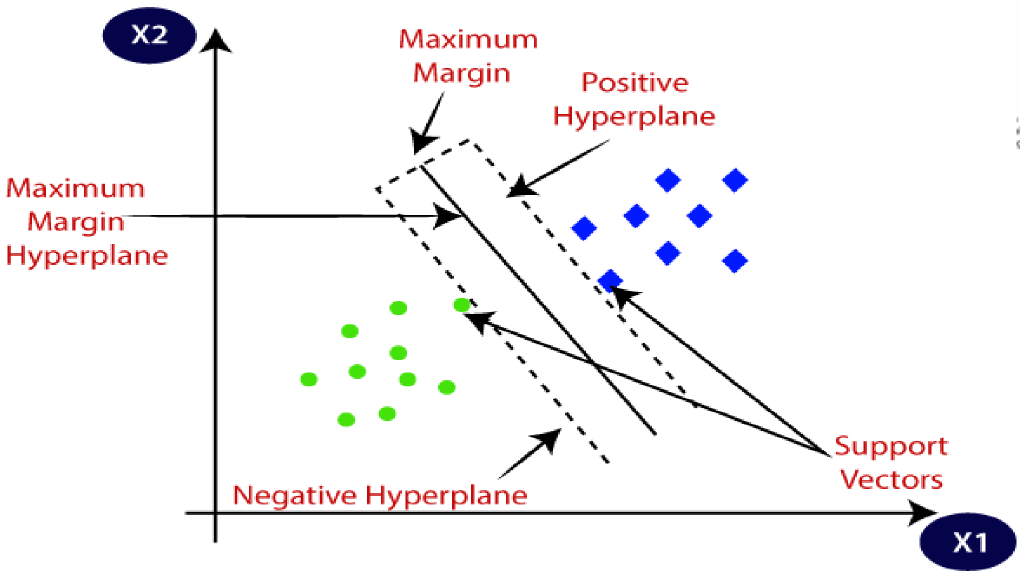
6. 

Fig. 5: Support Vector Machine (Image obtained from: link)

7. **Linear Regression:** Linear Regression is an ML algorithm used for supervised learning. Linear regression performs prediction of a feature called dependent variable based on the input data it receives (independent variables) (Rong, Shen and Zhang, 2018). The model works by finding a linear relationship between two sets of variables.

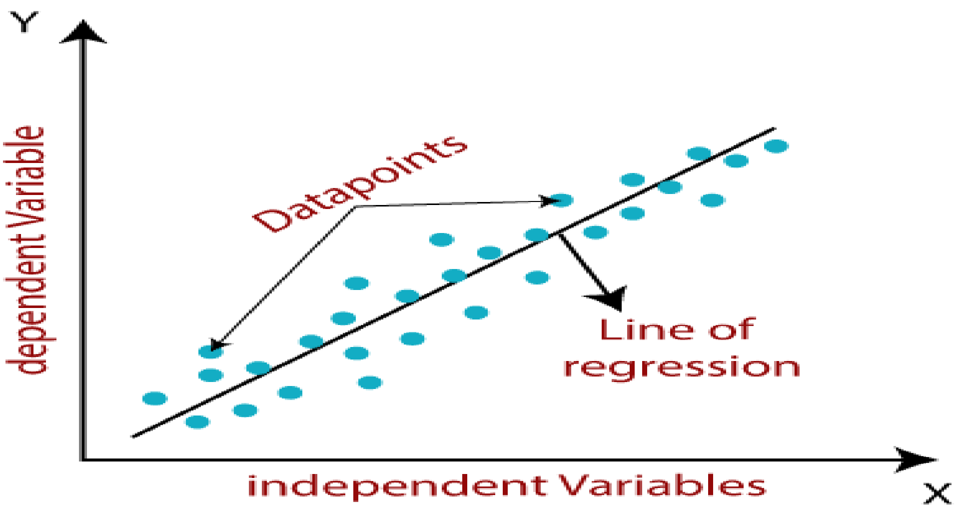


Fig. 6: Linear regression (Image obtained from:link)

8. **Convolutional Neural Network (CNN):** A Convolutional Neural Network (CNN) is a type of deep learning artificial neural network that

was originally designed for image classification tasks. However, CNNs have also been applied to non-image datasets. To use a CNN on non-image data, the data must first be transformed into a numerical representation that can be input into the network. A CNN's architecture is analogous to the connectivity pattern of the human brain (Rahul A.,2022). The figure below represents the architecture of a CNN algorithm on a sample data:

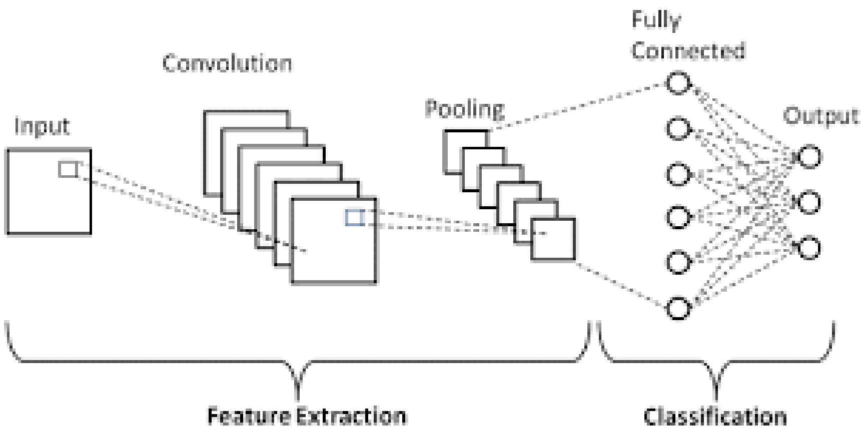


Fig. 7: Convolutional Neural Network (Image obtained from: link)

2.5 Overview of the Emirates Mars Infrared Spectrometer (EMIRS)

The Emirates Mars Infrared Spectrometer (EMIRS) is one of three scientific instruments abroad the Emirates Mars Mission (EMM) spacecraft. The Emirates Mars Infrared Spectrometer (EMIRS) instrument was launched to Mars on 19 July 2020 at 21:58:14 UTC from the Tanegashima in Japan. The instrument was designed to measure the dynamic nature in martian atmosphere. The EMIR'S instrument was designed to specifically look at the thermal state in the lower atmosphere, the geographical distribution of dust, water vapor and water ice, as well as the three-dimensional thermal structure of the Martian atmosphere and its variability on sub-seasonal timescales (EMM,2020).

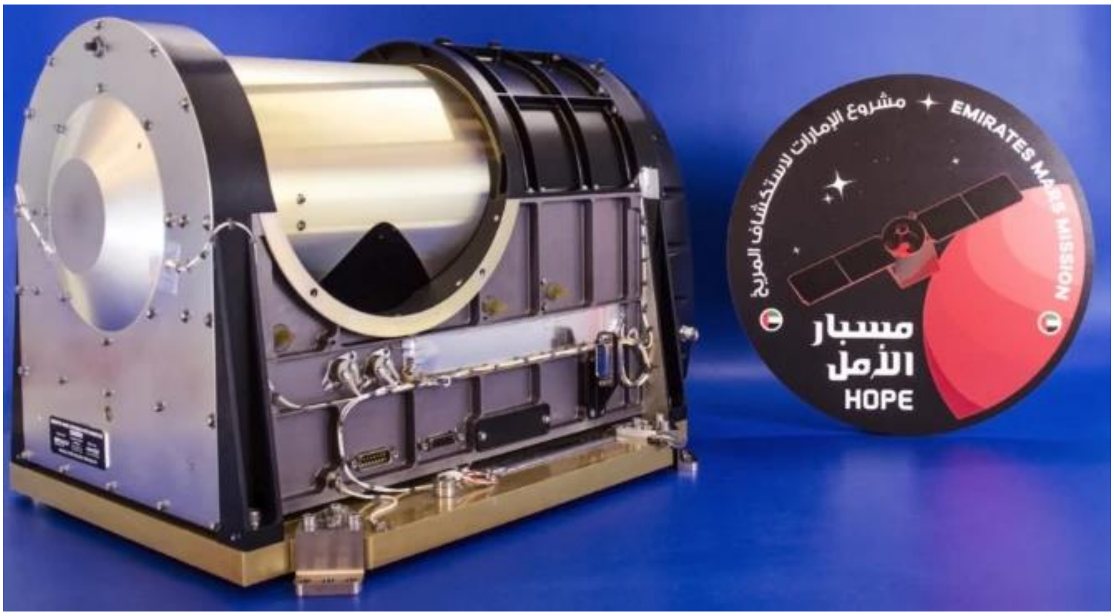


Fig. 8: The EMM EMIRS Instrument

The instrument was developed by Arizona State University (ASU) and Mohammed Bin Rashid Space Centre (MBRSC).It is managed by ASU's Mars Space Flight Facility, including the Thermal Emission Spectrometer (TES), Miniature Thermal Emission Spectrometer (Mini-TES), and the OSIRIS-REx Thermal Emission Spectrometer (OTES). EMIRS collects spectral data from 6-40+ μm at 5 cm-1 spectral sampling, which is enabled by a diamond beamsplitter and digital servo interferometer control electronics. The EMIRS instrument has a rotating mirror that will allow the instrument to do scans of Mars. EMIRS contributes heavily to the scientific objectives of EMM and will provide a new view of the Martian lower and middle atmosphere, capturing the sub-seasonal and diurnal evolution of key atmospheric constituents over the entire globe during the EMM two-year primary mission (Edwards C et. al., 2021).

CHAPTER 3

Methodology

3.1 Dataset used

The dataset used in this report was obtained from EMM official website. The used dataset was downloaded from the EMM EMIRS instrument available at: https://sdc.emiratesmarsmission.ae/data/science?instrument_id=emr&latest=latest&sort_by=Date%2Ftime&sort_order=desc&search_now=true.

The downloaded dataset was in Flexible Image Transport System (fits) formats hence the files were extracted. Fits files are digital file formats mostly used in astronomy to store data in array-like or table like formats. To perform extraction of the files, a python script was written to perform extraction of the fit files and convert the extracted files to csv. The written script is shown in the figures below:

```
from astropy.table import Table
from astropy.io import fits
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
def fit_cv(path):
    hdul=fits.open(path)
    print('######################fits tile imformation################')
    print(fits.info(path))
    print()
    print('=======fits file headers descriptions==========')
    print(hdul[1].header)
    print()
    events = Table.read(hdul, hdu=1)
    print('==============COLUMN NAMES CONTAINED IN FIT TABLE WHHICH HOLDS ACTUAL DATA=========')
    print(events.columns)
    print()
    s={}
    for x in events.columns:
        s[x]=events[x].data
    def converter(column_name):
        s=list(events[column_name][0:])
        for x in range (len(s)):
            s[x]=s[x].mean()
        events[column_name]=s
    converter('x2d')
    converter('y2d')
    converter('temp')
    converter('qobsrad')
    data={}
    for x in events.columns:
        data[x]=list(events[x].data)
    df=pd.DataFrame(data,columns=events.columns)
    root = os.path.abspath(os.curdir)
    df.to_csv("emir_dataset.csv",mode='a',index=False,header=True)
    print(df.head(3))
```

Fig. 9a: python function to extract fits file

add fits file to be extracted here

```
import os
path=r"datasets\fit_file"
dr_list=os.listdir(path)
dr_list

for x in dr_list:
    if str(x).endswith('.csv'):
        pass
    else:
       fit_cv(path+'\\'+x)
print('CSV created sucessfully!')
```

Fig. 9b: Extracting a fit file using the created python script

The above script was used to extract all fit files downloaded and convert the files to csv formats which can be used for analysis. The downloaded dataset were records of mars weather patterns from February 2021 to November 2021. The extracted datasets were combined to form a single excel file. The combined excel file has 70404 instances and 50 columns which contains data of mars daily and hourly weather patterns alongside some information about the used instrument. For proper analysis, some of these features were removed (the removal of such columns are discussed in the data preprocessing steps of this report). Some essential attributes of the dataset are briefly described in the table below:

S/NO	COLUMN NAME	DESCRIPTION
1.	utc	Time record was taken by the instrument. This is equivalent to the current earth time mars pattern was recorded
2.	temp	Mars Atmospheric temperature
3.	tsurfco2	Effective surface temperature outside CO_2 band(region)
4.	chi2temp	Quality of atmospheric temperature
5.	pres0	Surface pressure

6.	taudust	Dust column extinction optical depth
7.	tauice	Water ice column extinction optical depth
8.	tsurfa	Mars surface temperature
9.	dustuncert	Estimated uncertainty in dust optical depth
10.	iceuncert	Estimated uncertainty in water ice optical depth
11.	chi2a	Quality of aerosol fit
12.	tsurfh2o	Surface temperature near water on mars
13.	wateruncert	Estimated uncertainty in water vapor abundance in mars
14.	chi2b	Quality of water vapour fit
15.	qls	Solar longitude Ls
16.	latitude	Latitude from where space craft takes record
17.	longitude	Longitude from where space craft takes record
18.	sc_altitude	Range on mars surface

Table 1: EMM EMIRS dataset attributes

3.2 Experimental setup and tool used

The experiment was conducted in an 8gig ram Toshiba computer system with python 2.8 installed. For proper machine learning, tools such as sklearn, NumPy, matplotlib and pandas were installed. Also, Jupiter notebook was installed through the command prompt. This was necessary as all experiments conducted to predict variation in daily mars atmospheric temperature were carrid out within the Jupiter notebook environment.

3.3 Strategies used to conduct analysis

The figure below shows the workflow followed to conduct proper analysis on the dataset to perform proper analysis on the Martian dataset used:

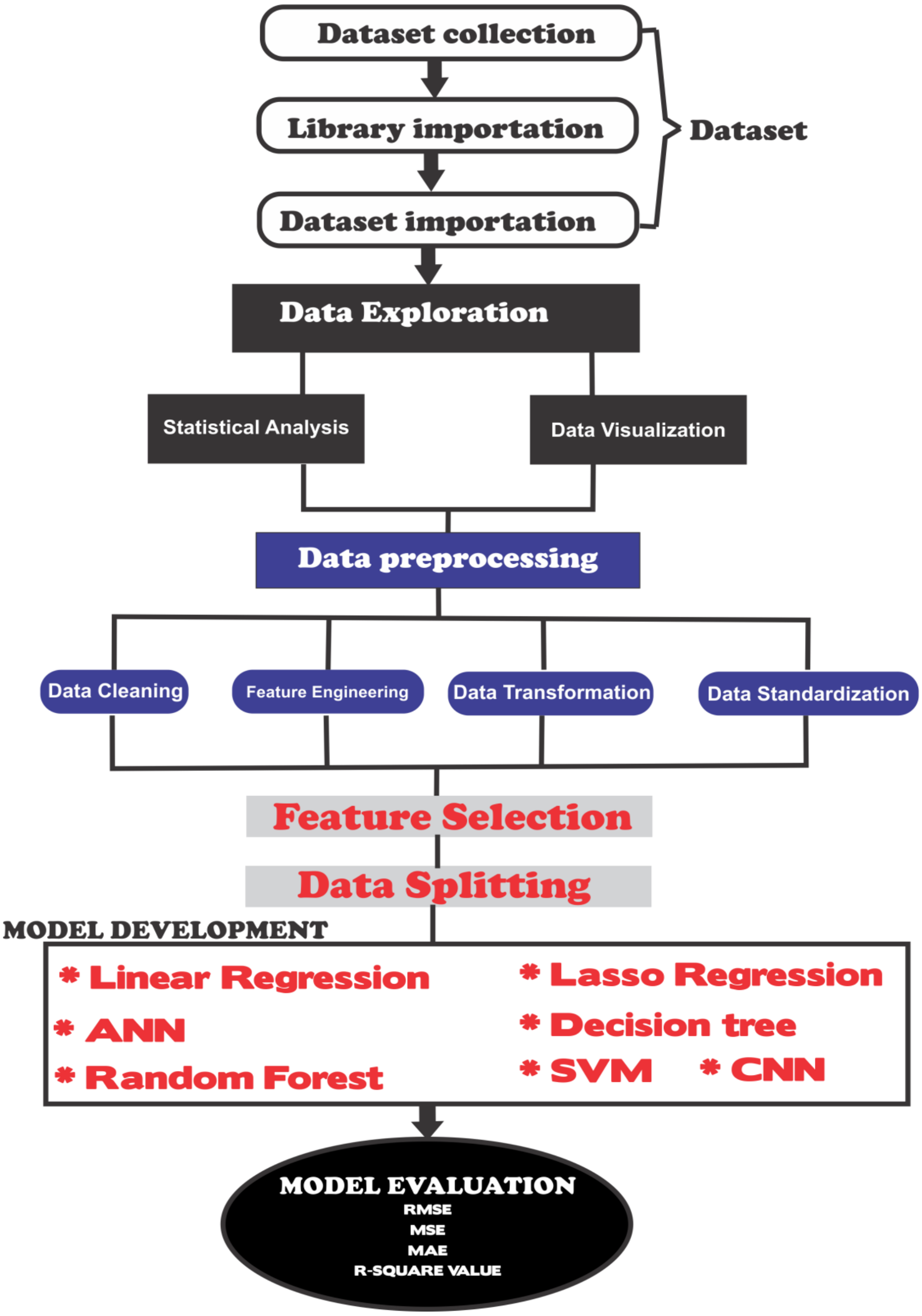


Fig. 10: workflow for Martian analysis

DATASET

The data used was obtained from Emirate mars mission official website and processed as discussed in section **3.1** above. The extracted excel file was loaded into the jupyter notebook environment alongside all libraries needed to perform analysis on the dataset. The dataset was imported into the jupyter environment using python pandas library. The figure below shows how the dataset was imported and all libraries used in experiment conducted.

```
import pandas as pd
import numpy as np
import warnings
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split,cross_val_score,StratifiedKFold,KFold,GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
import warnings
warnings.filterwarnings("ignore")
```

