Micro-climate Prediction - Multi Scale Encoder-decoder based Deep Learning Framework

Peeyush Kumar
peeyush.kumar@microsoft.com
Microsoft Research
Redmond, WA, USA

Ranveer Chandra
Microsoft Research
Redmond, WA, USA

Chetan Bansal
Microsoft Research
Redmond, WA, USA

Shivkumar Kalyanaraman
Microsoft
Bengaluru, India

Tanuja Ganu
Microsoft Research
Bengaluru, India

Michael Grant[*]
University of Washington Seattle
Seattle, WA, USA

ABSTRACT

This paper presents a deep learning approach for a versatile Micro-climate prediction framework (DeepMC). Micro climate predictions are of critical importance across various applications, such as Agriculture, Forestry, Energy, Search & Rescue, etc. To the best of our knowledge, there is no other single framework which can accurately predict various micro-climate entities using Internet of Things (IoT) data. We present a generic framework (DeepMC) which predicts various climatic parameters such as soil moisture, humidity, wind speed, radiation, temperature based on the requirement over a period of 12 hours - 120 hours with a varying resolution of 1 hour - 6 hours, respectively. This framework proposes the following new ideas: 1) Localization of weather forecast to IoT sensors by fusing weather station forecasts with the decomposition of IoT data at multiple scales and 2) A multi-scale encoder and two levels of attention mechanisms which learns a latent representation of the interaction between various resolutions of the IoT sensor data and weather station forecasts. We present multiple real-world agricultural and energy scenarios, and report results with uncertainty estimates from the live deployment of DeepMC, which demonstrate that DeepMC outperforms various baseline methods and reports 90%+ accuracy with tight error bounds.

CCS CONCEPTS

• **Applied computing** → **Environmental sciences**; • **Mathematics of computing** → *Multivariate statistics*; **Stochastic processes**; • **Theory of computation** → **Theory and algorithms for application domains**; • **Hardware** → Sensor applications and deployments; • **Computer systems organization** → **Neural networks**.

KEYWORDS

Micro-climate prediction; Sustainability; Deep learning; Encoder-decoder; Attention mechanism; Transfer learning; IoT sensors; Sequence to sequence; Time series; Agriculture; Energy; Wavelet transform

ACM Reference Format:
Peeyush Kumar, Ranveer Chandra, Chetan Bansal, Shivkumar Kalyanaraman, Tanuja Ganu, and Michael Grant. 2021. Micro-climate Prediction - Multi Scale Encoder-decoder based Deep Learning Framework. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447548.3467173

1 INTRODUCTION

It is the month of April and a farm in Eastern Washington, USA is producing wheat and lentil crops. The spring is just settling in while the temperature is slightly above freezing. The farmer is getting ready to fertilize his fields as the conditions become safe from winter runoff and frost [39]. The plants are significantly susceptible to fertilizers at freezing temperatures, therefore, the farmer consults the local weather station for temperature forecasts, which is located in the closest metropolitan valley about 50 miles away from the farm. The 3-day predictions show consistent temperatures above the freezing point. The farmer rents equipment and fertilizers and starts fertilizing the farm. A couple of nights the temperature in certain parts of the field drop below freezing and kills around 20% of the crops. Despite the availability of weather forecasts from commercial weather stations, this is a common situation that can affect up to 20% of the crops [27, 28, 39]. This is because the climatic parameters around the plant not only vary from the nearest weather stations but also between various regions of the farm.

In this paper, we present a micro-climate prediction framework (DeepMC), which, among other related problems, addresses the problem presented above. Micro-climate is the accumulation of climatic parameters formed around an (approximately) homogeneous and relatively smaller region [19, 30]. Knowledge of micro-climate and micro-climate predictions are of importance in agriculture [5, 31], forestry [33], architecture [12], urban design [1], ecology conservation [38], maritime [8] and many other domains. DeepMC predicts various micro-climate parameters with 90%+ accuracy at IoT sensor locations deployed in various regions across the world. For brevity and ease of explanation, this paper will focus on micro-climate prediction for agriculture and energy, and demonstrate

[*] Also with The Climate Corporation as current affiliation at the time of publication.

KDD '21, August 14–18, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467173

results computed in farm conditions for micro-climate parameters - soil moisture, wind speed, radiation, humidity, soil temperature, and ambient temperature. More results are available for different domains and micro-climate parameters using the same methodology, which will not be presented in this paper. Readers are encouraged to try the same architecture presented in this paper for similar problems with varying input-output climatic parameters.

Climatic parameters are stochastic in nature and quite challenging to model for prediction tasks. Some of the challenges with developing a framework for micro-climate predictions are:

1) Non-stationary features: Non-stationarity of the climatic time-series data makes it difficult to model the input-output relationship. Each input feature affects the output variable at a different temporal scale, for example, the effect of precipitation on soil moisture is instantaneous while the effect of temperature on soil moisture is accumulated over time. Additionally, climatic parameters behave differently during different times of the year, especially during the times when seasons are transitioning. To capture these varying effects, a solution needs to capture multiple trends in the data in a stationary way [13, 22, 29]. DeepMC utilizes a multi scale wavelet decomposition based approach to capture these effects [13, 22] (see Section 6.4). This approach decomposes the input signals into various scales capturing trends and details in the data.

2) Varying input dimension: Various factors influence the trend of a particular climatic parameter of interest. For example, soil moisture predictions are correlated with climatic parameters such as ambient temperature, humidity, precipitation, and soil temperature [17]; While ambient humidity is correlated with parameters - ambient temperature, wind speed, and precipitation [40]. This creates a challenge for a machine learning system to accept vectors of varying dimensions as input. Additionally, in many cases, the data for a single or a group of climatic parameters might be unavailable, which creates a requirement for a micro-climate prediction framework to work with varying input dimensions. DeepMC solves this problem by: First, decomposing the input into signals across various scales for each feature and combining them through a paired cartesian product; Then using a heuristic to apply specialized architecture for specific scales. This heuristic assigns the components of the architecture based on the nature of the paired scales rather than specific features, enabling adaptability for varying input dimensions. More details are provided in Section 6.4.

