Weather Prediction on Mars as a Multivariate Time Series Forecasting Problem

Sagar Uprety[1,2], Amel Bennaceur[1], Carlos Gavidia-Calderon[1,3], James A. Holmes[1], Manish R. Patel[1] and Kylash Rajendran[1]

[1] *The Open University, UK*
[2] *University College London, UK*
[3] *The Alan Turing Institute, UK*

Abstract

Accurate weather prediction on Mars is imperative for the safety of future human explorers and maximising the scientific return from robotic missions. Conventional physics-based numerical weather prediction models face challenges due to sparse observational data and the intricate Martian atmosphere. In this paper, we propose to use Machine Learning (ML) to forecasts Martian weather using the OpenMARS dataset and compare it to a physics-based numerical weather prediction model. OpenMARS is a reanalysis dataset that merges spacecraft observations and a Mars Global Circulation Model covering over a decade of Mars years, including three large dust storm events. We employ multiple ML models for time series forecasting to evaluate their performance against the OpenMARS dataset. The dataset includes variables such as surface pressure, temperature, near-surface winds, dust column, and water vapour, with a temporal resolution of two hours local time. Focusing on a 1-dimensional time series at a specific landing site location, resembling conditions for human exploration, we systematically train and test various ML models. Multiple ML models are efficient in the prediction of the dynamical variables up to one day in the future, with the TCN and TiDE models particularly effective at reproducing realistic intrinsic variability, but predicting the onset of a dust storm event remains challenging. Our findings contribute insights into Martian weather prediction, emphasising the potential and limitations of current ML-based approaches for timely decision making in future Martian missions. A replication package, including the OpenMARS dataset and the benchmarking results are publicly accessible at https://github.com/amelBennaceur/OpenMarsML. We hope that this work encourages collaborative efforts and advancements in ML for Martian weather research.

Keywords

Time Series Forecasting, Weather Forecasting, Martian Weather, Machine Learning, Numerical Weather Prediction

1. Introduction

Mars is one of our neighbouring planets and has long interested scientists and the general public alike. With current technology and resources, Mars is increasingly approaching being accessible for human exploration, with both NASA and the European Space Agency (ESA) planning to send humans to Mars [1, 2]. Critical to the safety of the human explorers is the accurate forecasting of local weather. Accurate forecasting of local weather can also be beneficial for current landers and rover missions that largely rely on solar power, which can be drastically impacted by local dust storms that cover the solar panels and therefore reduce the power to the lander/rover.

While Mars is a novel frontier in terms of weather forecasting, weather forecasting on Earth is performed primarily through data assimilation methods to initialise high resolution Earth Numerical Weather Prediction (NWP) models [3, 4] and are capable of up to 7-15 day forecasts with good accuracy. The use of Machine Learning (ML) models to forecast weather on Earth is gaining traction [5, 6]. In particular, most of those ML models use time-series forecasting. However, weather data has however rich features and involve more complexity than other time series datasets like electricity consumption or supermarket sales. Specifically, weather data contains multiple seasonality in a single variable itself, e.g. temperature varies in a day, within a season and also different at different geolocations. Furthermore, existing techniques do not compare the prediction results of those ML models to an NWP model, rather

Workshop on AI-driven Data Engineering and Reusability for Earth and Space Sciences (DARES'25), co-located with the 28th European Conference on Artificial Intelligence (ECAI 2025), Bologna, Italy, October 25, 2025

they compare it to other ML models, with the performance calculated by comparing predictions with the actual data.

Much like for Earth, current forecasting techniques for Mars are dominated by physics-based numerical weather prediction (NWP) models [7, 8], though the complex nature of underlying equations and additional modelling limitations (such as the inability to freely simulate the dust cycle to acceptable accuracy) mean that an accurate weather forecast is difficult to obtain. As mentioned earlier, NWP models also require large computational and time expense. Taking inspiration from the success of ML models for earth weather forecasting, in this paper we investigate whether ML based forecasting system for Martian weather can be as effective as a complex NWP model for Martian weather in a fraction of the time, thereby allowing for real-time decisions to be made.

In this paper, we trained multiple machine learning models on the OpenMars dataset [9, 10], which combines past spacecraft observations with a Mars Global Circulation Model (GCM). We measure the ability of ML models to forecast key variables of Martian weather in comparison to the GCM, which is an NWP model. We choose five different time-series forecasting models - RNN (LSTM), TCN, Transformers, NBEATS, and TiDE [11, 12, 13, 14, 15] where all five of them have different architectures. The rationale is to investigate which architectural components are more effective in representing the Martian weather data time-series. The purpose of this paper is to provide a benchmark for future work on Martian weather forecasting and these varied models serve as an important baseline. We find that for forecasting the surface temperature, and pressure variables, the TCN and TiDE models perform best. For the important task of predicting dust storms, no model succeeds at forecasting the storms, they rather forecast abrupt increases in dust optical depth after the large scale dust event had initiated in reality. One reason being that presence of large scale dust storms in the training data is very infrequent (only 2-3 dust storms in the training data as seen in figure 1). The rest of the paper is structured as follows. Section 2 presents the background and existing approaches to weather forecasting. Section 3 describes the OpenMars dataset, including the 1-D subset we processed and extracted for our experiments. Section 4 details the experiments we conducted including the configurations of the ML models. Section 5 reviews and discusses the results of the different ML approaches Finally, Section 6 concludes the results and discusses future work.