Data Importation

```
df=pd.read_excel("final.xlsx")
```

```
df
```

	sclk	sclk_sub	det_num	target_type_num	utc	incidence	emission	phase	bore_flag	nfov	...	longitude	qlt	qshgt	ta
0	666656480	27673	0	0	2021-02-20 11:24:20.929	0.0	0.0	0.0	0	0.0	...	0.0	0.0	39963.37109375	3996
1	666656482	27700	0	0	2021-02-20 11:24:22.929	0.0	0.0	0.0	0	0.0	...	0.0	0.0	39962.42578125	3996
2	666656484	27726	0	0	2021-02-20 11:24:24.929	0.0	0.0	0.0	0	0.0	...	0.0	0.0	39961.484375	39
3	666656486	27756	0	0	2021-02-20 11:24:26.93	0.0	0.0	0.0	0	0.0	...	0.0	0.0	39960.5390625	399
4	666656488	27777	5	3	2021-02-20 11:24:28.93	0.0	0.0	0.0	0	2.648076	...	0.0	0.0	39959.59375	3

Fig. 11: Dataset importation

DATA EXPLORATION

Data exploration refers to the first step in data analysis, it involves the use of data visualization and statistical techniques to define patterns underlying within a dataset. Such patterns include its size, shape, information etc. Which help provide understanding on the nature of the data. (Abhishek Sheshnath, 2022). Using python programming skills, data exploration involves the use of tools such as pandas, NumPy, matplotlib and seaborn for gaining revealing hidden patterns within a dataset. In this report, Exploration was conducted using the above-mentioned tools. For statistical analysis, panda's library was used as it provide other tools such as info, shape, describe and corr to get statistics underlying within the dataset. For visualization, matplotlib and seaborn were used to uncover patterns within the dataset. This visualization tools played significant part in understanding the variation that exist in the daily Martian atmospheric temperature.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 70404 entries, 0 to 70403
Data columns (total 27 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   utc                   70404 non-null  object
 1   temp                  46120 non-null  object
 2   shift                 1 non-null      object
 3   tsurfco2              46120 non-null  object
 4   chi2temp              46120 non-null  object
 5   pres0                 46120 non-null  object
 6   taudust               46120 non-null  object
 7   tauice                46120 non-null  object
 8   tsurfa                46120 non-null  object
 9   dustuncert            46120 non-null  object
 10  iceuncert             46120 non-null  object
 11  chi2a                 46120 non-null  object
 12  zdust                 46120 non-null  object
 13  prum                  9806 non-null   object
 14  zh2o                  9807 non-null   object
 15  tsurfh2o              9806 non-null   object
 16  wateruncert           9807 non-null   object
 17  chi2b                 46119 non-null  object
 18  qls                   64547 non-null  object
 19  latitude              46120 non-null  object
 20  longitude             46120 non-null  object
 21  sc_altitude           64547 non-null  object
 22  temp_quality_flag     4302 non-null   object
 23  taudust_quality_flag  31151 non-null  object
 24  tauice_quality_flag   29345 non-null  object
 25  prum_quality_flag     43105 non-null  object
 26  tsurfa_quality_flag   5574 non-null   object
dtypes: object(27)
memory usage: 15.0+ MB
```

Fig. 12: Information about the used dataset

The above statistical exploration was performed on the dataset to uncover the attributes of the dataset. The info() function used above tells of the number of values which are not empty within the dataset and also p[provides information on the various datatype within the dataset. From the figure, it can be clearly seen that the datasets are not in the right format hence feature transformation is to be conducted on the dataset (Xiao M. et. al., 2022). Results obtained from visualization of the dataset will be discussed in the result section.

DATA PREPROCESSING

Data preprocessing is a technique used to improve the quality of data. It involves making data ready and useful in the stage of model development. Preprocessing data in machine learning involves both data and feature engineering. In data engineering, raw data is processed and converted to prepared data for analysis. Feature engineering on the other hand involves tuning the prepared data to create the features that are expected by the Machine learning model (cloud Architecture center,2022). It involves steps like data cleaning (removing missing values or imputing missing value), outlier removal (Fan C. et. al., 2021), removal of outliers, normalizing or standardizing data, and feature transformation (Xiao M. et. al., 2022). In this report, data preprocessing was conducted using the knowledge of python programming alongside some functions within the pandas module to transform, drop and make the entire dataset ready for proper analysis. Preprocessing techniques applied include: data cleaning, outlier removal, data transformation, data standardization and feature engineering were performed on the dataset. Various functions such as drop, rename and replace within the pandas library were used in performing preprocessing on the dataset. This step is crucial and essential because when a model is developed with an unprocessed dataset, it always result to poor performance by the developed model.

```
In [9]: df=df.replace('0.0',np.nan)

In [ ]: df=df.drop_duplicates()

In [10]: df=df.replace('0',np.nan)

In [11]: df.drop(['incidence','emission','phase','bore_flag','nfov','scan_period','ra','dec','x2d','sclk','sclk_sub','det_num','target_ty
```

Fig. 13a: Removal of missing values

```
df['utc']=pd.to_datetime(df['utc'])
```

```
date=[]
for x in df.utc:
    x=str(x)
    x=x.split(' ')
    date.append(x[0])
df['date']=date
```

Fig. 13b: Data transformation

```
cols=list(df.columns)
cols=cols[1:]
convert_types=dict()
for x in cols:
    convert_types[x]=float
df=df.astype(convert_types)
```

Fig. 13c: Datatype transformation

Removal of outliers

```
df=df.drop(['utc','date','sc_altitude'],axis=1)
for x in df.columns:
    ll=df[x].quantile(0.05)
    ul=df[x].quantile(0.95)
    df=df[(df[x]>ll) & (df[x]<ul)]
df
```

Fig. 13d: Removal of outliers

FEATURE SELECTION

Before the development of a machine learning model, important features are needed to be selected. Improper feature selection can affect the overall performance of a machine learning model (Ayat H., 2023). In this report, features such as pressure, various temperatures available within the dataset etc. Were selected. This selection was done following the visualization of the correlation that exist among each attribute to the atmospheric temperature which was to be predicted. The figure below shows the correlation that exist among the features present within the dataset.

Fig. 14 Removal of outliers

The above figure was instrumental in feature selection. Features were selected by dropping attributes with low significance within the dataset. Below figure shows how the feature selection was performed:

```
x=df2.drop(['temp','longitude','utc','date','sc_altitude'],axis=1).values
y=df2['temp'].values
```

Fig. 15: Feature selection

DATA SPLITTING

This stage will come up next after feature selection. It involves splitting the selected features into training and testing splits. In this report, the dataset was splitted in the ratio of 80% to 20% for training and testing purpose respectively. This splitting was conducted using the sklearn library which provided a function present within the model section library called train_test_split. Below figure shows how the selected features were splitted into training and testing sets:

```
x=df2.drop(['temp','longitude','utc','date','sc_altitude'],axis=1).values
y=df2['temp'].values
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.2,random_state=0)
scaler=StandardScaler()
scaler.fit(x_train)
x_train=scaler.transform(x_train)
x_test=scaler.transform(x_test)
```

Fig. 16: Data splitting and standardization

MODELS DEPLOYMENT

Machine learning consists of different models for carrying out regression. For this work 7 machine learning models were used. The essence of using this model is to test and make selection of the best performing model from the used models as conclusions about performance of a model cannot be made without testing it. The used models are: linear regression, Random Forest, Decision Tree, Artificial Neural Networks, Lasso algorithm, Convolutional neural network and support vector machine (SVM). These models were accessed from the sklearn module especially available for all machine learning tasks in python. All these algorithms were trained and tested using the splitted EMM EMIRS dataset. The figures below show how each model was developed:

```
rf=RandomForestRegressor()
rf.fit(x_train,y_train)
rf_pred=rf.predict(x_test)
print('Mean Absolute Error(Random Forest Algorithm:)',mean_absolute_error(y_test,rf_pred))
print('Mean Squared Error(Random Forest Algorithm:)',mean_squared_error(y_test,rf_pred))
print('Root Mean squared Error(Random Forest Algorithm:)',np.sqrt(mean_squared_error(y_test,rf_pred)))
print('R squared value(Random Forest Algorithm:)',r2_score(y_test,rf_pred))
```

Fig. 17a: Development of Random forest Model

```
lr=LinearRegression()
lr.fit(x_train,y_train)
lr_pred=lr.predict(x_test)
print('Mean Absolute Error(Linear regrssion Algorithm:)',mean_absolute_error(y_test,lr_pred))
print('Mean Squared Error(Linear regression Algorithm:)',mean_squared_error(y_test,lr_pred))
print('Root Mean squared Error(Linear regression Algorithm:)',np.sqrt(mean_squared_error(y_test,lr_pred
print('R squared value(Linear regression Algorithm:)',r2_score(y_test,lr_pred))
```

Fig. 17b: Development of Linear regression Model

```
svm=SVR()
svm.fit(x_train,y_train)
svm_pred=svm.predict(x_test)
print('Mean Absolute Error(Support vector machine Algorithm:)',mean_absolute_error(y_test,svm_pred))
print('Mean Squared Error(Support vector machine Algorithm:)',mean_squared_error(y_test,svm_pred))
print('Root Mean squared Error(support vector machine Algorithm:)',np.sqrt(mean_squared_error(y_test,sv
print('R squared value(support vector machine Algorithm:)',r2_score(y_test,svm_pred))
```

Fig. 17c: Development of SVM Model

```
ann=MLPRegressor()
ann.fit(x_train,y_train)
ann_pred=ann.predict(x_test)
print('Mean Absolute Error(Neural Network Algorithm:)',mean_absolute_error(y_test,ann_pred))
print('Mean Squared Error(Artificial Neural network Algorithm:)',mean_squared_error(y_test,ann_pred))
print('Root Mean squared Error(Artificial Neural network Algorithm:)',np.sqrt(mean_squared_error(y_test
print('R squared value(Artificial neural network Algorithm:)',r2_score(y_test,ann_pred))
```

Fig. 17d: Development of ANN Model

```
dt=DecisionTreeRegressor()
dt.fit(x_train,y_train)
dt_pred=dt.predict(x_test)
print('Mean Absolute Error(Decision Tree Algorithm:)',mean_absolute_error(y_test,dt_pred))
print('Mean Squared Error(Decision Tree Algorithm:)',mean_squared_error(y_test,dt_pred))
print('Root Mean squared Error(Decision Tree Algorithm:)',np.sqrt(mean_squared_error(y_test,dt_pred)))
print('R squared value(Decision Tree Algorithm:)',r2_score(y_test,dt_pred))
```

Fig. 17e: Development of Decision tree Model

```
lasso=Lasso()
lasso.fit(x_train,y_train)
lasso_pred=lasso.predict(x_test)
print('Mean Absolute Error(LASSO Algorithm:)',mean_absolute_error(y_test,lasso_pred))
print('Mean Squared Error(LASSO Algorithm:)',mean_squared_error(y_test,lasso_pred))
print('Root Mean squared Error(LASSO Algorithm:)',np.sqrt(mean_squared_error(y_test,lasso_pred)))
print('R squared value(LASSO Algorithm:)',r2_score(y_test,lasso_pred))
```

Fig. 17f: Development of LASSO Model

```
cnn= Sequential()
cnn.add(Dense(1, input_shape=(x.shape[1],)))
cnn.compile(optimizer='adam', loss='mean_absolute_error')
cnn.fit(x_train, y_train, epochs=100)
test_loss = cnn.evaluate(x_test, y_test)
print('Test loss:', test_loss)
cnn_pred = cnn.predict(x_test)
print('Mean Absolute Error(Deep Neural Network Algorithm:)',mean_absolute_error(y_test,cnn_pred))
print('Mean Squared Error(Deep Neural Network Algorithm:)',mean_squared_error(y_test,cnn_pred))
print('Root Mean squared Error(Deep Neural Network Algorithm:)',np.sqrt(mean_squared_error(y_test,cnn_p
print('R squared value(Deep Neural Network Algorithm:)',r2_score(y_test,cnn_pred))
```

Fig. 17g: Development of CNN Model

HYPER PARAMETER FINE TUNING

Hyper parameter fine tuning is a stage in machine learning that involves improving the performance of a machine learning model (Gera P., 2022). In this work, grid search technique was used on each developed model to find the best parameters that will provide optimal performance of the model.