3) Result accuracy: Generating high accuracy results is an obvious challenge for any real-world deployment of a machine learning solution. In the context of micro-climate predictions, the challenges described above - small quantity of labeled datasets, heterogeneity of features, and non-stationary of input features make the learning problem itself quite difficult. In this work, instead of predicting the climatic parameter directly, we predict the error between the nearest commercial weather station forecast and local micro-climate forecast. This is based on the hypothesis that hyperlocalization of weather station forecasts is easier to learn than learning the relationships of the predicted climatic parameter with the predictor climatic parameters from the ground-up. DeepMC performs better than other commonly used models across multiple applications, sensors, and farms, where accuracy is quantified with multiple metrics of accuracy (please see Section 7 for more details).

To the best of our knowledge, there is no other comprehensive framework which can be consistently and accurately used for predicting various climatic parameters. We identify other related work in literature for micro-climate prediction in Section 2

2 RELATED WORK

Digital transformation in many traditional industries such as agriculture, forestry, supply chain & logistics, and renewable energy has increased the demand for accurate micro-climate predictions. There are traditional statistical and algebraic time-series forecasting mechanisms, such as ARIMA [3], matrix factorization models [4] and other statistical models such as genetic modeling, vector machines, etc., see [24]. These mechanisms are not reliable for micro-climate predictions because of the non-stationarity of data and sharp jumps in signal profiles.

There is another body of work which predicts micro-climate for a specific climatic parameter based on the area of interest. There are many physical models available to characterize the weather dynamics [23] but these models cannot be used *as-is* in practice due to various factors such as constraints on model usability, data availability, and application-specific characteristics [37]. Deep learning methods alleviate some of these challenges enabling learning relationships between various climatic parameters, provided there is enough high quality labelled datasets. They either use physical models to forecast micro-climate or specialized neural networks. For example, there has been work on rainfall predictions [13], soil moisture predictions [17], wind speed Predictions [11, 22], and micro-temperature predictions [31], among other work. Most physical models are only suitable for climate predictions over a broad spatial area, therefore they are not very useful for micro-climates. While, the specialized neural networks achieve good performance for the specific climatic parameter they are designed for, they lack generalizability across other micro-climate signals. *Vanilla* deep learning models, such as MLP, LSTM, CNNLSTM and CNNs, are insufficient for micro-climate predictions due to the inherent difficulty in capturing the complex dynamics and interplay of climatic parameters, especially during times when seasons transition. There has been work to resolve this issue by developing hybrid systems combining physical models with classical machine learning techniques, for example see Galanis et al. [11]. These models still lack generalizability and are harder to reproduce commercially. DeepMC addresses the above shortcomings of previous work through the contribution summarized in Section 3.

3 DEEPMC CONTRIBUTIONS SUMMARY

To summarize, the major contributions of this paper are:

(1) A versatile system for micro-climate prediction with real-world deployment: This work develops a micro-climate prediction framework which can be used for multiple input-output paired climatic parameters. We highlight four real-world deployments that characterize a diverse set of conditions. We conduct a comprehensive validation of DeepMC across various regions around the world and various micro-climatic parameters. The predictions computed are being used through real-world deployments of the FarmBeats system in Ireland, India, Australia, Kenya, and United States. The results presented here are computed for predictions

of local temperature, local wind speed, soil moisture, soil temperature, radiation, and humidity. The framework is generalizable to other input-output combinations of the climatic parameters.

(2) New technology and workflow: DeepMC proposes new ideas, which address the challenges of solving and deploying real-world micro-climate prediction systems; Specifically the following 4 parts address the problem of micro-climate prediction in a new way and enables DeepMC to generate compelling results: a) the design mechanism which enables DeepMC to be used across multiple input-output climatic parameters and to adapt to varying non-stationary signal characteristics, b) the application of multi-scale learning and a new mechanism for attention model for effective and accurate micro-climate predictions, and c) Hyperlocalization of weather station forecasts by predicting forecast error as an endogenous variable. Also, we identify the key requirements of engineering a real-world micro-climate prediction system and propose ways to fulfill those. The workflow presented here is consistent across various applications and can be used as a blueprint for other micro-climate prediction problems.

We observe high accuracy of predictions across multiple applications. This work resulted in (MAPE) accuracy scores above 90% for many real-world application tests on direct learning, with high Root-Mean-Squared-Error (RMSE) and Mean-Absolute-Error (MAE) scores as well. This is better than other models which address the problem of micro-climate predictions for specific climatic parameters.

4 ENGINEERING A REAL-WORLD MICRO-CLIMATE PREDICTION SYSTEM

For this research, we identified several characteristics for engineering a real-world micro-climate prediction system.

- High Prediction accuracy: Micro-climate prediction aids various decisions on the farm - both for agricultural farms as well as renewable energy farms. For instance in agriculture, among others, certain operational decisions such as seeding, irrigating, fertilizing, harvesting, etc. can have high economical and work-effort consequences. These decisions are sensitive to minor changes in weather forecasts and therefore, there is a general requirement for a high degree of accuracy in the predictions.
- Data collection and data delivery: To develop a micro-climate prediction system, we require real-time data from sensor locations at the farm. More often than not, applications which require micro-climate prediction have very low network coverage [25]. One objective is to get the farm data to cloud storage reliably and in real-time. Additionally, it is also required to present the prediction results through a medium which can be ingested and understood by the end-user in real-time.

DeepMC is designed to satisfy each of the above requirements. Section 5 presents how we satisfy the data requirements for both - ingesting data to predict micro-climate and showcasing predictions for the end-user to intake. Section 6 presents details on the architecture and how it solves many of the challenges described in this section above. Section 7 provides some real-world scenarios where DeepMC is being used and demonstrates the performance of DeepMC across various applications and regions around the world. Section 8 concludes this paper with the impact of this work on environmental sustainability and in broader industrial applications.