2. Background

In this section, we starts by discussing methods for time series data, then we move on how they have been leveraged for weather forecasting. Finally, we discuss existing work on weather forecasting for Mars.

2.1. Time series forecasting methods

Statistical models, such as the Autoregressive Integrated Moving Average (ARIMA) family of models [16], offer foundational approaches for the analysis and forecasting of time series data. ARIMA models excel at capturing linear relationships by taking a weighted sum of past observations. They are widely used in time series forecasting due to their simplicity and interpretability, however they depend on stationarity of data and fail to capture non-linear patterns.

Apart from statistical models, deep learning (DL) models have also been used for time-series forecasting. Initial DL models adopted two types of standard DL architectures - Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). RNN models process input data sequentially and are effective in capturing the temporal dependencies of time series data. Different variations of RNNs have been successfully developed including DeepAR [17], deep state-space based models [18] - both of which output probability forecasts, and attention-based RNNs [19, 20, 21], which employ the attention mechanism to give more weights to certain parts of the input series in order to better capture long-range dependencies in the time-series.

The CNN architecture, while originally suited to capture local, short-term patterns has also been successfully applied in time-series modelling [22, 23, 24]. Researchers employ a modified CNN architecture called a Temporal Convolution Network (TCN) [24]. TCN enhances the power of convolutions by increasing the filter size and introducing dilated convolutions which increase the receptive field of CNNs, thus enabling them to look back much further in the length of sequence. Some models e.g. LSTNet [25] combine both CNN and RNN architectures wherein CNNs help model the short-term, local patterns and RNNs are used to model the longer term trends and patterns in the time-series data.

Transformers [13] are been a revolutionising neural network architecture that helped ML take a huge leap forward. It has become the default architecture in almost all the state-of-the-art models in all branches of ML and also forms the basis of the Large Language Models (LLMs) and the Generative AI revolution. Various authors have attempted to apply the transformer architecture to time-series modelling [26, 27]. One disadvantage of applying the original architecture directly to time-series is that the self-attention mechanism has quadratic complexity in sequence length N, thus taking a lot of compute and memory. Subsequent works like LogSparse [26], Reformer [28], Informer [29], Autoformer [30] have sought various ways to reduce the complexity to O(N(log N)). However, it is not only the computational complexity of the self-attention mechanism which makes the vanilla Transformer architecture sub-optimal for time-series data, the permutation-invariant nature of self-attention itself makes it harder to capture sequential correlations [31].

Some authors do not adopt the standard DL architectures discussed above in their time series models, e.g. the NBEATS model [14]. This model comprises stacks of fully connected blocks, wherein each block is a made up of number of neural network hidden layers. Each block attends to a certain part of the input time series, and all the downstream blocks combine to represent the whole of the input. This increases the model's depth, thereby enabling it to model complex sequences. Another unique architecture called TiDE (Time Series Dense Encoder) [15] aims to replace the self-attention of Transformers with encoders and decoders consisting of multi-layer perceptrons. This replaces the quadratically complex self-attention component, while retaining the capability of handling non-linearity in data. The authors provide theoretical analysis to prove that linear models can achieve optimal performance at par with non-linear models when the ground truth is generated from a linear dynamical system. The paper claims that it is 10x faster in training and 5x faster at inference than a transformer based model, with comparative performance across a range of time-series modelling tasks.

2.2. Weather forecasting

Weather forecasting for Earth has a long heritage, with more recent approaches making use of ML models. Existing work [32] uses a deep neural network model for a point-wise rain classification task. 7 variables including humidity, temperature, pressure, and rain were collected from a local weather station in Japan. Each of these variables were passed on to two fully connected layers, before being concatenated together in a larger layer. Finally, a softmax layer was used to project the large fully connected layer to obtain the binary rain classification outcome. The forecast horizon was very short-term, of one hour only. This DL based model is shown to perform better than traditional ML based methods such as XGBoost and support vector machines. In existing work [33], a modified version of LSTM, called transductive LSTM is used to predict temperature in 5 European cities. The dataset is collected from a website called Weather Underground [1]. The transductive LSTM model is shown to perform better than vanilla LSTMs for this dataset. [34] uses a local weather station dataset to show superior performance of TCNs over vanilla LSTM models.

One of the recent models forecasting at a global scale, utilising the reanalysis datasets such as ERA5 is Graphcast from Google Deepmind [35]. It has an encoder-decoder architecture with the encoders and decoders comprising of Graph Neural Network (GNN) layers. GNN layers in the encoder are used to model a high spatial resolution multi-mesh over the globe, while those in the decoder map the learned features from the multi-mesh representation back to the latitude-longitude grid. Graphcast is found to perform better than the best performing NWP model - ECMWF's High Resolution (HRES), for certain latitude/longitude resolution on the surface and for certain vertical atmospheric levels. The Fourcastnet

[1] https://www.wunderground.com/

model [36] is a complex architecture with several components like Vision Transformers (ViT) [37] for learning from satellite images and Fourier neural operators [38] for representing and learning partial differential equations. ViT helps the model look at higher resolution images than traditional CNN based models. Another recent model which is comparable to the above two on the ERA5 dataset is Pangu-weather [39]. It uses a 3D model with a variant of the ViT architecture to represent the input weather data. It produces lower Root Mean Square Error scores than both Fourcastnet and the IFS NWP model.