```
: par_rf= {
      'n_estimators':[150,100,250,300,400],
  }

  grid_rf= GridSearchCV(rf, par_rf, n_jobs=-1, cv=10,scoring='accuracy')
  grid_rf.fit(x_train,y_train)
  best_r= grid_rf.best_params_
  print('Best parameters  for RF found:',best_r)
```

Fig. 18a: Development of CNN Model

```
par_lr= {
    'fit_intercept':[x for x in range(1,20)],
}

grid_lr= GridSearchCV(lr, par_lr, n_jobs=-1, cv=10,scoring='accuracy')
grid_lr.fit(x_train,y_train)
best_lr=grid_lr.best_params_
print('Best parameters  for LR found:', best_lr)
```

```
Best parameters  for LR found: {'fit_intercept': 1}
```

Fig. 18b: Development of CNN Model

```
par_svm= {
    'kernel':['linear','polynomial','radial'],
}

grid_svm= GridSearchCV(svm, par_svm, n_jobs=-1, cv=10,scoring='accuracy')
grid_svm.fit(x_train,y_train)
best_svm=grid_svm.best_params_
print('Best parameters  for SVM found:', best_svm)
```

Fig. 18c: Development of CNN Model

```
par_dt= {
    'max_depth':[1,3,5,6,10],
    'min_samples_split':[2,4,6,8],
    'max_features':[2,5,6,9],
    'min_samples_leaf':[2,4,6,8],
}

grid_dt= GridSearchCV(dt, par_dt, n_jobs=-1, cv=10,scoring='accuracy')
grid_dt.fit(x_train,y_train)
best_dt=grid_dt.best_params_
print('Best parameters  for DT found:', best_dt)
```

Fig. 18d: Development of CNN Model