5 DATA REQUIREMENTS

DeepMC uses weather station forecasts and IoT sensor deployments as inputs through the FarmBeats platform to predict micro-climatic parameters in real-time.

Weather Station Forecasts: Weather station forecasts are collected for training and inference through commercial weather stations. The model is trained and tested with various weather data providers - DarkSky[1], NOAA[2], AgWeatherNet[3], National Weather Service[4] and DTN[5].

IoT Sensor Climate Data: DeepMC uses FarmBeats [34] platform to collect climatic and soil data from multiple sensors around the world. FarmBeats is an end-to-end IoT platform for data-driven agriculture, which provides consistent data collection from various sensor types with varying bandwidth constraints. We chose the FarmBeats system for this work because of high system reliability and system availability, especially during events such as power and Internet outages caused by bad weather - scenarios that are fairly common for a farm. This data collected by FarmBeats IoT sensors is persisted in the cloud and accessed there.

The actual parameters collected depend on the predicted variable of interest. The superset of climatic parameters for which we collect both the current as well as forecasted data includes Ambient Temperature, Ambient Pressure, Humidity, Soil Moisture, Soil Temperature, Radiation, Precipitation, Wind Speed, Wind Direction.

We use the FarmBeats platform dashboard to deliver micro-climate predictions to the end-users using their Azure marketplace offering[6].

6 APPROACH: MICRO-CLIMATE PREDICTION

The DeepMC learning framework is shown in Figure 1. The framework is based on a sequence to sequence [32] encoder-decoder framework. The encoder consists of 5 distinct parts: A) Pre-processor, B) Forecast Error Computer, C) Wavelet Packet Decomposition. D) Multi-scale deep learning, E) Attention Mechanism. The decoder is a multi-layer LSTM and fully connected layer. Each component of the framework is described in the following subsections with some implementation details for the sake of reproducibility.

6.1 Pre-processing of sensor data

Sensor data is received using IoT sensors deployed on the farm. The raw data which is received from the sensors is usually noisy with missing data and with varying temporal resolution. We standardize the temporal resolution using average values for the data collected.

[1] https://darksky.net/dev
[2] https://www.ncdc.noaa.gov/cdo-web/webservices/v2
[3] https://weather.wsu.edu/
[4] https://www.weather.gov/documentation/services-web-api
[5] https://cs-docs.dtn.com/apis/weather-api/
[6] https://azuremarketplace.microsoft.com/en-us/marketplace/apps/microsoftfarmbeats.microsoft_farmbeats

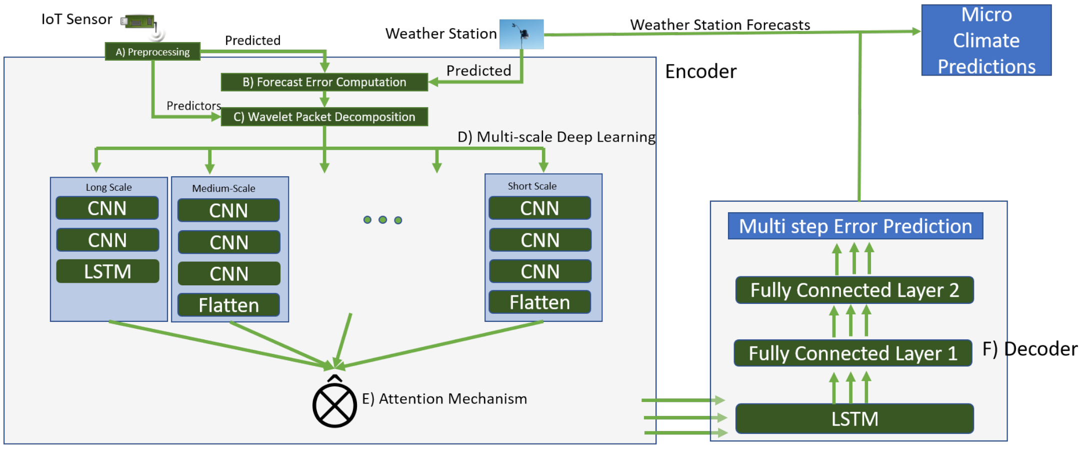

Figure 1: DeepMC architecture for the multi scale encode-decoder deep learning system. The architecture consists of 6 distinct parts- A) The preprocessor, B) Forecast error computation, C) Wavalet packet decomposition, D) Multi-scale deep learning, E) Attention Mechanism, F) Decoder. The attention mechanism is depicted in Figure 3.

We denote the weather data from the sensors as a tuple (z_k, y_k), where y is the climatic parameter to be predicted, z is the multi-variate predictors - the climatic data which affects the predicted parameter and k is the time epoch when the corresponding values were recorded. Also we denote the required temporal resolution to be Δ, then the values (z_t, y_t) are the averaged values within the time interval $[k, k+\Delta)$.

$$(z_t, y_t) = \left(\frac{\sum_k^{k+\Delta} z_k}{\Delta}, \frac{\sum_k^{k+\Delta} y_k}{\Delta} \right)$$

For preprocessing and filling in missing values there has been a lot of specialized work on weather sensor data [2, 15]. We use ARIMA forecasting model [3] to fill in missing data because of its general availability and ubiquitousness.