2.3. Weather Forecasting for Mars

Weather forecasting for Mars has a shorter history, and forecasting of atmospheric phenomena is partly driven through observations by landers and satellites. The InSight lander is one of such system, with instruments that are capable of gathering data from its landing site location that can also be used to interpret larger scale patterns in weather [40]. The authors suggest that the observations from the InSight lander will be important for advancing predictive capabilities of relevance to future exploration. With respect to dust storms, InSight captured data of the dust optical depth during a regional scale dust storm and corresponding surface pressure and near-surface air temperature. Other studies focus on the prediction of specific Mars atmospheric phenomena. For example, recent papers have reviewed the state of the art of forecasting models for dust storms [41, 42]. They found the current approaches —based on domain knowledge and statistical analysis— are inadequate for accurate and timely forecast. Also, they believe forecasting of dust storms can be improved by incorporating real observation data into the atmospheric models, in a process called *data assimilation*. It has been shown that forecasts of carbon monoxide are much improved when the technique of data assimilation is used [7].

The only previous use of ML models to predict specifically martian weather is detailed in [43]. They used observations of maximum temperature by the NASA Curiosity rover collected over 5.5 Earth years and compared the prediction error of different ML models. They used a univariate approach in predicting a single variable and there are no conclusions drawn on which of the ML models performed best from a physics perspective.

3. Dataset

To train and test our machine learning models, we used the publicly available Open access to Mars Assimilated Remote Soundings (OpenMARS) dataset [9, 10]. The OpenMARS dataset is a reanalysis product combining past spacecraft observations with a Mars Global Circulation Model (GCM). It is a global surface/atmosphere reference database of surface and atmospheric properties from July 1998 to April 2019 (equivalent to around eleven Mars years). It can be used by scientists and engineers interested in global surface/atmospheric conditions and the physical, dynamical and chemical behaviour of the atmosphere for the recent past on Mars. The OpenMARS dataset includes spacecraft observations of temperature, dust and water vapour from the Thermal Emission Spectrometer instrument [44, 45, 46] on the NASA Mars Global Surveyor spacecraft. It also includes temperature and dust column optical depth from the Mars Climate Sounder instrument [47, 48] aboard NASA's Mars Reconnaissance Orbiter spacecraft. The observations are combined with the Mars GCM at the appropriate time and location using the Analysis Correction scheme [49] that performs successive corrections to relevant variables and weights the observation data over a short time window and spatial distance from each observation to prevent instabilities from rapid changes to the simulation output.

For further details about the OpenMARS dataset, we refer the interested reader to existing work [9]. The complete OpenMARS dataset is a 4-dimensional time series of multiple atmospheric and surface variables associated with the Mars system. For this study, the underlying scenario is of a human explorer at a given landing site location and the problem is to predict in real time the weather on Mars expected in the coming days. Therefore, we pull out a near-surface time series of multiple variables at one particular location and reduce the dataset to a 1-dimensional time series for this first study of a

ML-based forecasting system utilising the OpenMARS dataset. This reduced dataset can be publicly accessed and downloaded at https://github.com/amelBennaceur/OpenMarsML.

The variables from the OpenMARS dataset used as input for the training, validation and testing of an ML-based forecasting system are detailed in Table 1.

Table 1
Variables used from the OpenMARS dataset.

Variable	Description	Unit
Tsurf	Surface temperature	Kelvin
Psurf	Surface pressure	Pascals
Cloud	Water ice optical depth at infrared wavelength	No unit
Vapour	Water vapour column	kg/m^2
u_wind	Near-surface (~4 m) zonal wind	m/s
v_wind	Near-surface (~4 m) meridional wind	m/s
Dust	Dust column optical depth at visible wavelength	No unit
Temp	Atmospheric temperature at ~20 km altitude	Kelvin

Table 2
First few rows of the cleaned dataset

Time	Tsurf	Psurf	Cloud	Vapour	u_wind	v_wind	Dust	Temp
1998-07-15 21:23:39	264.042	721.113	0.092	0.027	-7.451	8.604	0.428	179.686
1998-07-15 23:26:53	274.736	705.090	0.145	0.026	-7.053	4.934	0.427	174.502
1998-07-16 01:30:07	265.939	700.691	0.105	0.026	-6.825	-0.063	0.427	173.429
1998-07-16 03:33:21	238.624	697.252	0.134	0.025	-5.373	-4.048	0.426	173.556
1998-07-16 05:36:35	213.634	717.146	0.139	0.026	-3.899	-3.133	0.426	174.789

Table 2 lists a small sample of the dataset showing all the above variables. The temporal resolution of the dataset is every two hours and in total there are 88,560 data points for each variable. The chosen gridpoint from the complete OpenMARS spatial grid is the 5° gridbox (approximately 300 km^2 on the surface of Mars) centred at 2.5°N, 135°E which corresponds to the model gridbox over the InSIGHT landing site location [40]. The dataset covers the timespan from the start of Mars Year 24 to just after the end of Mars Year 34, equivalently from 15th July 1998 to 23rd April 2019.

4. Experiment

The ML models were trained on nine Mars years of OpenMARS data, which results in 88560 data points or timesteps, with step corresponding to exactly two hours local time on Mars (equivalent to $2 : 03 : 14$ on Earth). Since our primary aim in these experiments is to investigate the forecasting of dust storms, Figure 1 shows the column dust optical depth values for the full dataset. As one can see that the planet-encircling dust events occur either earlier in the dataset or in the last couple of years in the dataset. We therefore decide to keep a larger validation set to include as many smaller local scale dust storms as possible. Otherwise, the optimised hyperparameters learned for the models will not be as effective in predicting the dust storms in the test set. Thus the split ratios are 70, 20, 10.