```
par_lasso= {
    'alpha':[0.0001,0.001,0.01,0.1,1,10,100],
}

grid_lasso= GridSearchCV(lasso, par_lasso, n_jobs=-1, cv=10,scoring='accuracy')
grid_lasso.fit(x_train,y_train)
best_lasso=grid_lasso.best_params_
print('Best parameters  for LASSO found:', best_lasso)
```

Fig. 18e: Development of CNN Model

MODEL EVALUATION

Model evaluation stage involves scrutinizing a model so as to check its performance. In Machine Learning regression task, the commonly used Evaluation techniques are Mean Squared error (MSE), Root Mean square error (RMSE), Mean absolute error (MAE) and R squared value. These techniques are used to check if a model is performing as required or is under performing. Mean absolute error represents the average of the absolute difference between the actual and predicted values in the dataset. Mean Squared Error represents the average of the squared difference between the original and predicted values in the data set. Root Mean Squared Error is the square root of Mean Squared error. It measures the standard deviation of residuals. R-squared represents the proportion of the variance in the dependent variable (Akshita C., 2020). The above-mentioned techniques were used on the developed models to examine the performance of each model. Below are brief descriptions of these techniques as described by Akshita c. (2020).

- **The Mean absolute error** represents the average of the absolute difference between the actual and predicted values in the dataset. It measures the average of the residuals in the dataset.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}|$$

Where,

$\hat{y}$ $-$ *predicted value of* y

$\bar{y} -$ *mean value of* y

- **Mean Squared Error (MSE)** represents the mean of the squared difference between the predicted values by a model and the actual values within a dataset. It measures the variance of the residuals.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

- **Root Mean Squared Error** represents the root square of the MSE of a model. It measures the standard deviation of residuals.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

- **R-squared Value** represents the proportion of the variance in the dependent variable which is explained by the linear regression model. The closer an R-squared value is to 1, the better the accuracy of the model.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

CHAPTER 4

Result Analysis And Discussion

4.1 Results obtained from Experiment conducted on variation in temperature.

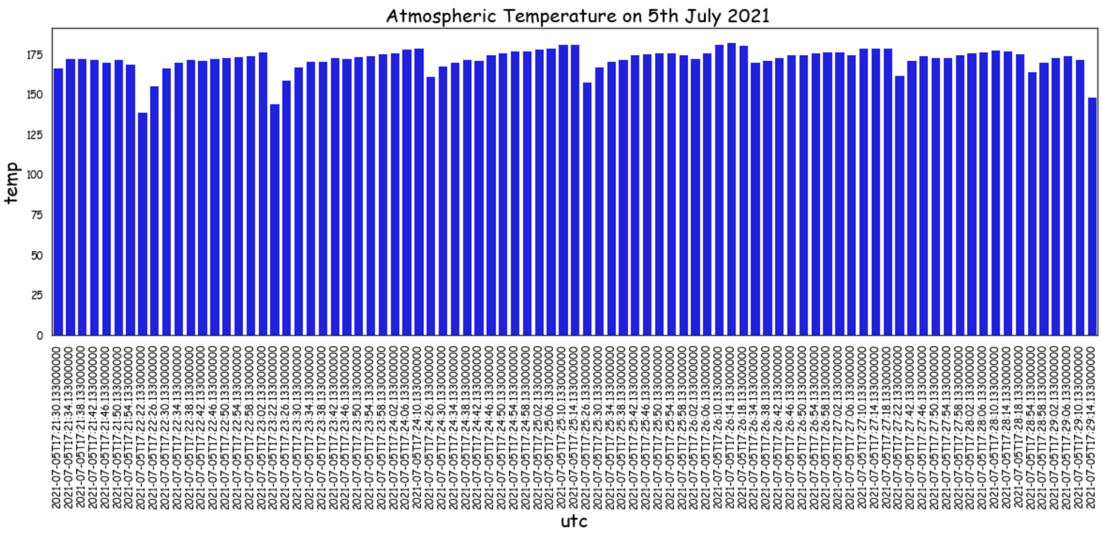


Fig. 19a: Atmospheric Temperature for 5th July 2021 (Bar chart)

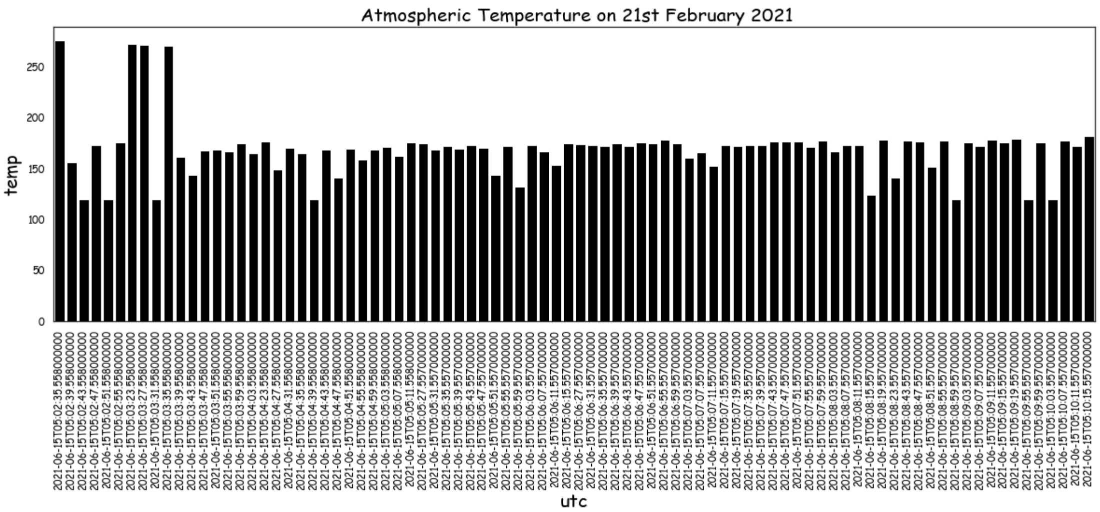


Fig. 19b: Atmospheric Temperature for 21st July 2021 (Bar chart)

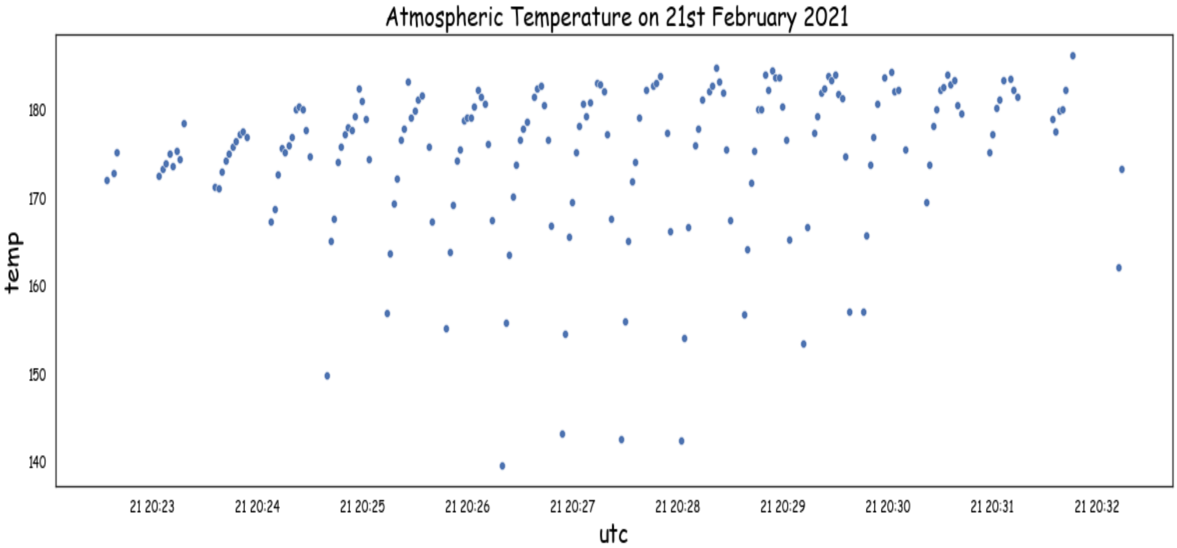


Fig. 19c: Atmospheric Temperature for 5th July 2021 (Scatter plot)

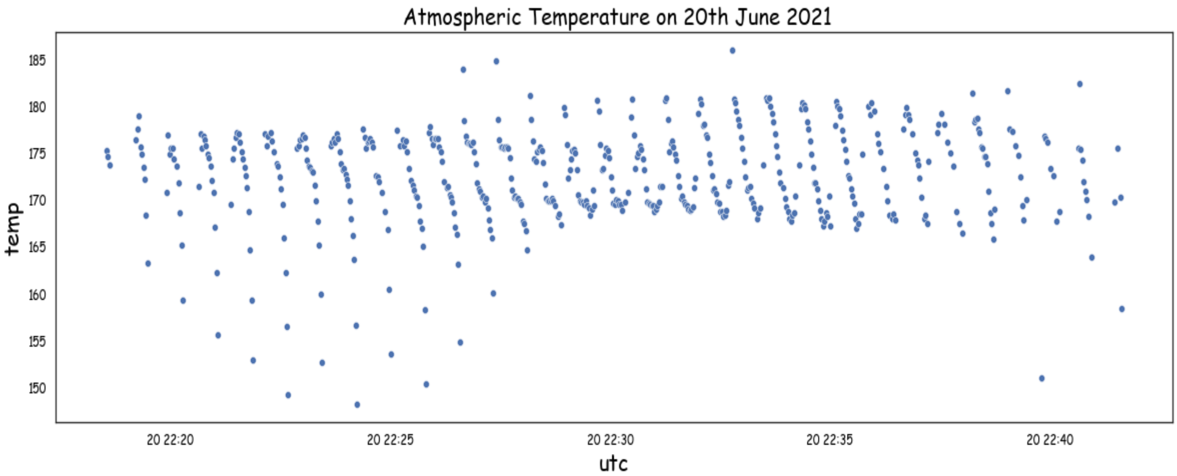


Fig. 19d: Atmospheric Temperature for 20th July 2021 (Scatter plot)

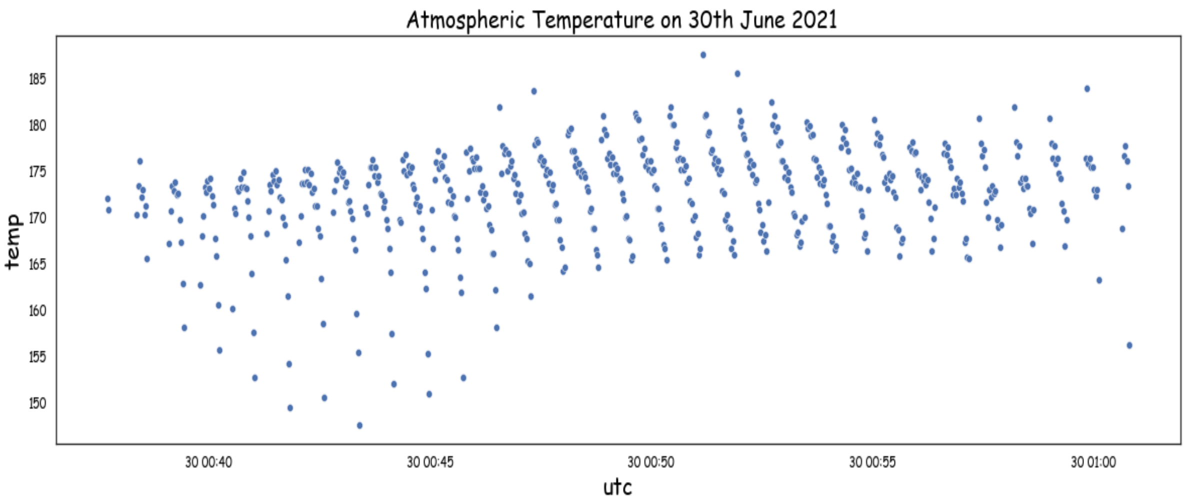


Fig. 19f: Atmospheric Temperature for 30th July 2021 (Scatter plot)

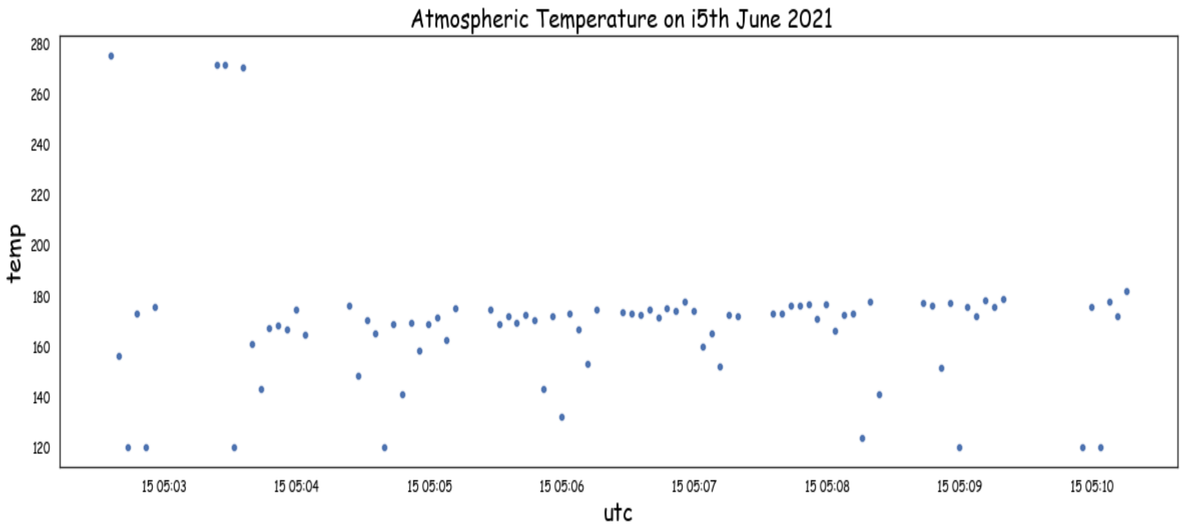


Fig. 19g: Atmospheric Temperature for 15th July 2021 (Scatter plot)

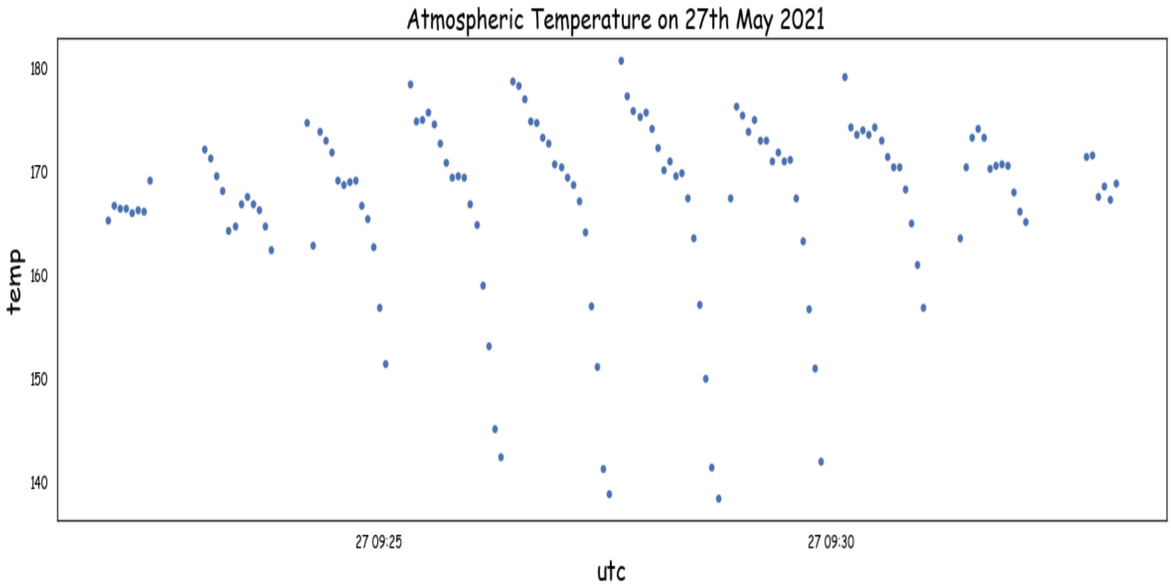


Fig. 19h: Atmospheric Temperature for 27th July 2021 (Scatter plot)

The above results were obtained from taking 6 days (I.e. sols) within the dataset and visualizing their atmospheric temperatures. From the above figures 16a-16h, it can be clearly seen that there is a variation in the daily temperature value for mars. From the figures, it can be observed that within a martian day, the temperature seems to be lower at early hours and higher at late hours. Although some days experience higher temperature values early hours compare to late hours e.g 21st and 15th July 2021. This is a clearly indication of diurnal variation in temperature of mars as presented by Dimitra A. et. al., 2023 and Michael D et. al, 2022. This variation in mars temperature are largely sponsored by the amount of dust present in its atmosphere. According to Royal Belgian institute for space Aeronomy (2022), variation in day and night temperature of mars is mainly due to the weakness of greenhouse effect which contributes to the Martian soil storing very few amounts of energy.

4.2 Results obtained from Experiment conducted for similarities among mars temperatures

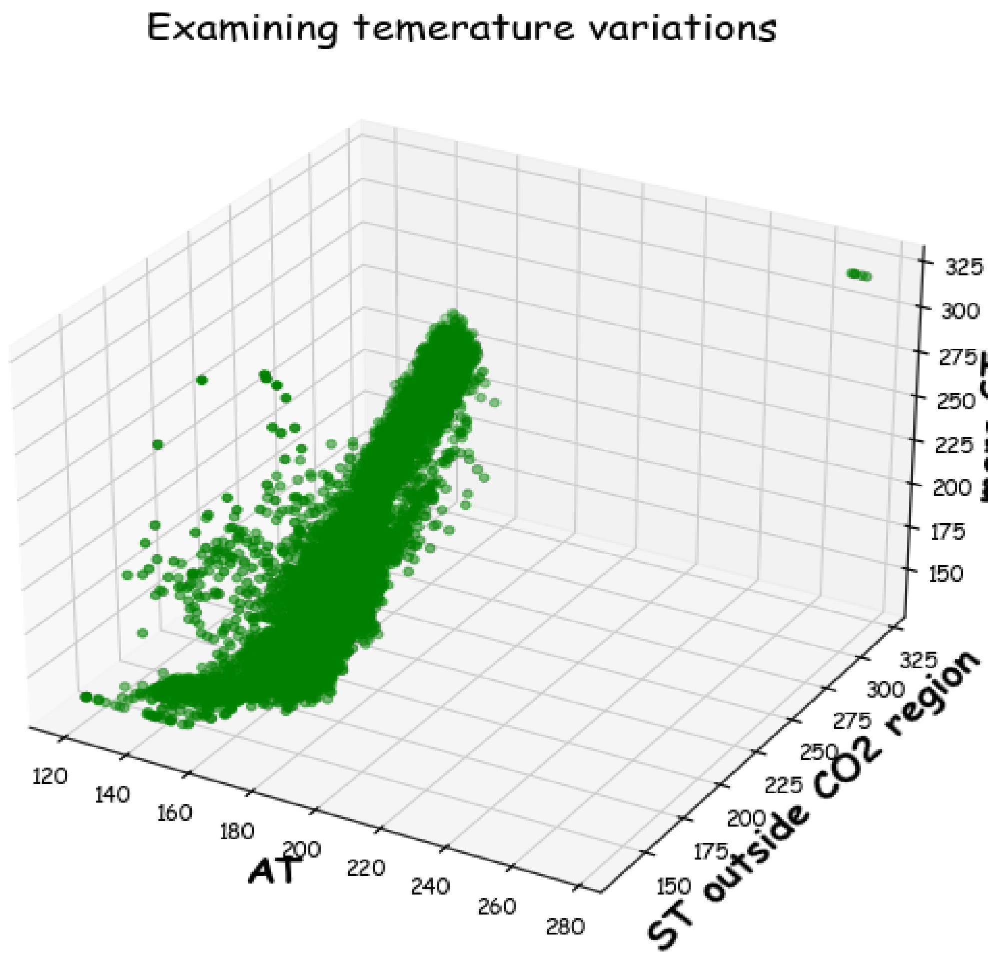


Fig. 20a: Relationship between AT, ST for CO2 and ST

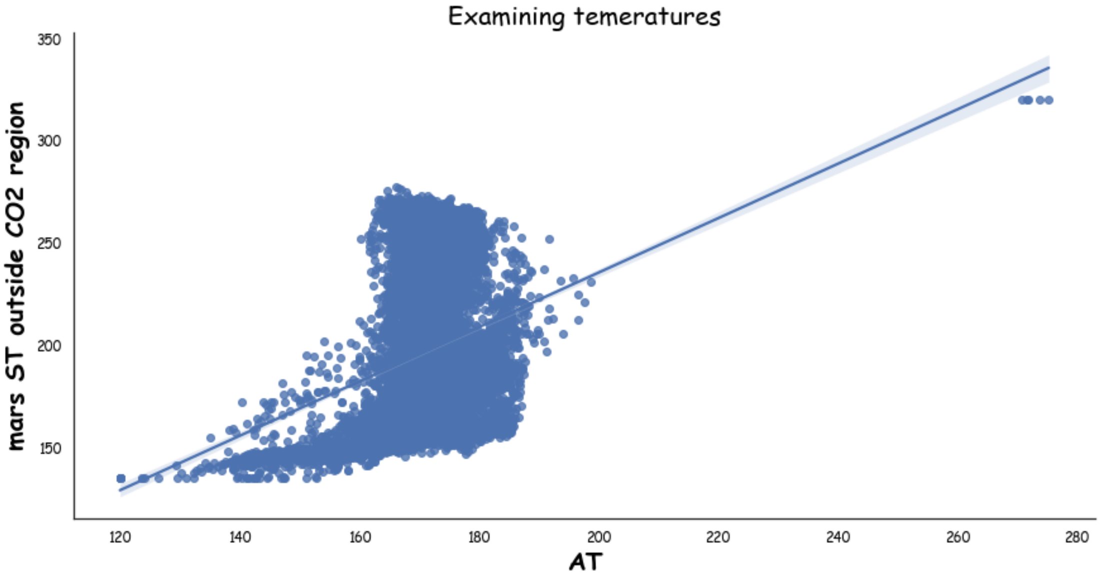


Fig. 20b: Linear Relationship between AT and ST outside CO_2

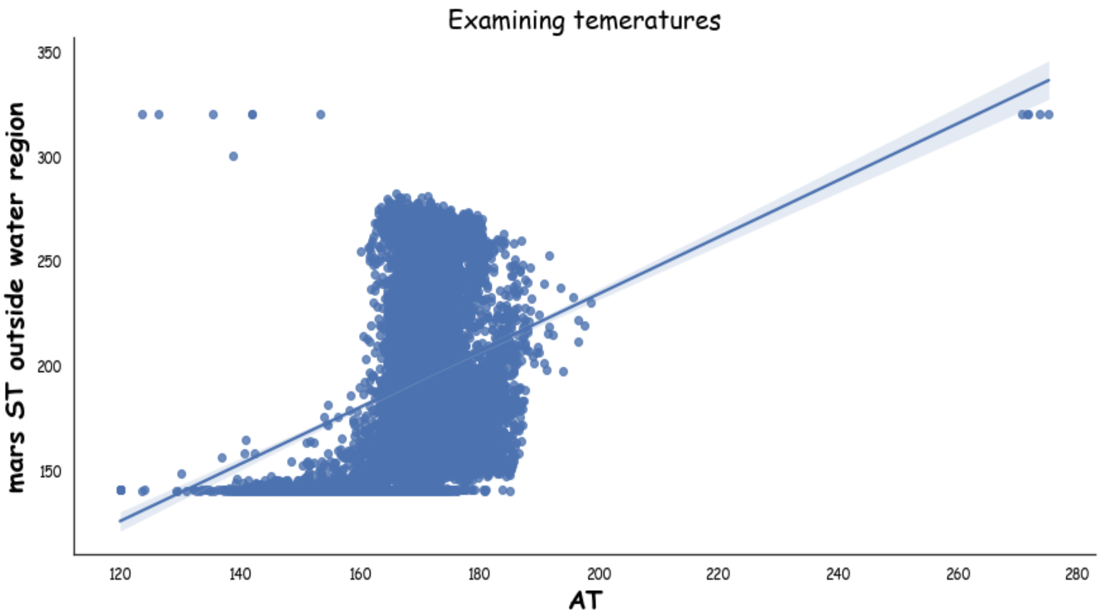


Fig. 20c: Linear Relationship between AT and ST for H_2O

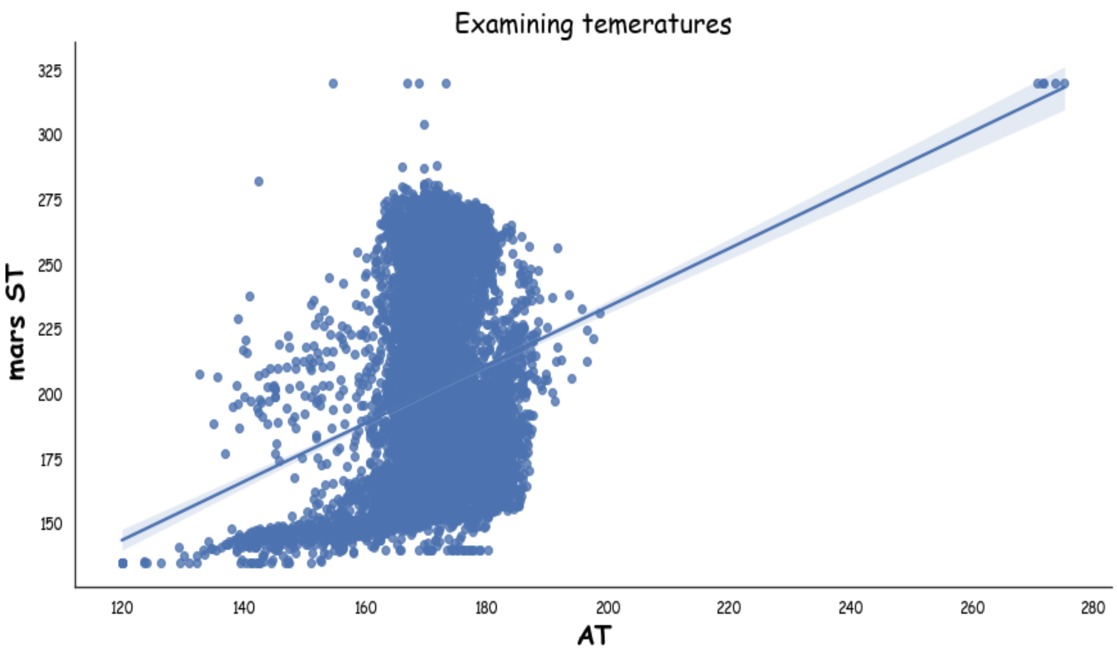


Fig. 20d: Linear Relationship between AT and ST in Mars

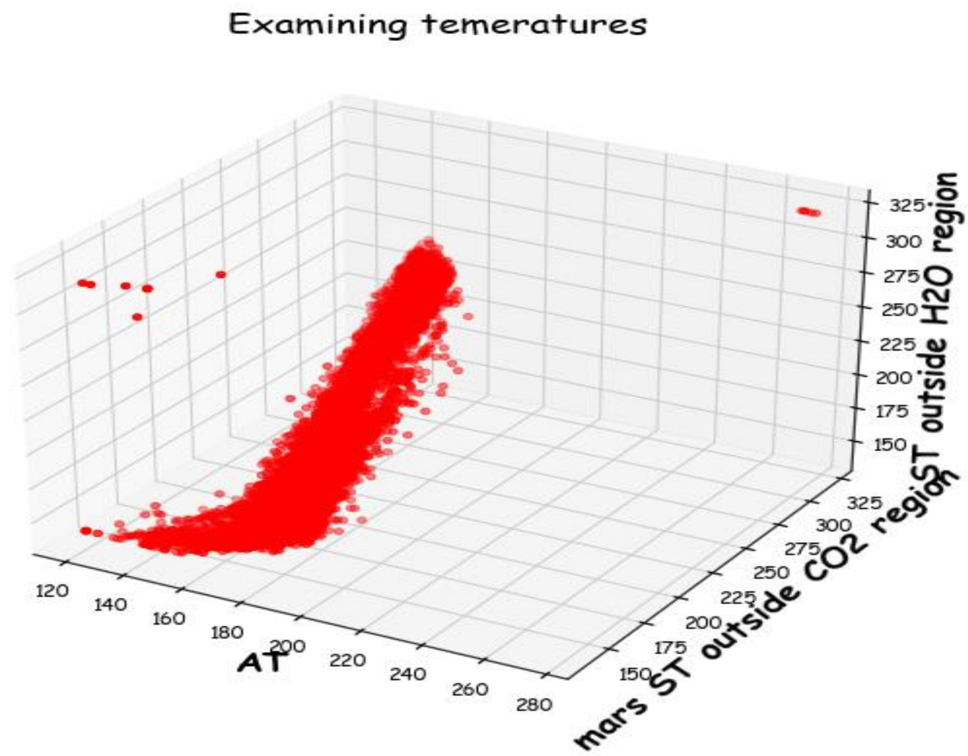


Fig. 20e: Relationship between AT, ST for CO2 and ST for H2O

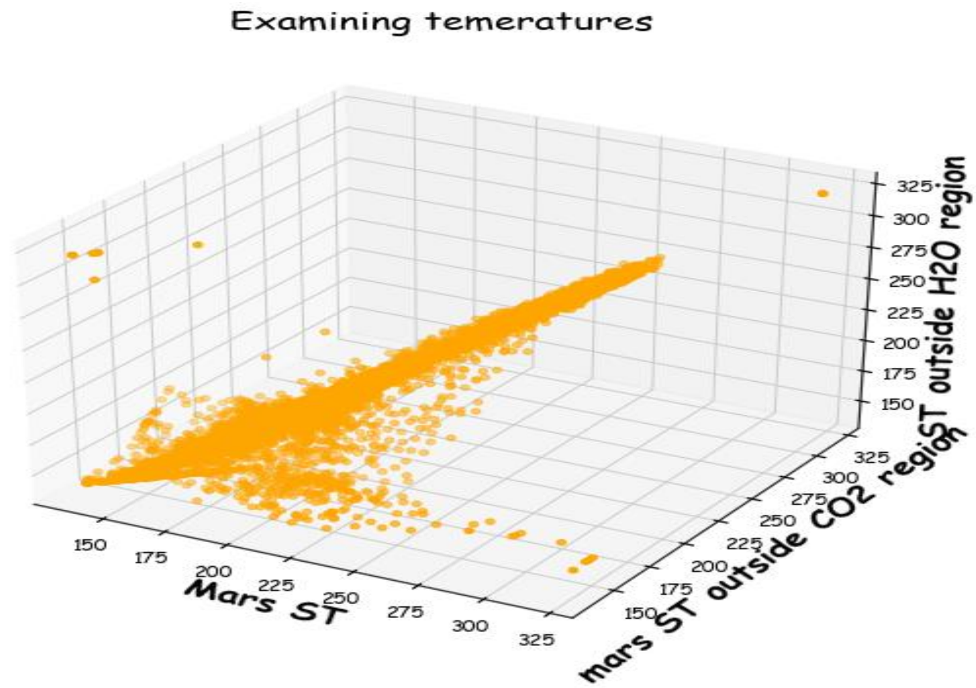


Fig. 20f: Relationship between ST, ST for CO2 and ST for H2O

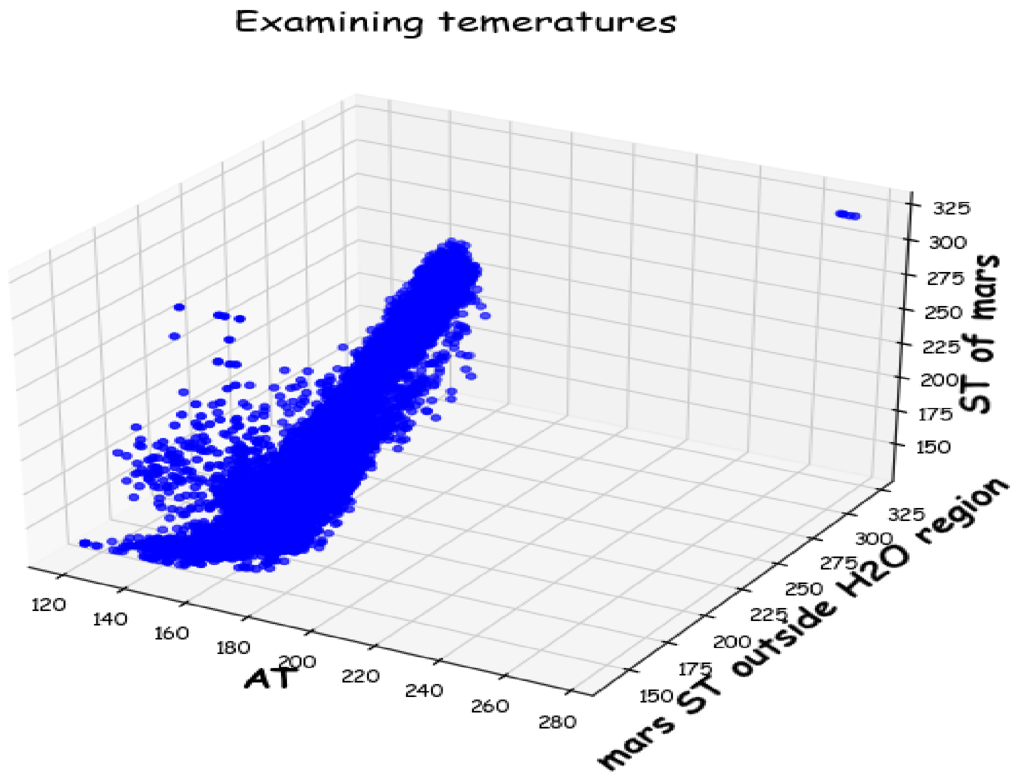


Fig. 20g: Relationship between AT, ST for H2O and ST.

The above results were obtained from conducting an experiment to verify if relationship exists among all temperatures recorded by the EMIRS instrument.

Following the results shown in fig. 20a-fig. 20g, it can be seen that a linear relationship exists among all the various temperatures present within the dataset. The EMIR instrument recorded 4 different temperatures which are Atmospheric temperature (AT), Surface Temperature(ST), Surface temperature outside Water region (ST outside H2O) and Surface temperature outside Carbon-dioxide region (ST outside CO2). From the above results, this temperatures complement each other as the various figures produced a linear form of relationship indicating that their values are closely related and are affected by one another.

4.3 Results obtained from model development

For model development, two different experiments were conducted on the dataset. The first experiment was conducted without removing outliers present within the dataset while the second experiment removed outliers present within the dataset before model development. The results obtained from this two experiments are presented below:

4.3.1 Results from experiment without removal of outliers

The results below were obtained after models were developed without removal of outliers present within the dataset:

Model	**MAE**	**MSE**	**RMSE**	**R_2 score**
Random forest	1.3179	4.1967	2.0486	0.9448
Linear Regression	3.4310	50.0191	7.0724	0.3423
Support Vector Machine (SVM)	3.3729	45.1238	6.7174	0.4067
Artificial Neural Network (ANN)	149.4879	22704.6600	150.6807	-297.5380
Decision Tree	1.4242	3.6380	1.9073	0.9521
LASSO regression	3.0438	18.8760	4.3447	0.7518
Convolutional neural Network (CNN)	166.4401	27810.7507	166.7656	-364.6769

Table 2: Results before removal of outliers

The above table presents the results obtained from development and evaluation of used machine learning models without handling outliers present within the dataset.

4.3.2 Results after handling outliers present within dataset

Model	MAE	MSE	RMSE	R_2 score
Random forest	1.1188	2.4228	1.5566	0.8275
Linear Regression	1.5275	3.5820	1.8926	0.7449
Support Vector Machine (SVM)	1.3887	3.9814	1.9953	0.7165
Artificial Neural Network (ANN)	31.6600	1530.9090	39.1268	-108.0029
Decision Tree	1.5839	5.1040	2.2592	0.6366
LASSO regression	2.0469	7.2166	2.6864	0.4862
Convolutional neural Network (CNN)	169.4396	28725.9820	169.4874	-2044.3299

Table 3: Result after removal of outliers

Table 2 and Table 3 above present the results obtained from performing two different experiments on the EMIRS dataset. In table 1 where outliers were not handled, it could be observed that despite the high R2 scores obtained by Random forest and Decision tree, the errors obtained are high compared to the errors obtained in table 2. This is an indication that the R2 scores in Table 1 are largely influenced by over fitting (Frost J., 2017). The R2 scores obtained in table 2 are low compared to Table 1 but the error scores are minimal which is an indication that outliers present within in the dataset at first contributed to the overfitting issue present within the dataset. Hence removal of outliers was an important step in this work as it contributed to the improvement of the accuracies. From Table 2, Neural networks (ANN and CNN) used produced negative R2 scores and high error values which implies that the models are not suitable for the datasets. These issues arose as a result of the models' predictions been worse than a constant function that always predicts the mean of the data (Desmos, 2022) hence they are not suitable for the work under review. For the 5 other used models, Random forest outperformed the other models by producing minimal error values hence making it the best model amist the others. Also the R2 score produced by the Random forest model is very high and realistic when compared to the R2 score obtained in Table 1 which was largely sponsored by overfitting. The comparison between the obtained scores by each model are shown in the figures below:

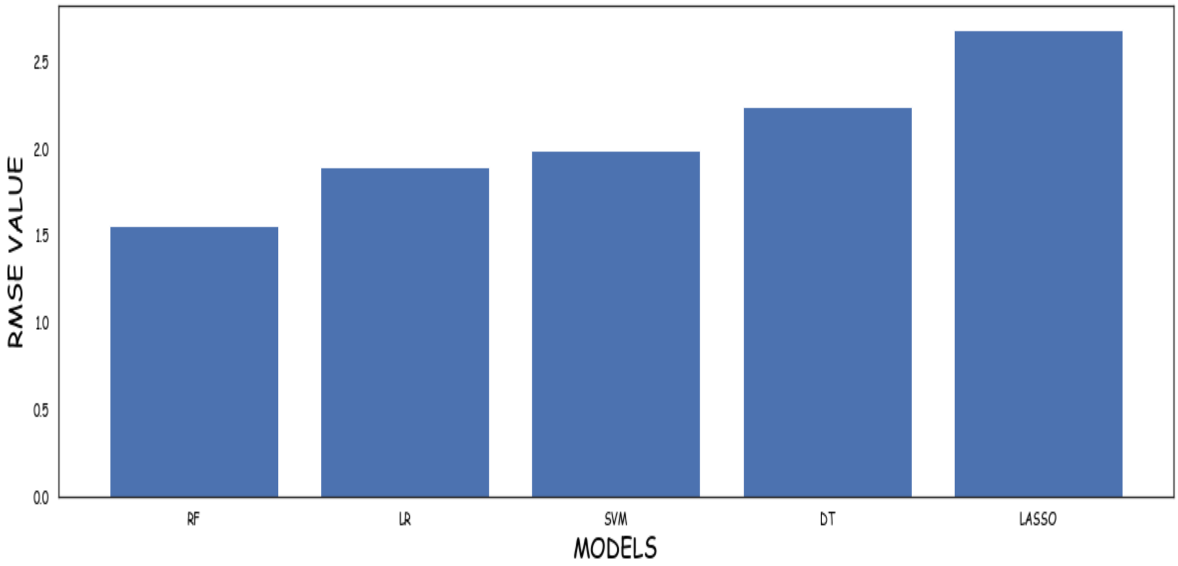


Fig. 21a: RMSE for the 5 models with high performance

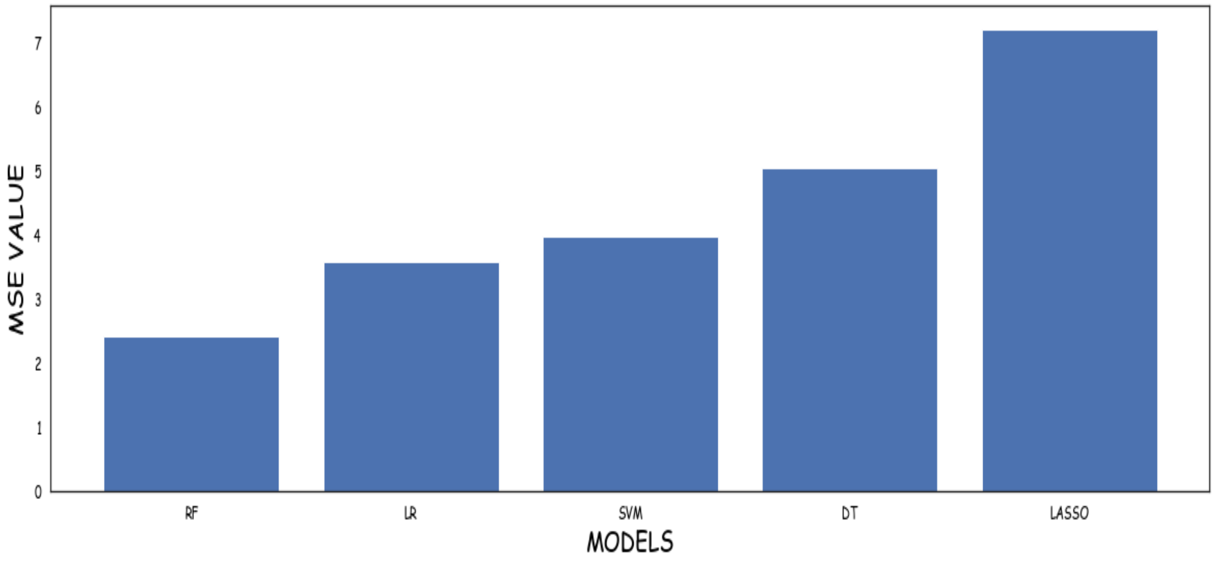


Fig. 21b: MSE for the 5 models with high performance

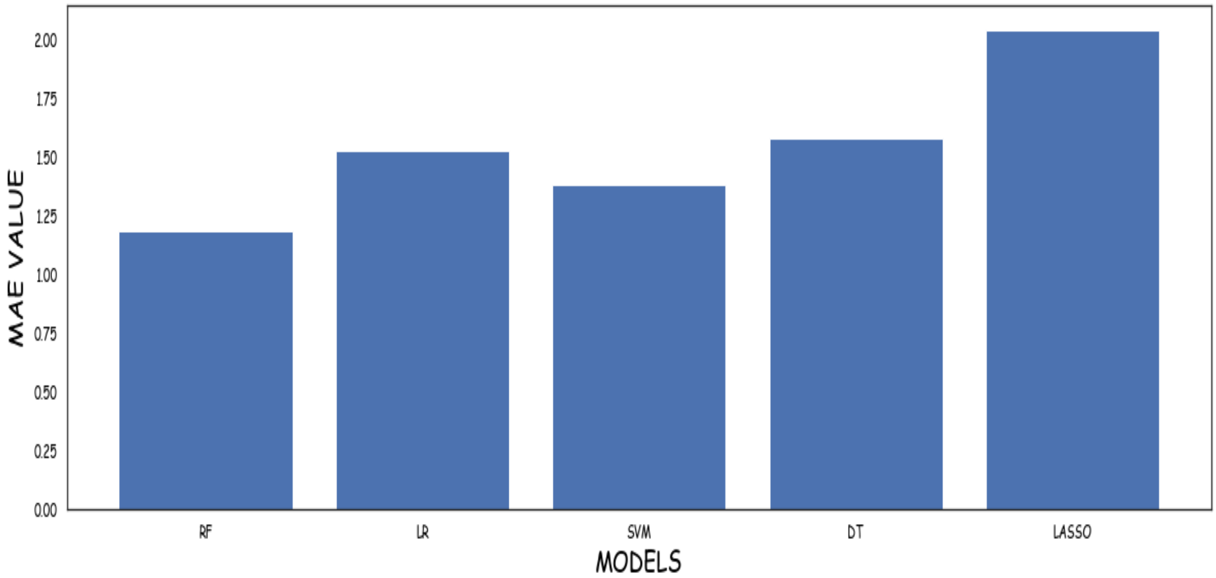


Fig. 21c: MAE for the 5 models with high performance

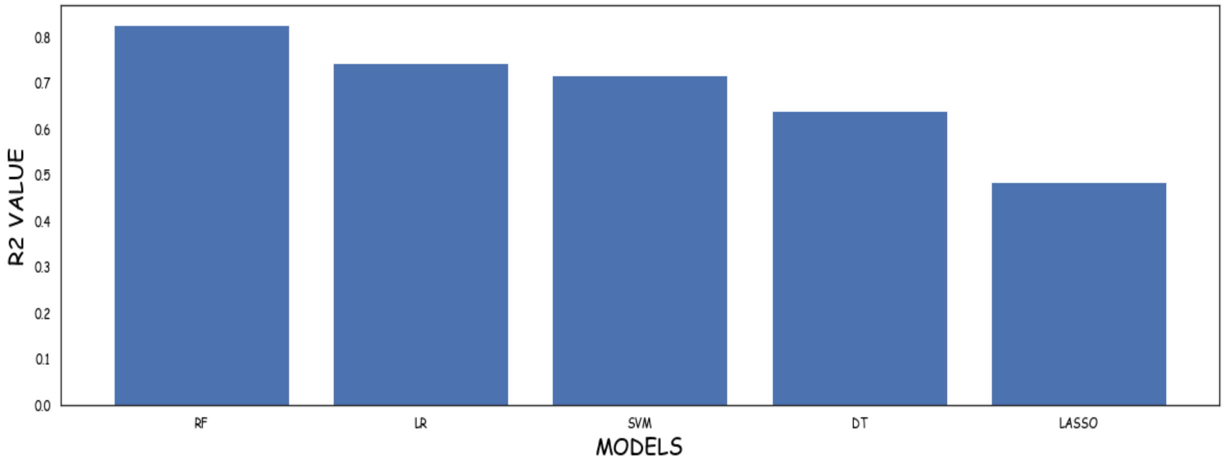


Fig. 21d: R2 Values for the 5 models with high performance

The high performance of Random Forest can be clearly seen across all 4 figures.

Due to this performance the 5 models were subject to hyper parameter fine tuning using GridsearhCV (Joseph R., 2018) and the results obtained are presented in Table 4 below:

Model	**MAE**	**MSE**	**RMSE**	**R_2 score**
Random forest	1.1939	2.4105	1.5525	0.8283
Linear Regression	1.5275	3.5819	1.8926	0.7449
Support Vector Machine (SVM)	1.5038	3.6944	1.9221	0.7369
Artificial Neural Network (ANN)	--------------	---------------	--------------	--------------

Decision Tree	1.5898	5.0146	2.2393	0.6429
LASSO regression	1.5271	3.5809	1.8923	0.7450
Convolutional neural Network (CNN)	--------------	-------------	--------------	--------------

Table 4: Result after Hyper parameter fine tuning

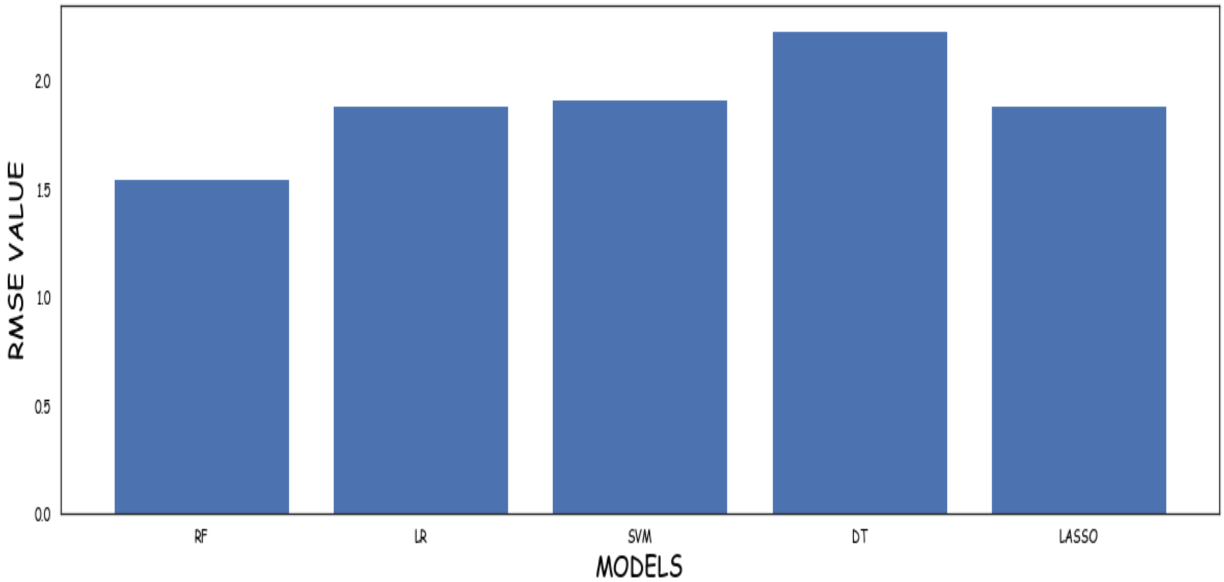


Fig. 22a: RMSE for the 5 models with high performance (After hyper parameter fine tuning)

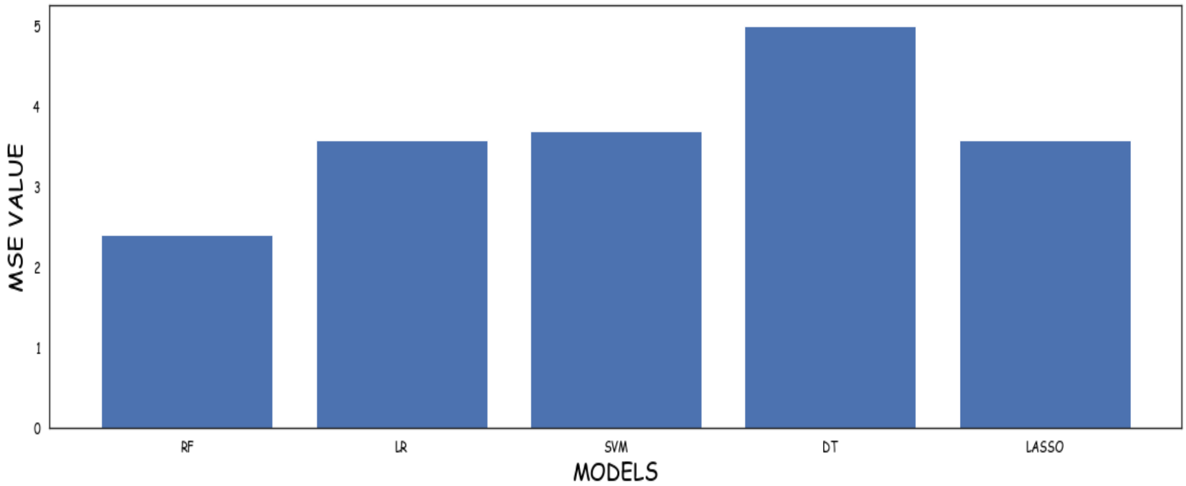


Fig. 22b: MSE for the 5 models with high performance (After hyper parameter fine tuning)

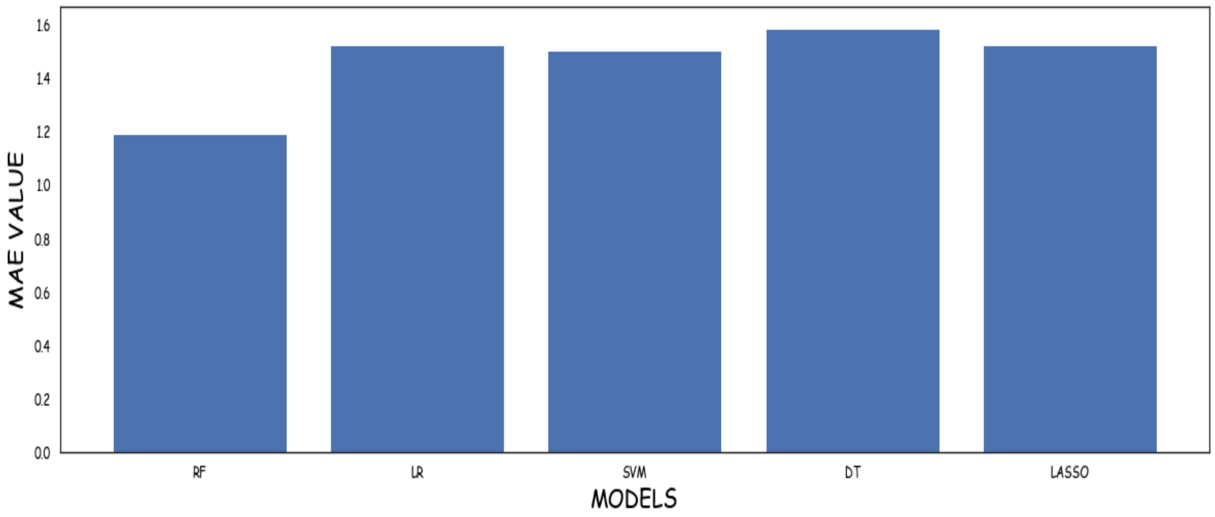


Fig. 22c: MAE for the 5 models with high performance (After hyper parameter fine tuning)

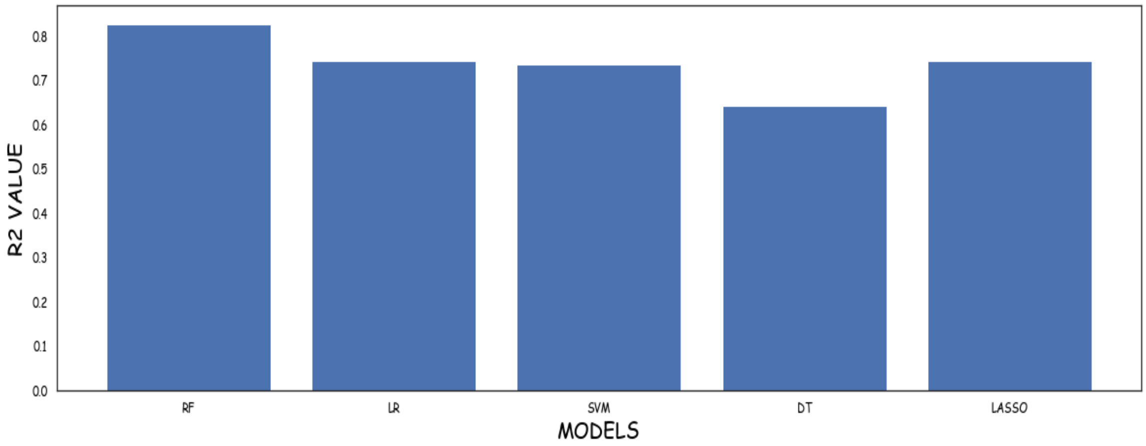


Fig. 22d: R2 Values for the 5 models with high performance (After hyper parameter fine tuning)

Table 4 and figures 22a -22d shows the improved performance of the 5 models after hyper parameter fine tuning. Random forest which is the best performing model used improved in terms of MSE, RMSE and R2 score. Compared to the figures shown in 21a-21b and results presented in table 3 above, the results presented in fig. 22a-22b and Table 4 produced improve accuracies. The MSE of the used Random Forest model improved with a reduction by 0.0123, the RMSE improved with a reduction of 0.0041 while the R2 score improved with an increase of 0.0008. This is an indication that the hyper parameter technique deployed produced the optimal results for the models used.

4.4 Critical Discussion