6.2 Forecast Error Computation

DeepMC uses weather station forecasts of the predicted variable to learn better models for micro-climate predictions. In this work, instead of predicting the climatic parameter directly, we predict the error between the nearest commercial weather station forecast and the local micro-climate forecast. This is based on the hypothesis that hyperlocalization of weather station forecasts is more efficient to learn than learning the relationships of the predicted climatic parameter y with the other parameters z and auto-relationship of the y with itself at earlier times. We denote the weather station forecasts for the predicted variable as $\bar{y}_{t+\ell}$, where $\ell \in [0, L]$ is the future interval from a given time t for which the forecast is recorded. For training purposes, we use historical weather forecasts and sensor data. Therefore, the corresponding recording of the sensor predicted data at time $t+\ell$ is $y_{t+\ell}$. Then the forecast error $u_{t+\ell}$ is

$$u_{t+\ell} = y_{t+\ell} - \bar{y}_{t+\ell}$$

DeepMC predicts $u_{t+\ell}$ using data recorded at and before time t for a retrospective horizon length of L'. The estimate of $u_{t+\ell}$ (denoted as $\hat{u}_{t+\ell}$) alongside the weather forecast, is used to obtain the prediction for the climatic parameter of interest,

$$\hat{y}_{t+\ell} = \bar{y}_{t+\ell} + \hat{u}_{t+\ell},$$

where $\hat{y}$ is the prediction of y.

One artifact of using the forecast error as the predictor signal is that it does not inherently capture the effect of distance of the weather station from the point of interest. For this purpose, we also include a Relative Latitude ($RLat$) and Relative Longitude ($Rlong$) as additional features.

$$RLat = Lat(Weather\ Station) - Lat(Micro-region),$$

$$RLong = Long(Weather\ Station) - Long(Micro-region).$$

Summarizing, the prediction problem takes in IoT sensor historical data $(z_{t-\ell'}, y_{t-\ell'})$ and weather station forecasts $\bar{y}_{t+\ell}$ to estimate $y_{t+\ell}$, where the estimate is denoted by $\hat{y}_{t+\ell}$, using an endogenous variable, the forecast error $u_{t+\ell}$ and its estimate $\hat{u}_{t+\ell}$, where $\ell \in (0, L]$ is the future time interval and $\ell' \in [0, L']$ is the retrospective time interval. In addition, the prediction problem also takes in the geo-coordinates of the weather station {Lat(Weather Station), Lat(Weather Station)} and the micro-region of interest {Lat(micro-region), Lat(micro-region)}. For convenience we will denote the historical data $(z_{t-\ell'}, y_{t-\ell'}, \{Rlat, Rlong\}) = x_{t-\ell'}$.

6.3 Wavelet Packet Decomposition

The Wavelet Packet Decomposition (WPD) is a classical signal processing method built on Wavelet Analysis. Wavelet analysis gives an efficient way to decompose time series from the time domain to scale domain. It localizes the change across time within different scales of the original signal [29]. In the last decade, there has been some research on using WPD for time series forecasting [13, 22, 29]. We use the same idea to decompose our input signals into a combination of various scales. The multiscale decomposition (WPD) uses low-pass and band-pass filters. Applying this pair of filters to a time series leads to a first-order series which contains the trend, the long scale dynamics, and a second one which contains the details (shorter scale dynamics). The original time series may be reconstructed by summing up the trend and the detail series. The details for WPD are provided in the next paragraph.

The wavelet packet decomposition is based on the wavelet analysis. In wavelet analysis, the wavelet transform decomposes the original signal into a mutually orthogonal set of wavelets. The discrete wavelet is defined as

$$\Psi_{j,k}(t) = \frac{1}{\sqrt{s_0^j}}\Psi\left(\frac{t - k\tau_0 s_0^j}{s_0^j}\right),$$

where j and k are integers, $s_0 > 1$ is a fixed dilation step and translation factor, τ_0 depends on the dilation step. For further information on interpretation of these terms please see Chun-Lin [6] The scaling function and wavelet function of the discrete wavelet transform are defined as:

$$\Phi(2^j t) = \sum_k h_j(k)\Phi(2^{j+1}t - k)$$

$$\Psi(2^j t) = \sum_k g_j(k)\Phi(2^{j+1}t - k),$$

where $g(k)$ and $h(k)$ are the corresponding discrete filters for the wavelet Ψ. Then the original signal $f(t)$ can be reconstructed as

$$f(t) = \sum_j \sum_k A_j(k)\Phi(2^j t - k) + \sum_j \sum_k D_j(k)\Psi(2^j t - k),$$

where A_j and D_j are approximation and detail coeffecients, respectively, of the Wavelet Packet Transform (WPT) at level j. Expanding and rearranging the outer summation terms, the coefficients of the time series $f(t) = \{f_\ell, \forall \ell = [1, T]\}$ can be cascaded as

$$\begin{aligned}\text{coefficients}(f_\ell) &:= \{A_1[\ell], D_1[\ell]\}\\ &= \{\{A_2^{A_1}[\ell], D_2^{A_1}[\ell]\}, \{A_2^{D_1}[\ell], D_2^{D_1}[\ell]\}\}\\ &= \vdots,\end{aligned}$$

where $D_n^*(t)$ and $A_n^*(t)$ are detail and approximation coefficients at level n, respectively. The wavelet Ψ used is Daubechies wavelet [6] with the corresponding scaling function Φ and filters h and g. A n level wavelet packet decomposition produces 2^n different sets of coefficients. Figure 2 shows the wind speed signal at various scales using wavelet packet decomposition for $N = 5$. We can see that decomposing signal using WPD gives signals with multiple levels of trends and details. In the context, of climatic data, this corresponds variations such as long term trends, yearly variation, seasonal variation, daily variations, etc. We denote the wavelet packet decomposition for the predictor (x) and error signal (u) as the sets $w^x = \{w_1^x, \cdots, w_N^x\}, w^u = \{w_1^u, \cdots, w_n^u\}$, respectively, where N is the level of decomposition and $w_n^x, (n \in [1, N])$ is the reconstructed signal for predictor variable x using N levels of decomposition. Please note that the predictor variable $x_{t-\ell'} = (z_{t-\ell'}, y_{t-\ell'})\forall \ell' \in [0, L']$, where for this section we have dropped the time subscripts for the sake of readability. From context it is clear that we are dealing with the predictor variable therefore, it deals with data before the current time t. The predictor is usually a multi dimensional variable, as it contains signals from multiple sensors. For sake of convenience we represent the multi dimensional predictor variable $x = [x^1, x^2, \cdots, x^S]$, where $x^s, s \in S$ is the signal from the s^{th} sensor. Then $w_n^{x^s}$ denotes the reconstructed signal from wavelet packet decomposition at level n for predictor signal x^s for the s^{th} sensor. Therefore w_n^x is the combination of all predictor signals reconstructed from WPD for each predictor sensor signal at level n.This combined with w_n^u is the output of this layer:

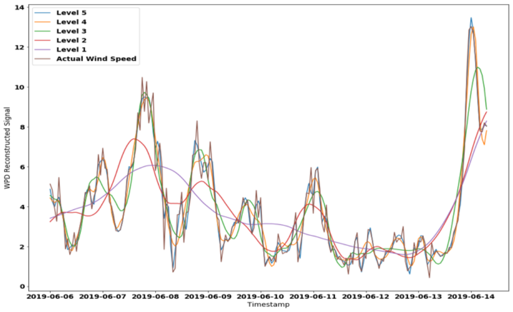

Figure 2: Wind Speed - Wavelet Packet Decomposition

$$\begin{aligned}o_{WPD}^{(n,m)} &= w^x \times w_n^u\\ &= \{(w_n^x, w_m^x); \forall n, m \in [1, N]\},\end{aligned}$$

where × is the cartesian product. Each element of the output contains all combinations of various scales of the predictor signals x and the error signal u. We do not generate all combinations of various signals for each individual predictor sensor signal to keep the size of the deep neural network manageable.

For training, the data is trained as paired variable (x, u), where each paired set consists of $\{x_{t-\ell'}, \ell' \in [0, L']\}$ as the input and $\{u_{t+\ell}, \ell \in (0, L]\}$ as the output.

6.4 Multi-Scale Deep Learning

Once we have prepared the output data from WPD in the previous step $o_{WPD}^{(n,m)}, \forall n, m \in [1, N]$, this data is the input data for the deep learning network. We separate out the data into long scale (n or $m = 1$), medium-scale ($n, m \in [2, N-1]$) and short scale (n or $m = N$) signals. The long scale signals pass through a CNN-LSTM stack. The medium-scale and the short scale signals pass through a multi-layered CNN stack. For the data with short-term dependencies (medium and short scale data), the CNN layer has similar performance and faster computing speed when compared to the LSTM recurrent layer, thus we use CNN network layers for the medium and short scale data. While, for the long scale data, the CNN network layers extract the deep features of the time series and the LSTM layer sequentially process the temporal data with long-term and short-term dependence. Therefore CNNLSTM architecture is used for long scale data.

We settled on this particular design choice because it achieved the best performance compared across predictions of all climatic parameters. The heuristic on choosing CNNs for short and medium scale ($n, m \in [2, N-1]$) while CNNLSTM for long scale data (m or $n = 1$) enables generalization of DeepMC across varying dimension of the input considered. More implementation details are in the Supplementary section A.

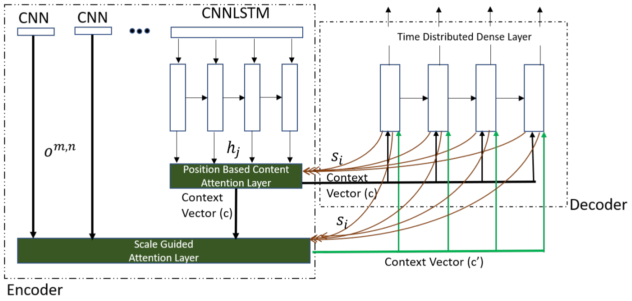

Figure 3: Encoder-decoder two level attention mechanism

6.5 Attention Mechanism

DeepMC uses a 2 levels of attention models. A similar attention model is used in vision-to-language tasks [21]. The first level attention model is long range guided attention model which is used with the CNN-LSTM output and it memorizes the long term dynamics of the input time series. Various attention models have been used in direct sequence to sequence RNNs to capture memory of the input sequence (or time series representation) and pseudo-periods within the time series [7, 35]. DeepMC uses a position based content attention model described by Cinar et al. [35] for this level. DeepMC uses multivariate version of the attention mechanism described in [35] but for sake of keeping the equations legible we will omit notations specifying for each individual feature vector in the formulation below. The LSTM in the CNNLSTM encoder stack represents each input $l_i, 1 \le i \le T$ as a hidden state: $h_i = F(l_i, h_{i-1})$, with $h_i \in \mathbb{R}^H$ and where the function F is a non-linear transformation corresponding to the LSTM layers and H is the dimension of the hidden layer. The LSTM decoder (described in Section 6.6) parallels the encoder by associating each output $m_i, 1 \le i \le T'$ to a hidden state vector s_i that is directly used to predict the output:

$$m_i = G(m_{i-1}, s_{i-1}, c_i),$$

with $s_i \in \mathbb{R}^{H'}$, H' is the dimension of the decoder hidden layer, c_i is usually referred to as a context and corresponds to the output of the memory model. For DeepMC function G corresponds to an LSTM with a context integration. Using these notations the long-range guided attention model based on the position based content attention mechanism is formulated as RNN-$\pi^{(2)}$ in [7].

$$RNN - \pi^{(2)} :$$
$$\begin{cases} e_{ij} &= v_a^T tanh(W_a s_{i-1} + U_a((\pi^{(2)} \Delta^{i,j}) \odot h_j))\Delta_{i+T-j} \\ \alpha_{ij} &= softmax(e_{ij}), c_i = \sum_{j=1}^{T} \alpha_{ij} h_j, \end{cases}$$

where $W_a, U_a, \pi \in \mathbb{R}^{2H \times (T+T')}$ and v_a are trained in conjunction with the entire DeepMC deep learning architecture, $\Delta^{(i,j)} \in \mathbb{R}^T$ is a binary vector that is 1 on dimension $(i+T-j)$ and 0 elsewhere, $\odot$ denotes the element wise multiplication (Handmard product) and $\Delta \in \mathbb{R}^{T+T'}$ has 1 on its first T coordinates and 0 on the last T'.