4.1. Tools

We use all free and open-source tools to perform our experiments, so that they can be easily reproduced. We utilise the Darts [50] python library for training the models. This library provides architecture implementations of various time series models. All models were implemented and trained using the Darts library. We also utilise Optuna [51] library for tuning hyperparameters. The MLflow[2] library is used for experiment tracking - logging parameters, plots, models and other artifacts.

[2] https://mlflow.org/docs/latest/index.html

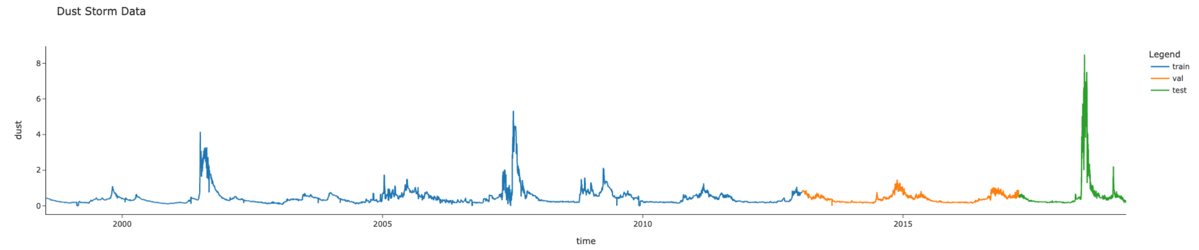

Figure 1: Full dataset for dust time series with different splits.

4.2. Training details

Model training is accompanied by early stopping and hyperparameter tuning. For early stopping, we monitor the validation loss with a patience of 3 (which is standard practice) and minimum delta of 0.00008 (empirically, we found it to strike a good balance between stopping very early and running many epochs with very little change in loss). Some hyperparameters are common across all the models and some are model specific. Two important hyperparameters for all the models are 1) ***input_chunk_length*** - Also called Window or lookback window, this parameter defines the size of the input sequence or the historical context that the model considers for making predictions. A larger input chunk length allows the model to capture longer-term dependencies in the time series data, while a smaller window emphasises more recent patterns. Very large window sizes often make it difficult for models to capture all possible dependencies over the window length. Too short a size would mean that the models are not able to capture much dependencies and correlations. It is imperative to have an ideal window size. One could always consider it as a hyperparameter and tune the models to find the window size which gives best results over a set of metrics. However, for the purpose of this paper, we fix the input chunk length of all the models as it gives us a common ground to compare the models. We use a input chunk size of 84 timesteps, which is equivalent to one week in Mars. Too large a value of input chunk size is likely to add additional longer-term seasonal changes that would reduce the accuracy of the forecast, while a week on Mars should be able to capture short-term trends relevant for a daily forecasting system. 2) ***output_chunk_length*** - also called horizon, it is the number of time steps in the future predicted in one forward pass of the model. For a model to predict a time series much greater than ***output_chunk_length***, it needs to autoregressively call itself multiple times with current output becoming the input to the next time step. It needs to be noted that ***output_chunk_length*** cannot be greater than ***input_chunk_length***. For the purpose of this paper, we fix the output chunk length to be 12 timesteps, which is one day at Mars. Empirically, we found the models struggle to predict well for longer horizons and horizon lesser than one day seems to be too small for forecasting purposes. Martian missions would be better off with knowledge of Martian weather at least a day in advance.

Besides these, common hyperparameters include *batch_size*, *dropout*, and *learning_rate*. Then there are model specific hyperparameters which are discussed in the next section. We initially performed hyperparameter tuning of all parameters other than input and output chunk lengths, over the validation set. However, as one can see from figure 1, the validation time series in the middle doesn't contain any major dust storm. There are bigger dust storms in the training set and the test only. Hence, the tuned hyperparameters on the validation set did not lead to improved performance on the test set. We therefore fixed all the hyperparameters to standard/default values as used in the Darts library. The hyperparameters used in training models are listed in table 3.

4.3. Metrics

We use standard metrics in evaluating the performance of the above models for the OpenMARS dataset - Mean Average Precision (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE). Note that the metrics are evaluated on the validation set while performing the hyperparameter

Table 3
Best hyperparameter values

Model	Hyperparameters
RNNModel	n_rnn_layers=2, hidden_dim=30, batch_size=96, dropout=0.25, learning_rate=0.0005
TCNModel	kernel_size=2, num_filters=6, dilation_base=2, batch_size=96, dropout=0.05, learning_rate=0.0005
Transformers	d_model=12, n_head=6, num_encoder_layers=2, num_decoder_layers=4, dim_feedforward=64, batch_size=96, dropout=0.05, learning_rate=0.0005
NBEATS	num_blocks=3, num_layers=4, layer_widths=512, batch_size=96, dropout=0.05, learning_rate=0.000953
TiDe	num_encoder_layers=2, num_decoder_layers=2, decoder_output_dim=1, temporal_decoder_hidden=1, batch_size=96, dropout=0.05, learning_rate=0.0005, hidden_size=30

optimisation and on the test set for reporting model performance and comparison.