This paper investigated the EMM EMIRS dataset to explore the daily Martian atmospheric temperature variation. 7 models were developed and used in two experiments on the dataset. In the first experiment, the seven models were developed with outliers present within the dataset, while in the second experiment, the models were developed with outliers removed from the dataset. This was necessary to see if the dataset's outliers would affect the models' performance positively or negatively. Following the two experiments, in terms of R-squared scores, the first experiment produced higher values, as seen in **Table 2**, compared to scores obtained in the second experiment,
as seen in **Table 3**. In terms of error scores (RMSE, MSE and MAE), the second experiment produced more miniature scores, indicating better performance. Following these two results, results obtained from the first experiments, especially the high R squared values obtained, are primarily influenced by overfitting. The second experiment was further subjected to hyperparameter fine-tuning, producing an improved performance for each model, as shown in **Table 4**.

Random forest performed better in both experiments and after fine-tuning than other models. It produced r2 scores of 83%, 82% r and 82.5%, respectively. This model's high performance indicated its ability to overcome fitting issues that may cross its path. Joanna W. (2022) got a similar performance as this work in his work as random forest outperformed the used neural network and SVM algorithms. This study agrees with what was applied by Joanna W.(2022), as the random forest model replicated its superiority when compared to other models developed in Martian weather exploration.

In regards to Martian weather patterns prediction, Ishaani and Puri (2021) focused on predicting hourly pressure and air temperature, Ali M (2021) predicted pressure, and Alejandro (2019) predicted hourly pressure and air temperature similar to what was done by Ishaan. The authors mentioned above used the REMS dataset, which could be more extensive in data size and attributes. We took a new approach using a more comprehensive dataset from the EMM EMIRS instrument. Not just the dataset, this work produced a different paradigm of mars weather pattern prediction as we focused on predicting the daily variation in atmospheric temperature on mars, which was quite different from the experiments they conducted in mars weather exploration.

This study disagrees on the strength of the neural networks in terms of performing predictions in Martian weather patterns, as several research conducted in the past identified neural networks as the best methodology for mars weather prediction (Ishaani, P. and Puri V,2021; Alejandro, 2019; Christopher Lee, 2019). This disagreement was seen as both ANN and CNN performed exceptionally poorly on the used dataset, contrary to the high performance obtained by each of the authors mentioned above. This disagreement was sponsored mainly by the change in the dataset used as this work used a new dataset different from what most of this author used in the past. Hence, in machine learning, modelling always remains an area that needs to be improved regularly as data change can affect a model's absolute reliability and performance level.

Also, following visualizations conducted, the results agreed with the point given by Michael D. et al., 2022 who pointed out that there is a variation in the daily Martian temperature. These observations can be seen in figures 19a - 19h, which confirmed what Michael D. et al. stated in 2022.

CHAPTER 5

CHALLENGES, CONCLUSION AND FUTURE WORK

5.1 Challenges

5.1.1 Missing values within the dataset

Model development relies heavily on quality data. Unfortunately, the used EMM EMIRS dataset has a lot of missing values present within it. These missing values are highly sponsored by many uncertainties present within the dataset, as identified by Michael D. et al., 2022. These missing values posed many problems as much prepossessing needed to be performed on the dataset before proper model development could be conducted. Furthermore, these missing values led to many assumptions and the dropping of entries within the dataset, which would have improved the model's overall accuracy.

5.1.2 Time and data size

The EMIRS dataset is large and takes much time to be imported into the Jupiter notebook environment. This made the entire analysis process take a long time to complete as the massive side of data takes about 5 minutes to upload. Due to this, it consumes many resources in the system being used. Also, due to size, the time taken for hyperparameter fine-tuning to be completed on the model took a very long time. For each model, the hyperparameter fine-tuning took about 30 minutes due to the size of the dataset used.

5.2 Conclusion

In this research, the presented questions were fully answered following the exploration and experiments conducted in this paper. Also, the objectives outlined were met. Firstly, the diurnal variation in mars' atmospheric temperature was explored. The visualizations confirmed the variation between temperatures during the day and at night on mars, per what was said by Michael D. et al., 2022. Secondly, explorations were conducted on the four temperature types present within the dataset. From the visualization, it was seen that a linear relationship exists among all four mars temperatures recorded by the EMIRS instrument. Subsequently, various preprocessing was conducted on the dataset to get it ready for model development which was one of the sole aims of this paper. Various preprocessing steps were conducted, such as feature engineering, removal and renaming of missing values, feature selection and data splitting. Two different experiments were conducted on model development. The first set of models was developed without removing outliers, which produced high error values. The second experiment was conducted by removing outliers present within the selected features. The second experiment produced higher accuracies compared to the first experiment. The second experiment was further subjected to hyperparameter fine-tuning for which optimal performance for each model was identified. Among these models, random forest outperformed other models with scores as RMSE=1.5525, MSE=2.4105, MAE=1.1939 and R2=0.8283.

5.3 Future work

Despite the results obtained in this work, modelling in machine learning remains a limitation, as making decisions based on the performance of a few models can only be generalised as a basis for some models. Hence this work is limited in terms of modelling. In the future, models such as RNN, LSTM, CCNN can be used on the dataset as these models were commonly used in the past for mars weather prediction although on a different dataset. On this dataset, these models can be developed and used so as to compare the performance of this models with the ones used in this work.

Also, the experiment in this work was based on one data split, which makes it limited as other splits may have produced better accuracies than the one obtained in this work. In future work, K-fold cross-validation techniques can be used to split the entire dataset into folds e.g. 5 or 10 on which models can be built on. This can help in producing better accuracies compared to the splitting technique used in this work. The latest release by EMM on February 9, 2023, showed that the space body had updated its database to include mars weather data up to August 2022. In future work, the dataset used in this work can be enlarged by incorporating the latest released data, which will provide more data that can be used for model development. To perform this, the added dataset which can possibly come in fits file format can be extracted using programming skills e.g. python programming after which same knowledge can be used to merge the extracted dataset to the existing set used in this work. Finally, more rigorous hyperparameter fine tuning can be applied upon model development to improve and obtain optimal results compared to the once obtained in this report. The hyper parameter tuning performed is limited due to resource and time constraint. The entire machine learning process in the future should be conducted in time as this will give room to better improve the performance of models by tuning many hyper parameters as hyper parameter tuning requires much time perform better.

RFERENCES

Abhishek Sheshnath (2022). 'Importance of Data Exploration in Data Analysis A Review Paper'. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)* 2(2). pp. 1-5. DOI: 10.48175/568 362

Akshita C. (2020). 'MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared — Which Metric is Better?'. Available at: https://medium.com/analytics-vidhya/ (Accessed: 15 January, 2023)

Ali Momennasab (November 2021). 'Machine Learning for Mars Exploration'. pp. 1-11.

Alejandro de Cabo Garcia, Alfonso Degado-Bonai, Maria Bele Perez Lancho, Jorge Garcia and German Martinez. (May 27,2019). 'MARSchine Learning Prediction: prediction of mars Metrological variables using Artificial Neural Networks'. pp. 1-2.

Arizona University (2022). "mars atmosphere". Available at: https://marsed.asu.edu/mep/atmosphere. (Accessed: 11 December, 2022)

Arizona University (2022). "Wind". Available at: https://marsed.asu.edu/mep/wind (Accessed: 5 February, 2023)

Ayat H., Sahar A., Basim A. and Mohammed N. (2023). 'Impact of feature reduction techniques on classification accuracy of machine learning techniques in leg rehabilitation'. *Measurement: Sensors,* 25, 100544. Available at: https://doi.org/10.1016/j.measen.2022.100544

Christopher Lee (16 April 2019). 'Automated crater detection on mars using deep learning'. pp. 2-9.

Charalambous, A. E. Stott, W. T. Pike, J. B. McClean, T. Warren, A. Spiga, D. Banfield, R. F. Garcia, J. Clinton, S. Stähler, S. Navarro, P. Lognonné, J.-R. Scholz, T. Kawamura, M. van Driel, M. Böse, S. Ceylan, A. Khan, A. Horleston, G. Orhand-Mainsant, L. M. Sotomayor, N. Murdoch, D. Giardini, B. Banerdt (2021). 'A Comodulation Analysis of Atmospheric Energy Injection Into the Ground Motion at InSight, Mars'. *JGRPlanets* 126(4). Available at: https://doi.org/10.1029/2020JE006538.

Chang K. (2013). 'Mars Could Have Supported Life Long Ago, NASA Says'. Available at: https://www.nytimes.com/2013/03/13/science/space/mars-could-have-supported.(Accessed: 21 January, 2023).

Charles Q., Mike W., Elizabeth H., Daisy Dobrijevic(2022).'Mars: Everything you need to know about the Red Planet'. Available at: https://www.space.com/47-mars-the-red (Accessed: 23 December, 2022).

Cloud Architecture Center (2022). 'Data preprocessing for ML: options and recommendations'. Available at: https://cloud.google.com/architecture/data-preprocessing-for-ml-with-tf- transform (Accessed: 11 January, 2023).

Crisp J., Grotzinger J. P., Vasavada A. R., Anderson R. C., Baker C. J., Barry R., BlakeD. F., Conrad P., Edgett K. S., Ferdowski B., Gellert R., Gilbert J. B., Golombek M., Gómez-Elvira J., Hassler D. M., Jandura L., Litvak M., Mahaffy P., Maki J. and Wiens R. C. (2012). 'Mars Science Laboratory Mission and Science Investigation'. *Space Science Reviews* 170(4). pp.5–56. Available at: https://doi.org/10.1007/s11214-012-9892-2.

Daniele Bellutta (2017). 'The Fog War: A Machine Learning Approach to forecasting weather on Mars'. pp. 2-5.

Desmos (2022). 'Why am I seeing a negative R^2 value?'. Available at: https://help.desmos.com/hc/en-us/articles/202529139-Why-am-I-seeing-a-negative-R-2. (Accessed: 1 February, 2023).

Dimitra A., Nour A., Dattaraj B. and Mathilde S. (2023). 'Diurnal variation of the surface temperature of Mars with the Emirates Mars Mission: a comparison with Curiosity and Perseverance rover measurements'. *Monthly Notices of* *the Royal Astronomical Society: Letters* 518 (1).pp. L1–L6. Available at: https://doi.org/10.1093/mnrasl/slac094

Doran G., Francis R., Estlin T., Johnstone S., Gaines D., Verma V., Burl M., Frydenvang J., Montaño S., Wiens R.C.(2017). ‘AEGIS Autonomous Targeting for ChemCam on Mars Science Laboratory: Deployment and Results of Initial Science Team Use’. *Sci. Robot.*, 2, eaan4582.