The second level attention model is scale guided attention model and is used to capture the respective weighting of different scales.

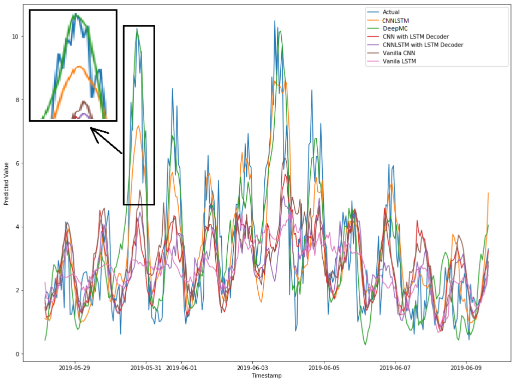

Figure 4: Micro-Climate Wind Speed prediction comparisons at the 24th hour with a resolution of 1 hour over a 10 day period

The scale guided attention model uses an additive attention mechanism described here. The outputs of the multi-scale model (including the output of the long-range guided attention mechanism on the CNNLSTM stack) is represented as $o^{(m,n)}, m, n \in [1, N]$. For sake of convenience, we introduce a single index j for the tuple (m, n). Then the attention mechanism context vector c'_i is defined as:

$$c'_i = \sum_{j}^{N^2} \alpha'_{ij} o^{(j)}.$$

The weight α'_{ij} of each output $o^{(j)}$ is computed by

$$\alpha'_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{N^2} e_{ik}},$$

where $e_{ij} = \tanh(w'^{T}_{i,j}(s_{i-1}; o^{(j)}))$, $w'_{i,j}, i \in [1, T']; j \in [1, N^2]$ is trained in conjunction with the entire DeepMC deep learning architecture.

6.6 Decoder

The DeepMC decoder uses LSTM to generate a sequence of L outputs, which is equal to the number of future timesteps to be predicted. The decoder LSTM layer receives a multivariate encoded time series and produces a vector for each step of prediction. Each output of the LSTM is connected with 2 layers of time distributed fully connected layer.

7 REAL WORLD DEPLOYMENTS

DeepMC is deployed across many different regions of the world using the FarmBeats [34] technology. In this section, we present 4 real world scenarios in agriculture and energy which are a projection of common situations effected by weather conditions. We also show some results in comparison to common models used to solve prediction tasks in addition to comparisons with some variations on DeepMC.

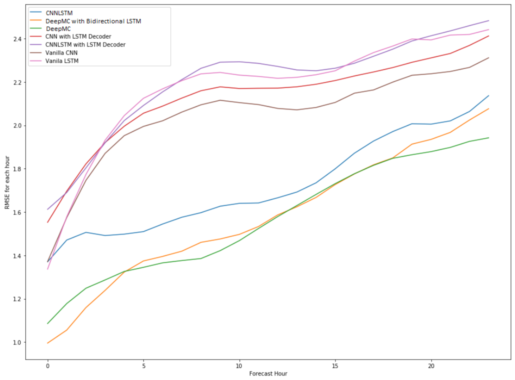

Figure 5: Micro-Climate Wind Speed prediction RMSE comparisions over 24 hour predictions

7.1 Comparison: Micro-Wind Speed Predictions

We compare DeepMC with other deep learning architectures and forecasting techniques across wide variety of climatic parameters. Figure 4 shows the wind speed predictions at the 24th hour over a period of 10 days with 1 hour resolution. Figure 5 plots the RMSE for each hour prediction and compares with other models. It is observed that DeepMC has significantly better performance compared to other models and is more likely to follow the details and trends of the time series data. Other models used for comparison (in this case for Wind Speed) are the CNNLSTM model proposed in [22], modified CNNLSTM with LSTM decoder, regular convolutional network with LSTM decoder, a vanilla LSTM based forecaster, and a vanilla CNN based forecaster. Another interesting observation is that the performance of all models decrease as the horizon of prediction increase, which is to be expected as it is more accurate to predict the next immediate hour vs a forecast on the 24th hour. Figure 6 plots the RMSE(Root Mean Squared Error), MAE(Maximum Absolute Error) and MAPE(Maximum Absolute Percentage Error) values for micro-wind speed predictions using various models, averaged over all 24 step predictions. The DeepMC model outperforms other commonly used models on all metric. The RMSE value averaged over all prediction steps for DeepMC is 1.52, MAE value is 1.2 and MAPE is 4.89. In comparison this gives a promoting percentage of 44.1% and 37.5% of RMSE, 19.1% and 27.5% of MAE, and 49.3% and 86.01% of MAPE over CNNLSTM network (the next best performing Deep Learning network) and ARIMA, respectively. Promoting percentage is defined as the percentage increase in accuracy compared to baseline. The predictors used for predicting micro-temperature are: a) From the IoT sensors - Ambient Temperature, Ambient Humidity, Precipitation, Wind Speed; b) From the weather station - historical Wind Speed forecasts. This data was collected for a period of 2 years with 1 hour resolution.

7.2 Solar Farm: Micro-radiation predictions

Micro-radiation predictions are required to estimate the electricity produced at solar farms. This scenario is for a commercial solar farm site. These predictions are fed into an optimization model to fulfill price and energy commitments by the utility company in the energy markets. Radiation received at the solar panel is sensitive to seasons of high overcast or rain. Figure 7 plots the predictions across months during the overcast season and after. The predictors used for predicting micro-radiation are: a) From the IoT sensors - Ambient Temperature, Ambient Humidity, Precipitation, Wind Speed, Radiation, and cloud cover; b) From the weather station - historical radiation forecasts. The training data available was over a period of 12 months and testing was conducted over a period of 3 months, both with 15 minute resolution averaged over each hour. The predictions attain a high accuracy for the month after the monsoon in July, with scores MASE[7] = 1.86, MAE= 65.14, RMSE = 116.30. While there is a slight decrease in performance during the monsoon season in June, with MASE = 1.96, MAE = 68.61, and RMSE = 125.44. The plots also show a 2σ Bollinger Bands which gives an idea of the uncertainty in predictions. We observe here that the uncertainty estimates are "targeted", i.e. there is more uncertainty in predictions when the reality is also more uncertain. The bands are tighter when we are more certain. This observation gives us a straw man verification of the underlying model. Table 1 compares DeepMC’s MAE, MAPE and RMSE scores with other commonly used models.