5. Results and Discussion

The results are broken down into the forecasting of dynamical variables contained within the dataset, followed by the forecasting of the dust optical depth and in particular the initiation of the large scale dust storm event that occurred during the test dataset.

5.1. Dynamical weather forecasting

This section focuses on the daily forecasting of the surface temperature, surface pressure and zonal wind. The error metrics for the different variables and different ML models averaged across the entire test dataset are shown in Table 4. Figure 2 shows the daily forecasting of the variables restricted to the global scale dust storm event and also zoomed in on a five day time period to see more clearly the variations in the forecast between each ML model. Dynamical variables are perturbed from their ambient state during a global scale dust event, with the increased dust loading in the atmosphere shielding the near surface and generally reducing the surface temperature variability across a diurnal cycle. This can be seen in the actual values in Figure 2.

Table 4
Metrics for multiple variables from the forecast models averaged across the whole test dataset. The lowest value for each error metric for each variable has bold emphasis.

Model	Variable	MAE	MAPE	RMSE
RNN	Tsurf	0.0247	11.07	0.0341
	Psurf	0.0275	6.02	0.0407
	u_wind	0.0646	16.67	0.0901
TCN	Tsurf	**0.00602**	**2.54**	**0.0109**
	Psurf	0.0130	2.67	0.0172
	u_wind	0.0612	15.25	0.0764
Transformer	Tsurf	0.0281	14.25	0.0362
	Psurf	0.0279	6.01	0.0408
	u_wind	0.0605	15.42	0.0861
NBEATS	Tsurf	0.0188	8.71	0.0258
	Psurf	0.0321	6.16	0.0404
	u_wind	**0.0534**	**13.76**	**0.0751**
TiDE	Tsurf	0.00699	2.99	0.0126
	Psurf	**0.0118**	**2.48**	**0.0160**
	u_wind	0.0665	16.14	0.0781

In regards to forecasting the surface temperature and pressure, the TCN and TiDE models are the best performers with a clear gap in error metrics between these two ML models and the others (Table 4). Not only do they outperform the other ML models, but their forecasts are also much more realistic from a physics perspective. The TCN and TiDE models are much more accurate in their daily forecast of maximum and minimum surface temperature in both the long-term (Figure 2a) and short-term (Figure 2b) time window. The RNN, NBEATS and Transformer ML models all consistently over-predict the maximum and minimum daily surface temperature and also have a steeper gradient in temperature when transitioning across the day-night terminator (Figure 2b). The Transformer ML model even forecasts a small perturbation just before the minimum surface temperature is reached each day which has no counterpart in reality (Figure 2b).

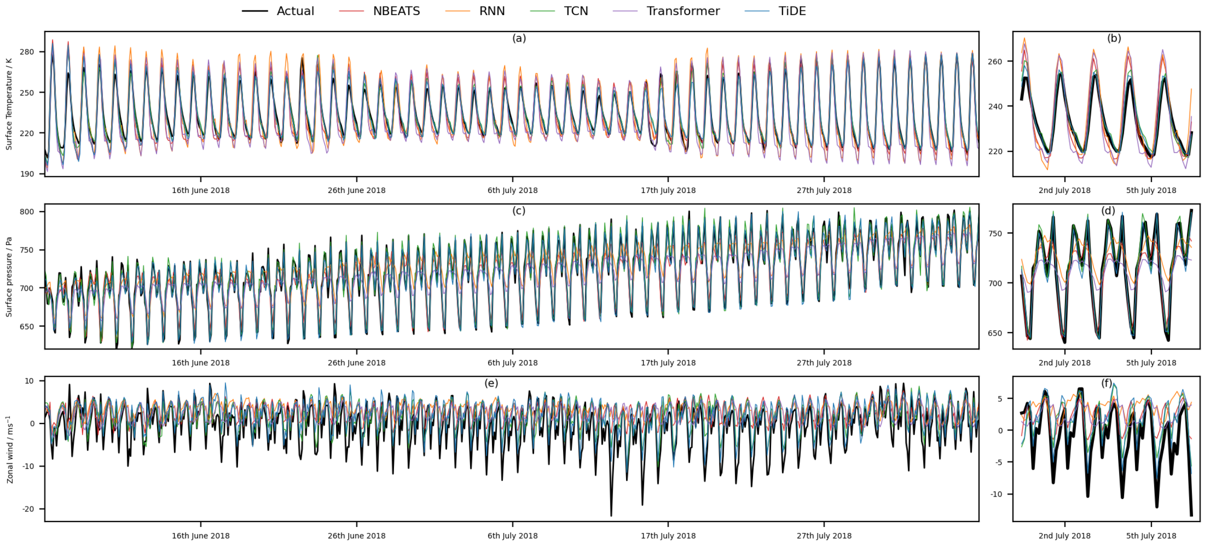

Figure 2: Daily forecasts of the surface temperature (a-b), surface pressure (c-d) and zonal wind (e-f) by all models in comparison to the actual values (black) during and after the large dust event (a,c,e) and zoomed in on a selected 5 sol window (b,d,f).