Dundar M, Ehlmann B. L., and Leask E. (2019). “Machine-Learning-Driven New Geologic Discoveries at Mars Rover Land- ing Sites: Jezero Crater and NE Syrtis.” *Wiley*. doi: 10.1002/es- soar.10501294.1.

Edwards C., Christensen P., Mehall G. et al. (2021). ‘The Emirates Mars Mission (EMM) Emirates Mars InfraRed Spectrometer (EMIRS) Instrument’. *Space Sci Rev* 217 (77). Available at: https://doi.org/10.1007/s11214-021-00848-1

EMM (2020). 'EMM Over view'. Available at: https://www.emiratesmarsmission.ae/mission/about-emm/(Accessed:5 January, 2023).

Fan C., Chen M., Wang X., Wang J., and Huang, B. (2021). 'A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data'. *Frontiers in Energy Research*, 9. Available at: https://doi.org/10.3389/fenrg.2021.652801

Fendall M., Denkins P., Fisher S. and Oluseyi H. (2014). “Human mars Exploration Research Objectives”. pp. 1-30

Frost J. (2017). 'Overfitting Regression Models: Problems, Detection, and Avoidance'. Available at: https://statisticsbyjim.com/regression/overfitting-regression (Accessed: 11 January, 2023).

Gera P.(2022). 'What are Hyper Parameter Tuning Techniques in Machine Learning?'. Available at: https://codedamn.com/news/machine-learning/what-are-hyper-parameter-tuning-techniques-in-machine-learning (Accessed:2 January, 2023).

REFERENCES

Gramigna E. (2020). ‘Calibration techniques for studying Venus and Mars atmospheres’. *Aerotecnica Missili & Spazio*, 99(4), pp.255–261.

Ishaani P. and Puri V. (2021). ‘Mars weather data analysis using machine learning techniques’. *Earth Science Informatics* 14. pp. 1885–1898. Available at: https://doi.org/10.1007/s12145-021-00643-0

Issam El Naqa and Ruijiang Li and Murphy M. J. (Eds) (2015). ‘Machine Learning in radiation oncology’. pp. 1-4.

Joanna Wendel (2022). 'Machine learning algorithms help scientists explore Mars'. Available at: https://phys.org/news/2022-01-machine-algorithms-scientists-explore-mars.html (Accessed: 2 January, 2023).

Joaquin Rojas (2022).’ Mars Weather’. Available at: https://public.tableau.com/app/profile/joaquin.rojas/viz/mars_16588821428780/Dashboar1 (Accessed: 26 September, 2022).

Joseph R. (2018). 'Grid search for model tuning'. Available at: https://towardsdatascience.com/grid-search-for-model-tuning (Accessed:12 January, 2023).

Kass D., Heavens N., Kleinbohl A. and Schofield J. (2020). “A multiannual record of gravity wave activity in Mars’s lower atmosphere from on-planet observations by the Mars Climate Sounder”. *Icarus 341:113630*.

Kumar R., Sharma, R., Sharma, D., Priyadarshini, I., Pham, B., Bui, D. T., & Rai, S. (2019).‘Inferring air pollution from air quality index by different geographical areas’. pp. 1347–1357.

Kristin R., Olivier G., Candice C., Erwin D., Jeremie L., David G., Travis G., Sylvestre M. and Roger C. (2021). “Clustering Supported Classification of ChemCam Data From Gale Crater, Mars”. Earth and Space science 8(12). pp. 1-10.https://doi.org/10.1029/2021EA001903

REFERENCES

Laura Dattaro (2013). 'Do mars Have seasons?'. Available at: https://weather.com/science/news/does-mars-have-seasons-20130910 (Accessed: 12 September,2022).

Michael D., Badri, K., Atwood, S.A., Edwards, C. S., Christensen, P. R., Wolff, M. J., et al. (2022). 'EMIRS observations of the aphelion-season Mars atmosphere'. *Geophysical Research Letters*, 49, e2022GL099636. Available at: https://doi. org/10.1029/2022GL099636.

Mohammad N., Mohammad A., Moses K. and Mehrdad K. (2020). 'Water on mars- A literature review'.*Galaxies*, 8(2), pp.1-40. Available at: https://doi.org/10.3390/galaxies8020040.

NASA (2021). "Mars". Available at: https://www.nasa.gov/mission_pages/mer/index.html (Accessed:1 January, 2023)

National Weather Service. (2022). 'The Planet Mars'. Available at: https://www.weather.gov/fsd/mars#:%7E:text=Temperatures%20on%20Mars%20average%20about,lower%20latitudes%20in%20the%20summer (Accessed: 26 September, 2022).

Olsen KS ,Korablev O, Trokhimovskiy A, Lefevre F, Montmessin,FFedorova A, Toplis M, Alday J, Belyaev D, Patrakeev A, Ignatiev N, Shakun A, Grigoriev A, Baggio L, Abdenour (2021). 'NASA Mars Exploration'. Available at:https://mars.nasa.gov/msl/mission/overview/(Accessed:27 October, 2022).

Priyadarshini I, Puri V., Cotton C. (2021). 'A novel LSTM–CNN–grid search based deep neural network for sentiment analysis'. *J Supercomput*. pp. 1-22.

Qiu D. , Rothrock B.,Tanvir Islam , Didier A., Vivian Z. Sun , Mattmann C. and Ono M. (2021). 'SCOTI: Science Captioning of Terrain Images for data prioritization and local image search'. pp. 1-9. Available at: https://doi.org/10.1016/j.pss.2020.104943.

Rahul A. (2022). ' Covolutional Neural Network (CNN)'. Available at: https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network. (Accessed: 3 January, 2023).

Rong, Shen Bao-wen and Zhang. (2018). 'The research of regression model in machine learning field'. *MATEC Web of Conferences*. 176. 01033. 10.1051/matecconf/201817601033.

Royal Belgian Institute for Space Aeronomy (2022). 'Mars climate, important temperature difference between day and night'. Available at: https://www.aeronomie.be/en/encyclopedia/mars-climate (Accessed: 5 January, 2023.).

Timothy Paul N., Louis S. and Nicholas Erickson (2022). 'Squeezing Data from a Rock: Machine Learning for Martian Science'. pp. 2-14. DOI:10.3390/geosciences12060248.

Xiao M., Wang D., Wu M., Liu K., Xiong H., Fellow, IEEE, Zhou Y., and Fu Y. (2022). 'Self-Optimizing Feature Transformation'. pp. 1-13. Available at: https://arxiv.org/pdf/2209.08044.pdf#:~:text=Abstract

APPENDICES

APPENDIX C (CONFERENCE STYLE)

Predicting variation in daily Martian Atmospheric temperature using machine learning techniques

Gabriel Tosin Ayodele

Faculty of Engineering and informatics - University of Bradford, UK

gtayodel@bradford.ac.uk

ABSTRACT

Mars exploration over time has been a field of interest to researchers. This work utilised the strength of machine learning to develop a model which can be used to predict the variation in daily Martian atmospheric temperature. Seven models were used from which Random Forest outperformed all the other six 6 used models.

Keywords: *Mars, temperature, EMM, EMIRS, machine learning.*

1. INTRODUCTION

Mars is believed to be a planet that has supported life (Chang K., 2013) due to its similarities with Earth. Out of curiosity, for mars exploration, humans have used rovers and spacecrafts. Despite success in landing of rovers in mars and obtaining information about mars weather conditions, the dynamic nature of its weather patterns remains an area that needs to be explored for better understanding on how it can possibly support life. Today, machine learning techniques have provided means for exploring mars from earth. Various machine learning techniques have been used by many authors to explore the Martian weather pattern in the past e.g., Ishaani and Puri (2021) used Neural networks to explore and develop models for predicting the daily martian temperature and pressure. Despite is usage, modeling still remains a challenge in this field as different datasets and models are used by different authors.

This study aims at developing a machine learning model for predicting the daily variation in atmospheric temperatures on Mars. The model will be trained on data collected from EMM EMIRS instrument and its accuracy will be evaluated using a suitable evaluation metric. The results of this study will provide valuable insight into the Martian climate, and contribute to the advancement of our understanding of weather patterns on the red planet. Machine learning skills have been used to explore the possibility of mars supporting life.

The following objectives are to be met at the end of this research paper:

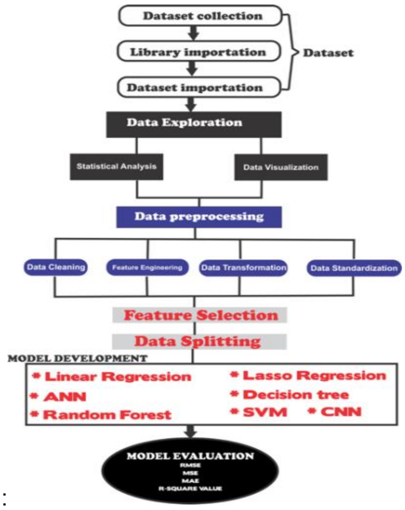
- **Identifying if there is a variation in the daily atmospheric temperature of planet mars.**
- **Detecting if a correlation exists among various temperatures in planet mars**
- **Develop a machine learning model for prediction of daily variations in mars atmospheric temperature.**
- **Contributing to the body of research on mars weather pattern exploration.**

2. DATASET

The used dataset in this work was obtained from the EMM EMIRS instrument database. The dataset was originally in fit files which needed to be converted for proper analysis. The extracted dataset consists of over 70,000 instances and over 50 attributes. This attribute contains records of daily Martian data recorded by the instrument.

3. METHODS

This paper deployed various machine learning procedures on the EMM EMIRS dataset. Python programming language alongside libraries such as NumPy, pandas, matplotlib, seaborn and sklearn will be used. Firstly, a python script was written to extract the original dataset which was in fits file. The figure below shows the experimental setup for this research:



Exploration, preprocessing and model development were performed on the dataset. For exploration, pandas provided several functions for exploring statistical information about a dataset. Data exploration are techniques used to see hidden patterns within a dataset. It involves both statistical exploration and visualization. Pandas, seaborn and matplotlib were used for exploring the EMM EMIRS datasett. Data pre-processing are techniques used to get data ready for model development. Inappropriate data pre-processing can affect the overall performance of a machine learning model. It involves techniques such as data cleaning, data transformation, removal of outliers, normalization and feature selection. This work used 7 machine learning models. These models are linear regression, random forest, support vector machine, decision tree, Artificial neural network, convolutional neural network and lasso regression. These models were used to conduct two experiments on the dataset: In the first experiment, outliers were included while in the second experiment, outliers were removed. Hyper parameter dine tuning involves exploring various parameter of a model for optimal performance. This can be done using gridsesrchcv technique.

Evaluation of model are done to examine the overall performance of a model. In this paper four evaluation techniques were used. This techniques are RMSE, MAE, MSE and R-squared value.

RESULT

Following evaluation of the developed models, the following results were obtained:

Model	**MAE**	**MSE**	**RMSE**	**R_2 score**
Random forest	1.1939	2.4105	1.5525	0.8283
Linear Regression	1.5275	3.5819	1.8926	0.7449
Support Vector Machine (SVM)	1.5038	3.6944	1.9221	0.7369
Decision Tree	1.5898	5.0146	2.2393	0.6429
LASSO regression	1.5271	3.5809	1.8923	0.7450

Table 1: Result with outliers

Model	**MAE**	**MSE**	**RMSE**	**R_2 score**
Random forest	1.3179	4.1967	2.0486	0.9448
Linear Regression	3.4310	50.0191	7.0724	0.3423
Support Vector Machine (SVM)	3.3729	45.1238	6.7174	0.4067
Artificial Neural Network (ANN)	149.4879	22704.6600	150.6807	-297.5380
Decision Tree	1.4242	3.6380	1.9073	0.9521
LASSO regression	3.0438	18.8760	4.3447	0.7518
Convolutional neural Network (CNN)	166.4401	27810.7507	166.7656	-364.6769

Table 2: Result without outliers after hyper parameter fine tuning

CONCLUSION

The study showed the strength of machine learning in performing exploration of Martian weather patterns. This worked deployed various machine learning techniques to develop models which can be used for predicting the variation in daily Martian atmospheric temperature. Following evaluation, the study showed that random forest performed excellently with score of 83% while neural networks performed poorly due to their complex nature on the dataset.

APPENDIX D

With outliers present within the dataset