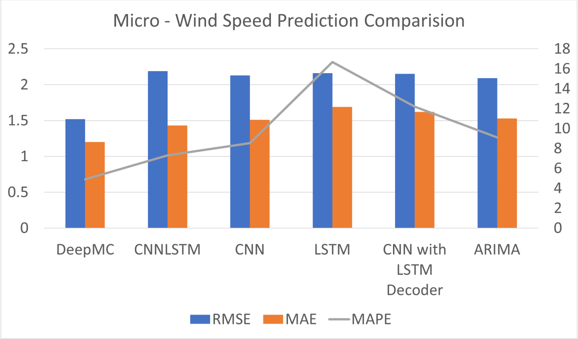

Figure 6: RMSE, MAPE and MAE comparision for Micro-Climate Wind Speed predictions

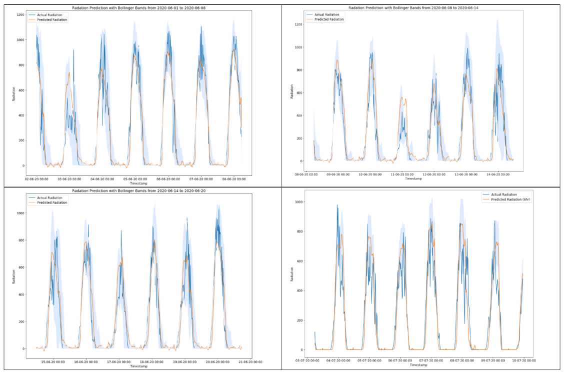

Figure 7: DeepMC Micro-Climate radiation prediction at the 24th hour and 1 hour resolution with Bollinger Bands

[7] We use Mean Absolute Scaled Error[18] rather than MAPE because of the division by zero isse

	DeepMC	CNN	LSTM	CNNLSTM	ARIMA
RMSE	124.5	167.4	192.3	155.6	530.60
MAE	68.15	111.77	130.99	90.02	397.45
MASE	1.95	3.20	3.75	2.89	11.39

Table 1: Micro-radiation Prediction Scores for various models

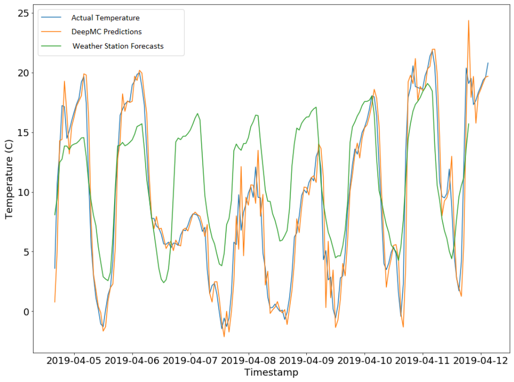

Figure 8: DeepMC Micro-Climate temperature 6 day sequential predictions with a resolution of 6 hour

7.3 Fertilization: Micro-temperature predictions

This scenario is the one presented in the Introduction. The farmer owns approximately 9000 acres of land across a region which is quite hilly. There are many distinct micro-climate regions in this farm. Climatic parameters vary significantly among various regions of the farm and also between the nearest commercial weather forecast provider and the readings on the ground. The farmer uses DeepMC predictions for advisory on temperature forecasts at specific locations on his farm. In this scenario, the farmer consults DeepMC for temperature predictions for specific locations to plan logistics and operations for fertilization. Figure 8 shows a 6 day forecast with a temporal resolution of 6 hours. The figure shows the comparison of the results obtained by DeepMC with Dark Sky weather forecast (from the nearest station) and the actual temperatures recorded using IoT sensors in retrospect. Based on DeepMC’s predictions the farmer postponed his fertilization for the period between 07-April-2019 to 11-April-2019 as the temperature predicted by DeepMC were below freezing. Instead, had the farmer relied on weather station forecasts, which consistently showed temperatures above freezing (more that 5C), then he would have been at risk of endangering the crop losing upto 20% in yield. In many places, especially small holder farms, this percentage is significant enough to decide whether the farmers will be able to achieve basic sustenance of food and supplies in the coming year or not. For this particular farm and location, DeepMC predictions for ambient temperature has recorded RMSE of 1.35 and MAPE of 7.68% (implying accuracy of 92.32%) for the data recorded in Figure 8. The predictors used for predicting micro-temperature are: a) From the IoT sensors - Ambient Temperature, Ambient Humidity, Precipitation, Wind Speed; b) From the weather station - historical ambient Temperature forecasts.