From a physics perspective, the TCN and TiDE model also capture the realism of the surface pressure cycle (Figure 2c,d). The RNN, NBEATS and Transformer ML models all display a dampened pressure cycle in long-term (Figure 2c) and short-term trends (Figure 2d). The TCN and TiDE ML models however forecast the surface pressure daily fluctuations much more accurately, rarely deviating from the actual variations and therefore also capturing the semi-diurnal and diurnal tide (i.e. the double peak and trough structure on a given day) in surface pressure (Figure 2d). The other ML models all forecast either a much weaker diurnal and semi-diurnal tide in the case of NBEATS or do not even manage to capture the signature of the semi-diurnal tide at all (RNN and Transformer).

Regarding the forecasting of zonal winds, which across the analysed time period shows increased intrinsic variability (Figure 2e) when compared to surface temperature (Figure 2a) and pressure (Figure 2c), the gap in error metrics between all ML models is much reduced (Table 4). In terms of error metrics alone, the NBEATS ML model is the best performer. When interpreting the daily forecasts on a short-term time window in Figure 2f the NBEATS daily forecast does not capture well the peak magnitude of westward (negative) winds each day, consistently forecasting westward winds of lower magnitude than the actual values (this is true for the RNN and Transformer model that rarely forecast westward/negative winds apart from at the start of the long-term time period investigated as seen in Figure 2e). The TCN and TiDE model, from interpretation of the short-term trend alone (Figure 2f) appear to track the peak magnitudes of eastward and westward winds to better accuracy than the other ML models. This seems to not be the case for the longer-term time period analysis in which the TiDE model in particular is regularly seen to forecast peak eastward winds above the actual values (Figure 2e).

5.2. Forecasting large scale dust events

An outstanding capability even for highly complex Mars GCMs is the forecasting of an impending large scale dust event that will have significant impact on human explorers in terms of a reduction in solar energy to vehicles/equipment and increased risk of inhalation of fine particles. While trends in the dynamical weather variables analysed in Section 5.1 are seasonal and shift slowly over time, changes in dust optical depth can be abrupt as seen when large scale dust events occur throughout the dataset (Figure 1). Figure 3a displays the daily forecast for each model in the second half of the test dataset

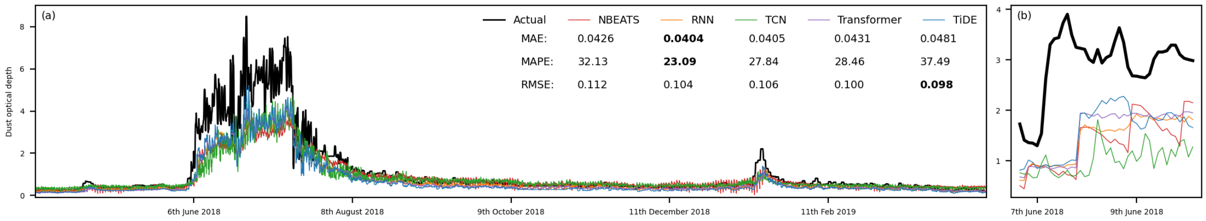

Figure 3: Daily forecasts of the dust column optical depth by all models in comparison to the actual values (blue) during the large dust event and regional dust storm (a) and in zoomed in to the start of the large dust event (b). The error metrics for each model are included in the figure, with bold emphasis on the lowest value for each error metric.

(from June 2018 and beyond, when the dust season begins) that includes the large dust event and the regional dust storm later in time. All ML models have reasonable success in reconstructing the dust optical depth during quiet periods where actual values are below 1, with the TCN and NBEATS ML models by eye tracking the actual values marginally better than the other ML models.

All the ML models also show increased dust optical depth during the global scale dust event and regional scale event in mid-January 2019, with a greater spread in the ability of each model to capture the precise timing and magnitude of each event. Figure 3b shows a zoom-in of forecasting the dust optical depth during the initiation of the global dust event, and identifies contrasting behaviours between the different ML models. The TCN model increases the dust optical depth above 1 at exactly the same time as the actual values abruptly increase, although the increase in the TCN forecast is short-lived and drops below 1 around 4 hours later in contrast to the actual values. The TCN model then does not elevate dust levels until 6 hours later than all the other ML models, which show a sharp and abrupt rise in dust optical depth around 16 hours after the actual large scale dust event has initiated (Figure 3b). While the initiation of the large scale dust event is at best only seen 16 hours later in time in the ML forecasts, all ML models forecast the decay of the large dust event reasonably well, with a decrease in dust optical depth inline with the actual values (Figure 3a).

The metrics for daily forecasting of the dust optical depth for each different ML model are also shown in Figure 3a. While the RNN model has the lowest value for two of the three metrics, the spread in variability across all ML models is small enough to be largely insignificant. Although the metrics are largely similar, there are clear differences in how the daily forecast evolves between different ML models, showing the dual approach of analysing metrics and forecast trajectories to be beneficial.

The main challenge for all ML models, perhaps unsurprisingly as it is also extremely difficult using more complex weather models, is in forecasting the onset of a dust storm event before the knowledge of an actual dust storm event is available to guide the forecast (and is therefore redundant as an early warning system). The results from the TCN model are encouraging as it does forecast an abrupt increase in dust optical depth at the same time as the actual large scale dust event initiated, but it also decayed rapidly which is in contrast to what actually happened and further analysis is warranted to see if this increase is coincidence and not above the general noise level of the dust forecast throughout the rest of the test dataset. Hope for predicting a global dust event before it fully initiates was previously tantalisingly found in increased wind speeds approaching this exact global dust event utilising a data assimilation approach [52], but no evidence was found for a similar feature just before other observed global dust events in different years. It also was not a global-wide increase in wind speed and not evident at the specific spatial location chosen for the analysis conducted here.