```
x=df2.drop(['temp','longitude','utc','date','sc_altitude'],axis=1).values
y=df2['temp'].values
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.2,random_state=0)
scaler=StandardScaler()
scaler.fit(x_train)
x_train=scaler.transform(x_train)
x_test=scaler.transform(x_test)
```

```
rf=RandomForestRegressor()
rf.fit(x_train,y_train)
rf_pred=rf.predict(x_test)
print('Mean Absolute Error(Random Forest Algorithm:)',mean_absolute_error(y_test,rf_pred))
print('Mean Squared Error(Random Forest Algorithm:)',mean_squared_error(y_test,rf_pred))
print('Root Mean squared Error(Random Forest Algorithm:)',np.sqrt(mean_squared_error(y_test,rf_pred)))
print('R squared value(Random Forest Algorithm:)',r2_score(y_test,rf_pred))
```

```
Mean Absolute Error(Random Forest Algorithm:) 1.3179233448483598
Mean Squared Error(Random Forest Algorithm:) 4.196720225363468
Root Mean squared Error(Random Forest Algorithm:) 2.0485898138386482
R squared value(Random Forest Algorithm:) 0.9448183253270575
```

```
lr=LinearRegression()
lr.fit(x_train,y_train)
lr_pred=lr.predict(x_test)
print('Mean Absolute Error(Linear regrssion Algorithm:)',mean_absolute_error(y_test,lr_pred))
print('Mean Squared Error(Linear regression Algorithm:)',mean_squared_error(y_test,lr_pred))
print('Root Mean squared Error(Linear regression Algorithm:)',np.sqrt(mean_squared_error(y_test,lr_pred)))
print('R squared value(Linear regression Algorithm:)',r2_score(y_test,lr_pred))
```