6. Conclusion and Future Work

We have explored the forecasting of dynamical variables and dust storms in the OpenMARS dataset using a suite of ML models. Daily forecasting of the surface pressure/temperature by the TCN and TiDE ML models was extremely successful and could capture realistic tidal structures in surface pressure which were not captured by the other ML models. The forecasting of zonal winds was more tricky, with the metrics increasing across all ML models.

Forecasting the onset of a large scale dust event was not successful as all ML models forecast abrupt increases in dust optical depth after the large scale dust event had initiated in reality. This isn't entirely surprising, especially as in the training dataset large dust events that did occur happened at different times during a given Mars year and therefore the three large dust events in the whole dataset can be considered as unique events. From an ML perspective, there is lack of training data pertaining to these storm events. Some intriguing results were found with the TCN ML model increasing dust optical depth exactly when the large scale dust event happens (albeit only for a brief 6 hour time window) that warrant further exploration.

For shorter lived regional dust storm events and abrupt changes, a shorter input chunk length would possibly allow the ML models to converge to the increased values faster in time. Still, the goal to forecast dust events remains something to achieve and may not be possible using this dataset alone. Testing ML models on actual data rather than simulated data from a GCM would also be worthwhile although Mars is far less observed by satellites/landers and the data is relatively sparse currently which may present an issue for forecasting through an ML approach.

The results in this paper serve as a benchmark for further analysis and improvement by the wider community. Forecasting a dust storm before it happens remains a target since it will form a critical component of information for future human explorers to ensure their safety.

7. Data Availability

A replication package, including the OpenMARS dataset and the benchmarking results are publicly accessible at https://github.com/amelBennaceur/OpenMarsML.

Declaration on Generative AI

No generative AI tools have been used to produce any of the content of this paper.

References

[1] K. C. Laurini, W. H. Gerstenmaier, The global exploration roadmap and its significance for nasa, Space Policy 30 (2014). doi:10.1016/j.spacepol.2014.08.004.

[2] B. Hufenbach, T. Reiter, E. Sourgens, Esa strategic planning for space exploration, Space Policy 30 (2014) 174–177. doi:10.1016/j.spacepol.2014.07.009.

[3] N. E. Bowler, A. Arribas, K. R. Mylne, K. B. Robertson, S. E. Beare, The MOGREPS short-range ensemble prediction system, Quarterly Journal of the Royal Meteorological Society (2008).

[4] F. Molteni, T. Stockdale, M. Alonso-Balmaseda, G. Balsamo, R. Buizza, L. Ferranti, L. Magnusson, K. Mogensen, T. Palmer, F. Vitart, The new ECMWF seasonal forecast system (system 4), 2011.

[5] J. Pathak, S. Subramanian, et al., Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators, 2022. arXiv:2202.11214.

[6] Z. Ben-Bouallegue, M. C. A. Clare, L. Magnusson, et al., The rise of data-driven weather forecasting, 2023.

[7] J. A. Holmes, S. R. Lewis, M. R. Patel, M. D. Smith, Global analysis and forecasts of carbon monoxide on Mars, Icarus 328 (2019) 232–245. doi:10.1016/j.icarus.2019.03.016.

[8] C. E. Newman, M. de la Torre Juárez, J. Pla-García, R. J. Wilson, S. R. Lewis, et al., Multi-model Meteorological and Aeolian Predictions for Mars 2020 and the Jezero Crater Region, Space Sci. Rev. (2021).

[9] J. A. Holmes, S. R. Lewis, M. R. Patel, OpenMARS: A global record of martian weather from 1999 to 2015, Planet. Space. Sci. 188 (2020) 104962. doi:10.1016/j.pss.2020.104962.

[10] P. M. Streeter, S. R. Lewis, M. R. Patel, J. A. Holmes, K. Rajendran, An eight-year climatology of the martian northern polar vortex, Icarus 409 (2024) 115864. doi:10.1016/j.icarus.2023.115864.

[11] Z. C. Lipton, J. Berkowitz, C. Elkan, A critical review of recurrent neural networks for sequence learning, arXiv preprint arXiv:1506.00019 (2015).

[12] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, G. D. Hager, Temporal convolutional networks for action segmentation and detection, in: CVPR, 2017.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017.

[14] B. N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio, N-beats: Neural basis expansion analysis for interpretable time series forecasting, arXiv preprint arXiv:1905.10437 (2019).

[15] A. Das, W. Kong, A. Leach, S. Mathur, R. Sen, R. Yu, Long-term forecasting with tide: Time-series dense encoder, arXiv preprint arXiv:2304.08424 (2023).

[16] G. E. Box, G. M. Jenkins, Some recent advances in forecasting and control, Journal of the Royal Statistical Society. Series C (Applied Statistics) 17 (1968) 91–109.

[17] D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, Deepar: Probabilistic forecasting with autoregressive recurrent networks, International journal of forecasting 36 (2020) 1181–1191.

[18] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, T. Januschowski, Deep state space models for time series forecasting, Advances in neural information processing systems 31 (2018).

[19] S.-Y. Shih, F.-K. Sun, H.-y. Lee, Temporal pattern attention for multivariate time series forecasting, Machine Learning 108 (2019) 1421–1441.

[20] H. Song, D. Rajan, J. J. Thiagarajan, A. Spanias, Attend and diagnose: clinical time series analysis using attention models, in: Proc.AAAI Symposium on Educational Advances, AAAI Press, 2018.

[21] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, G. Cottrell, A dual-stage attention-based recurrent neural network for time series prediction, arXiv preprint arXiv:1704.02971 (2017).

[22] A. Borovykh, S. Bohte, C. W. Oosterlee, Conditional time series forecasting with convolutional neural networks, 2018. arXiv:1703.04691.

[23] R. Sen, H.-F. Yu, I. S. Dhillon, Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting, Advances in neural information processing systems (2019).

[24] S. Bai, J. Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, arXiv preprint arXiv:1803.01271 (2018).

[25] G. Lai, W.-C. Chang, Y. Yang, H. Liu, Modeling long- and short-term temporal patterns with deep neural networks, in: SIGIR, 2018.

[26] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, X. Yan, Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting, Advances in neural information processing systems 32 (2019).

[27] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, L. Sun, Transformers in time series: A survey, arXiv preprint arXiv:2202.07125 (2022).

[28] N. Kitaev, Ł. Kaiser, A. Levskaya, Reformer: The efficient transformer, arXiv preprint arXiv:2001.04451 (2020).

[29] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: AAAI, 2021.

[30] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, Advances in neural information processing systems (2021).

[31] A. Zeng, M. Chen, L. Zhang, Q. Xu, Are transformers effective for time series forecasting?, in: Proceedings of the AAAI conference on artificial intelligence, volume 37, 2023, pp. 11121–11128.

[32] K. Yonekura, H. Hattori, T. Suzuki, Short-term local weather forecast using dense weather station by deep neural network, in: 2018 IEEE International Conference on Big Data (Big Data), 2018.

[33] Z. Karevan, J. A. Suykens, Transductive lstm for time-series prediction: An application to weather forecasting, Neural Networks 125 (2020) 1–9.

[34] P. Hewage, A. Behera, M. Trovati, et al., Temporal convolutional neural (tcn) network for an effective weather forecasting using time-series data from the local weather station, Soft Computing (2020).

[35] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds,

Z. Eaton-Rosen, W. Hu, et al., Graphcast: Learning skillful medium-range global weather forecasting, arXiv preprint arXiv:2212.12794 (2022).

[36] T. Kurth, S. Subramanian, P. Harrington, et al, Fourcastnet: Accelerating global high-resolution weather forecasting using adaptive fourier neural operators, in: Proc. of the platform for advanced scientific computing conference, 2023.

[37] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, et al., An image is worth 16x16 words: Transformers for image recognition at scale, in: Inter. Conf. on Learning Representations, 2021.

[38] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, in: International Conference on Learning Representations, 2021.

[39] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, Q. Tian, Accurate medium-range global weather forecasting with 3d neural networks, Nature 619 (2023) 533–538.

[40] D. Banfield, A. Spiga, C. Newman, F. Forget, M. Lemmon, R. Lorenz, N. Murdoch, D. Viudez-Moreiras, J. Pla-Garcia, R. F. Garcia, et al., The atmosphere of mars as observed by insight, Nature Geoscience 13 (2020) 190–198.

[41] L. Montabone, F. Forget, On forecasting dust storms on mars, 48th International Conference on Environmental Systems, 2018.

[42] F. Forget, L. Montabone, Atmospheric dust on mars: A review, 47th International Conference on Environmental Systems, 2017.

[43] I. Priyadarshini, V. Puri, Mars weather data analysis using machine learning techniques, Earth Science Informatics 14 (2021) 1885–1898. doi:10.1007/s12145-021-00643-0.

[44] M. D. Smith, J. C. Pearl, B. J. Conrath, P. R. Christensen, Mars Global Surveyor Thermal Emission Spectrometer (TES) observations of dust opacity during aerobraking and science phasing, J. Geophys. Res. 105 (2000) 9539–9552. doi:10.1029/1999JE001097.

[45] M. D. Smith, The annual cycle of water vapor on Mars as observed by the Thermal Emission Spectrometer, Journal of Geophysical Research (Planets) 107 (2002) 5115.

[46] M. D. Smith, Interannual variability in TES atmospheric observations of Mars during 1999-2003, Icarus 167 (2004) 148–165. doi:10.1016/j.icarus.2003.09.010.

[47] A. Kleinböhl, A. J. Friedson, J. T. Schofield, Two-dimensional radiative transfer for the retrieval of limb emission measurements in the martian atmosphere, J. Quant. Spectrosc. Ra. (2017).

[48] A. Kleinböhl, A. Spiga, D. M. Kass, et al., Diurnal Variations of Dust During the 2018 Global Dust Storm Observed by the Mars Climate Sounder, J. Geophys. Res. (Planets) (2020).

[49] A. C. Lorenc, R. S. Bell, B. MacPherson, The Meteorological Office analysis correction data assimilation scheme, Q. J. R. Meteorol. Soc. 117 (1991) 59–89.

[50] J. Herzen, F. LÃ¤ssig, S. G. Piazzetta, et al., Darts: User-friendly modern machine learning for time series, Journal of Machine Learning Research (2022).

[51] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: KDD, 2019.

[52] K. Rajendran, S. R. Lewis, J. A. Holmes, et al., Enhanced super-rotation before and during the 2018 martian global dust storm, Geophysical Research Letters (2021).