```
Mean Absolute Error(Linear regrssion Algorithm:) 3.4309794632760977
Mean Squared Error(Linear regression Algorithm:) 50.019086422792164
Root Mean squared Error(Linear regression Algorithm:) 7.072417296992038
R squared value(Linear regression Algorithm:) 0.34231094611953383
```

```
svm=SVR()
svm.fit(x_train,y_train)
svm_pred=svm.predict(x_test)
print('Mean Absolute Error(Support vector machine Algorithm:)',mean_absolute_error(y_test,svm_pred))
print('Mean Squared Error(Support vector machine Algorithm:)',mean_squared_error(y_test,svm_pred))
print('Root Mean squared Error(support vector machine Algorithm:)',np.sqrt(mean_squared_error(y_test,svm_pred)))
print('R squared value(support vector machine Algorithm:)',r2_score(y_test,svm_pred))
```

```
Mean Absolute Error(Support vector machine Algorithm:) 3.3728954701981033
Mean Squared Error(Support vector machine Algorithm:) 45.12381290121316
Root Mean squared Error(support vector machine Algorithm:) 6.717426062206651
R squared value(support vector machine Algorithm:) 0.4066777317037331
```

```
ann=MLPRegressor()
ann.fit(x_train,y_train)
ann_pred=ann.predict(x_test)
print('Mean Absolute Error(Neural Network Algorithm:)',mean_absolute_error(y_test,ann_pred))
print('Mean Squared Error(Artificial Neural network Algorithm:)',mean_squared_error(y_test,ann_pred))
print('Root Mean squared Error(Artificial Neural network Algorithm:)',np.sqrt(mean_squared_error(y_test,ann_pred)))
print('R squared value(Artificial neural network Algorithm:)',r2_score(y_test,ann_pred))
```

```
Mean Absolute Error(Neural Network Algorithm:) 149.48794421009333
Mean Squared Error(Artificial Neural network Algorithm:) 22704.66292079703
Root Mean squared Error(Artificial Neural network Algorithm:) 150.68066538477
R squared value(Artificial neural network Algorithm:) -297.5382049730834
```

```
dt=DecisionTreeRegressor()
dt.fit(x_train,y_train)
dt_pred=dt.predict(x_test)
print('Mean Absolute Error(Decision Tree Algorithm:)',mean_absolute_error(y_test,dt_pred))
print('Mean Squared Error(Decision Tree Algorithm:)',mean_squared_error(y_test,dt_pred))
print('Root Mean squared Error(Decision Tree Algorithm:)',np.sqrt(mean_squared_error(y_test,dt_pred)))
print('R squared value(Decision Tree Algorithm:)',r2_score(y_test,dt_pred))
```

```
Mean Absolute Error(Decision Tree Algorithm:) 1.4242142621641816
Mean Squared Error(Decision Tree Algorithm:) 3.638006323469933
Root Mean squared Error(Decision Tree Algorithm:) 1.9073558460523126
R squared value(Decision Tree Algorithm:) 0.9521647213491723
```

```
lasso=Lasso()
lasso.fit(x_train,y_train)
lasso_pred=lasso.predict(x_test)
print('Mean Absolute Error(LASSO Algorithm:)',mean_absolute_error(y_test,lasso_pred))
print('Mean Squared Error(LASSO Algorithm:)',mean_squared_error(y_test,lasso_pred))
print('Root Mean squared Error(LASSO Algorithm:)',np.sqrt(mean_squared_error(y_test,lasso_pred)))
print('R squared value(LASSO Algorithm:)',r2_score(y_test,lasso_pred))
```

```
Mean Absolute Error(LASSO Algorithm:) 3.0438396811604314
Mean Squared Error(LASSO Algorithm:) 18.875992007074757
Root Mean squared Error(LASSO Algorithm:) 4.344650964930872
R squared value(LASSO Algorithm:) 0.7518040769626828
```



```
cnn= Sequential()
cnn.add(Dense(1, input_shape=(x.shape[1],)))
cnn.compile(optimizer='adam', loss='mean_absolute_error')
cnn.fit(x_train, y_train, epochs=100)
test_loss = cnn.evaluate(x_test, y_test)
print('Test loss:', test_loss)
cnn_pred = cnn.predict(x_test)
print('Mean Absolute Error(Deep Neural Network Algorithm:)',mean_absolute_error(y_test,cnn_pred))
print('Mean Squared Error(Deep Neural Network Algorithm:)',mean_squared_error(y_test,cnn_pred))
print('Root Mean squared Error(Deep Neural Network Algorithm:)',np.sqrt(mean_squared_error(y_test,cnn_pred)))
print('R squared value(Deep Neural Network Algorithm:)',r2_score(y_test,cnn_pred))
```

```
Epoch 1/100
4/4 [==============================] - 1s 3ms/step - loss: 167.5870
Epoch 2/100
4/4 [==============================] - 0s 3ms/step - loss: 167.5827
Epoch 3/100
4/4 [==============================] - 0s 5ms/step - loss: 167.5787
Epoch 4/100
4/4 [==============================] - 0s 3ms/step - loss: 167.5746
Epoch 5/100
4/4 [==============================] - 0s 4ms/step - loss: 167.5704
Epoch 6/100
4/4 [==============================] - 0s 3ms/step - loss: 167.5663
Epoch 7/100
4/4 [==============================] - 0s 3ms/step - loss: 167.5627
Epoch 8/100
4/4 [==============================] - 0s 3ms/step - loss: 167.5584
Epoch 9/100
4/4 [==============================] - 0s 3ms/step - loss: 167.5546
Epoch 10/100
```



```
Epoch 95/100
4/4 [==============================] - 0s 3ms/step - loss: 167.2105
Epoch 96/100
4/4 [==============================] - 0s 4ms/step - loss: 167.2065
Epoch 97/100
4/4 [==============================] - 0s 9ms/step - loss: 167.2026
Epoch 98/100
4/4 [==============================] - 0s 7ms/step - loss: 167.1985
Epoch 99/100
4/4 [==============================] - 0s 3ms/step - loss: 167.1946
Epoch 100/100
4/4 [==============================] - 0s 6ms/step - loss: 167.1906
1/1 [==============================] - 0s 165ms/step - loss: 166.4409
Test loss: 166.44091796875
1/1 [==============================] - 0s 227ms/step
Mean Absolute Error(Deep Neural Network Algorithm:) 166.4409104828113
Mean Squared Error(Deep Neural Network Algorithm:) 27810.750705978597
Root Mean squared Error(Deep Neural Network Algorithm:) 166.7655561139008
R squared value(Deep Neural Network Algorithm:) -364.6769370978755
```

Removal of outliers

```
df=df.drop(['utc','date','sc_altitude'],axis=1)
for x in df.columns:
    ll=df[x].quantile(0.05)
    ul=df[x].quantile(0.95)
    df=df[(df[x]>ll) & (df[x]<ul)]
df
```

	temp	tsurfco2	chi2temp	pres0	taudust	tauice	tsurfa	dustuncert	iceuncert	chi2a	tsurfh2o	wateruncert	chi2b	ql
1969	173.181961	203.959053	1.130275	5.264308	0.132028	0.263397	211.241576	0.066098	0.026650	1.133337	204.402553	69.362288	0.923580	58.64040
1975	179.551950	231.333021	1.013017	8.366886	0.167956	0.021300	236.725851	0.013838	0.004736	1.252246	234.627197	8.846959	2.447804	58.64049
1976	179.596473	231.244367	1.183926	8.282880	0.202582	0.139316	240.040805	0.017882	0.007328	1.178034	234.462433	16.231775	1.426785	58.64049
1977	176.120255	223.971354	0.875661	8.146422	0.247366	0.148567	230.677074	0.030279	0.010896	1.075459	226.798221	20.384648	0.982278	58.64051
1978	176.619119	213.105551	0.868895	7.680759	0.276279	0.320620	222.631449	0.055760	0.018711	1.325267	214.712115	59.960356	1.031755	58.64052
...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
9514	173.958676	247.303851	1.624857	6.715196	0.094746	0.153648	254.987697	0.012174	0.006373	1.500428	251.864379	5.762347	1.413258	85.36700
9515	170.872873	252.243122	1.765495	5.043581	0.053746	0.137971	258.331549	0.011815	0.006212	1.342205	256.847146	8.335163	1.761268	85.36702
9518	169.796290	251.570464	1.854313	5.408786	0.061513	0.086543	256.539118	0.013133	0.006627	1.276224	255.480869	8.416418	1.710361	85.36706
9519	170.209612	240.640785	1.598543	6.083259	0.022149	0.080918	244.183825	0.017054	0.008557	1.016201	244.069640	4.586082	1.836639	85.36708
9520	169.852193	223.746449	1.591085	6.695223	0.109540	0.136099	230.178658	0.024910	0.012163	1.010459	226.789103	8.195413	1.270890	85.36710

1011 rows × 16 columns

Feature Selection

```
x=df.drop(['temp','longitude'],axis=1).values
y=df['temp'].values
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.2,random_state=0)
scaler=StandardScaler()
scaler.fit(x_train)
x_train=scaler.transform(x_train)
x_test=scaler.transform(x_test)
```

```
rf=RandomForestRegressor()
rf.fit(x_train,y_train)
rf_pred=rf.predict(x_test)
print('Mean Absolute Error(Random Forest Algorithm:)',mean_absolute_error(y_test,rf_pred))
print('Mean Squared Error(Random Forest Algorithm:)',mean_squared_error(y_test,rf_pred))
print('Root Mean squared Error(Random Forest Algorithm:)',np.sqrt(mean_squared_error(y_test,rf_pred)))
print('R squared value(Random Forest Algorithm:)',r2_score(y_test,rf_pred))
```

```
Mean Absolute Error(Random Forest Algorithm:) 1.188753911401282
Mean Squared Error(Random Forest Algorithm:) 2.4228435299398896
Root Mean squared Error(Random Forest Algorithm:) 1.5565485954315366
R squared value(Random Forest Algorithm:) 0.827490167290017
```

```
lr=LinearRegression()
lr.fit(x_train,y_train)
lr_pred=lr.predict(x_test)
print('Mean Absolute Error(Linear regrssion Algorithm:)',mean_absolute_error(y_test,lr_pred))
print('Mean Squared Error(Linear regression Algorithm:)',mean_squared_error(y_test,lr_pred))
print('Root Mean squared Error(Linear regression Algorithm:)',np.sqrt(mean_squared_error(y_test,lr_pred)))
print('R squared value(Linear regression Algorithm:)',r2_score(y_test,lr_pred))
```

```
Mean Absolute Error(Linear regrssion Algorithm:) 1.5275958675005579
Mean Squared Error(Linear regression Algorithm:) 3.581958265793698
Root Mean squared Error(Linear regression Algorithm:) 1.8926062099110046
R squared value(Linear regression Algorithm:) 0.7449595842363199
```

```
svm=SVR()
svm.fit(x_train,y_train)
svm_pred=svm.predict(x_test)
print('Mean Absolute Error(Support vector machine Algorithm:)',mean_absolute_error(y_test,svm_pred))
print('Mean Squared Error(Support vector machine Algorithm:)',mean_squared_error(y_test,svm_pred))
print('Root Mean squared Error(support vector machine Algorithm:)',np.sqrt(mean_squared_error(y_test,svm_pred)))
print('R squared value(support vector machine Algorithm:)',r2_score(y_test,svm_pred))
```

```
Mean Absolute Error(Support vector machine Algorithm:) 1.3887219773036368
Mean Squared Error(Support vector machine Algorithm:) 3.98140196452779
Root Mean squared Error(support vector machine Algorithm:) 1.995345074048043
R squared value(support vector machine Algorithm:) 0.7165186367322174
```

APPENDICES

```
ann=MLPRegressor()
ann.fit(x_train,y_train)
ann_pred=ann.predict(x_test)
print('Mean Absolute Error(Neural Network Algorithm:)',mean_absolute_error(y_test,ann_pred))
print('Mean Squared Error(Artificial Neural network Algorithm:)',mean_squared_error(y_test,ann_pred))
print('Root Mean squared Error(Artificial Neural network Algorithm:)',np.sqrt(mean_squared_error(y_test,ann_pred)))
print('R squared value(Artificial neural network Algorithm:)',r2_score(y_test,ann_pred))
```

```
Mean Absolute Error(Neural Network Algorithm:) 31.659981067702947
Mean Squared Error(Artificial Neural network Algorithm:) 1530.9090437442437
Root Mean squared Error(Artificial Neural network Algorithm:) 39.12683278447469
R squared value(Artificial neural network Algorithm:) -108.00285543287711
```

```
dt=DecisionTreeRegressor()
dt.fit(x_train,y_train)
dt_pred=dt.predict(x_test)
print('Mean Absolute Error(Decision Tree Algorithm:)',mean_absolute_error(y_test,dt_pred))
print('Mean Squared Error(Decision Tree Algorithm:)',mean_squared_error(y_test,dt_pred))
print('Root Mean squared Error(Decision Tree Algorithm:)',np.sqrt(mean_squared_error(y_test,dt_pred)))
print('R squared value(Decision Tree Algorithm:)',r2_score(y_test,dt_pred))
```

```
Mean Absolute Error(Decision Tree Algorithm:) 1.5838708697259685
Mean Squared Error(Decision Tree Algorithm:) 5.103957179648427
Root Mean squared Error(Decision Tree Algorithm:) 2.259193922541495
R squared value(Decision Tree Algorithm:) 0.6365911424573458
```

```
lasso=Lasso()
lasso.fit(x_train,y_train)
lasso_pred=lasso.predict(x_test)
print('Mean Absolute Error(LASSO Algorithm:)',mean_absolute_error(y_test,lasso_pred))
print('Mean Squared Error(LASSO Algorithm:)',mean_squared_error(y_test,lasso_pred))
print('Root Mean squared Error(LASSO Algorithm:)',np.sqrt(mean_squared_error(y_test,lasso_pred)))
print('R squared value(LASSO Algorithm:)',r2_score(y_test,lasso_pred))
```

```
Mean Absolute Error(LASSO Algorithm:) 2.046925263362781
Mean Squared Error(LASSO Algorithm:) 7.2165658447789545
Root Mean squared Error(LASSO Algorithm:) 2.6863666623860105
R squared value(LASSO Algorithm:) 0.48617046406860587
```

```
cnn= Sequential()
cnn.add(Dense(1, input_shape=(x.shape[1],)))
cnn.compile(optimizer='adam', loss='mean_absolute_error')
cnn.fit(x_train, y_train, epochs=100)
test_loss = cnn.evaluate(x_test, y_test)
print('Test loss:', test_loss)
cnn_pred = cnn.predict(x_test)
print('Mean Absolute Error(Deep Neural Network Algorithm:)',mean_absolute_error(y_test,cnn_pred))
print('Mean Squared Error(Deep Neural Network Algorithm:)',mean_squared_error(y_test,cnn_pred))
print('Root Mean squared Error(Deep Neural Network Algorithm:)',np.sqrt(mean_squared_error(y_test,cnn_pred)))
print('R squared value(Deep Neural Network Algorithm:)',r2_score(y_test,cnn_pred))
```

```
Epoch 95/100
26/26 [==============================] - 0s 2ms/step - loss: 169.2860
Epoch 96/100
26/26 [==============================] - 0s 2ms/step - loss: 169.2597
Epoch 97/100
26/26 [==============================] - 0s 2ms/step - loss: 169.2339
Epoch 98/100
26/26 [==============================] - 0s 2ms/step - loss: 169.2082
Epoch 99/100
26/26 [==============================] - 0s 2ms/step - loss: 169.1818
Epoch 100/100
26/26 [==============================] - 0s 2ms/step - loss: 169.1557
7/7 [==============================] - 0s 2ms/step - loss: 169.4396
Test loss: 169.4396209716797
7/7 [==============================] - 0s 2ms/step
Mean Absolute Error(Deep Neural Network Algorithm:) 169.43961503652602
Mean Squared Error(Deep Neural Network Algorithm:) 28725.982046549856
Root Mean squared Error(Deep Neural Network Algorithm:) 169.48740969921587
R squared value(Deep Neural Network Algorithm:) -2044.3299175300995
```