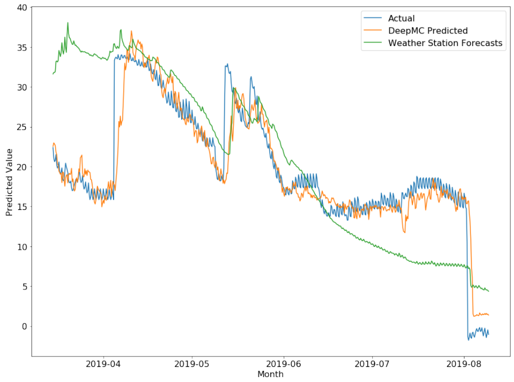

Figure 9: DeepMC Micro-climate soil moisture 4th day prediction with 6 hour resolution over a period of 10 months

7.4 Phenotyping Research: Micro-soil-moisture predictions

The producer is interested in experimenting with different growing techniques for vine tomatoes. The vine tomatoes are susceptible to rot if they are too close to the soil with high moisture values. Generally, growers use trellises to lift up the vines and provide structural stability. The trellises add more challenges to manage the crops over the growing season. The producer here is interested in growing tomatoes without the trellises. This critically depends on being able to predict the local soil moisture values accurately. The producer uses DeepMC for advisory on micro-soil-moisture conditions. The predictors used for predicting micro-soil-moisture are: a) From the IoT sensors - Ambient Temperature, Ambient Humidity, Precipitation, Wind Speed, Soil Moisture and Soil Temperature; b) From the weather station - historical Soil Moisture forecasts. The results are shown in Figure 9 with the recorded RMSE value of 3.11 and MAPE value of 14.03% (implying a 85.97% accuracy). Soil moisture values are highly sensitive to changing weather, as the moisture increases rapidly during times of heavy rainfall and slowly decreases during extended dry periods, which is observed in Figure 9. DeepMC tracks these sharp changes fairly accurately, and much better than the weather station forecasts, which is an evidence of the robustness of the model.

8 DISCUSSION, SUSTAINABILITY, AND CONCLUSION

Micro-climate predictions through DeepMC allow for better cost control by generating predictions based on relatively affordable IoT sensors. The predictions computed enables our partner farmers to apply chemicals with better timing allowing them to be as effective as possible. Many weeds that are controlled by chemicals are gaining resistance, so the more effective the chemical is at the time of application, the more likely that the weed will not develop resistance to it. This allows for less chemical application overall. Thus helping with sustainable agriculture. Another way that DeepMC helps with environmental issues is by helping to make commercial operations of non-renewable energy production successful. Energy utility companies can fulfill their power and price commitments in the energy market if they can successfully predict radiation and wind-speed at their solar and wind farms, respectively.

The supplementary section A provides various implementation details for the readers to experiment and build upon the micro-climate prediction framework presented here. Also we present how the DeepMC architecture allows itself to be used as a platform where specialized models provided by domain experts can be embedded to increase the collective performance of predictions, and thus allowing for commercial adoption at scale.

DeepMC is observed to achieve compelling results on multiple micro-climate prediction tasks. To the best of our knowledge, this is the most versatile study and framework for micro-climate prediction for multiple climatic parameters and multiple geographical conditions around the world. Still, in this study, we found many opportunities for further improvement for better reliability, robustness, and accuracy. Specifically, the model is brittle on transfer learning. We observe that it requires careful hyper-parameter tuning and initialization to achieve good performance. Additional work using GANs can be explored to increase the transfer-ability of the DeepMC framework. Nonetheless, there are plenty of demonstrable cases where DeepMC is being used and aiding decisions on the farms effectively. This work pushes the boundaries of engineering a real-world micro-climate prediction framework.

REFERENCES

[1] Jonas Allegrini and Jan Carmeliet. 2017. Coupled CFD and building energy simulations for studying the impacts of building height topology and buoyancy on local urban microclimates. *Urban Climate* 21 (2017), 278–305.

[2] J Scott Armstrong and Long-Range Forecasting. 1985. From crystal ball to computer. *New York ua* (1985).

[3] Dimitros Asteriou and Stephen G Hall. 2011. ARIMA models and the Box–Jenkins methodology. *Applied Econometrics* 2, 2 (2011), 265–286.

[4] Mohammad Taha Bahadori, Qi Rose Yu, and Yan Liu. 2014. Fast multivariate spatio-temporal analysis via low rank tensor learning. *Advances in neural information processing systems* 27 (2014), 3491–3499.

[5] Yu Cai, Wengang Zheng, Xin Zhang, Lili Zhangzhong, and Xuzhang Xue. 2019. Research on soil moisture prediction model based on deep learning. *PloS one* 14, 4 (2019), e0214508.

[6] Liu Chun-Lin. 2010. A tutorial of the wavelet transform. *NTUEE, Taiwan* (2010).

[7] Yagmur Gizem Cinar, Hamid Mirisaee, Parantapa Goswami, Eric Gaussier, Ali Aït-Bachir, and Vadim Strijov. 2017. Position-based content attention for time series forecasting with sequence-to-sequence rnns. In *International Conference on Neural Information Processing*. Springer, 533–544.

[8] A. Eleftheriou, K. Kouvaris, P. Karvelis, and C. Stylios. 2018. Micro Climate Prediction Utilising Machine Learning Approaches. In *2018 IEEE International Workshop on Metrology for the Sea; Learning to Measure Sea Health Parameters (MetroSea)*. 197–200. https://doi.org/10.1109/MetroSea.2018.8657903

[9] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. 2019. InceptionTime: Finding AlexNet for Time Series Classification. *arXiv preprint arXiv:1909.04939* (2019).

[10] Rao Fu, Jie Chen, Shutian Zeng, Yiping Zhuang, and Agus Sudjianto. 2019. Time Series Simulation by Conditional Generative Adversarial Net. *arXiv preprint arXiv:1904.11419* (2019).

[11] George Galanis, Evgenia Papageorgiou, and Aristotelis Liakatas. 2017. A hybrid Bayesian Kalman filter and applications to numerical wind speed modeling. *Journal of Wind Engineering and Industrial Aerodynamics* 167 (2017), 1–22.

[12] José Luis Pérez Galaso, Isidro Ladrón de Guevara López, and Javier Boned Purkiss. 2016. The influence of microclimate on architectural projects: a bioclimatic analysis of the single-family detached house in Spain's Mediterranean climate. *Energy Efficiency* 9, 3 (2016), 621–645.

[13] M Ghamariadyan, MA Imteaz, and F Mekanik. 2019. A hybrid wavelet neural network (HWNN) for forecasting rainfall using temperature and climate indices. In *IOP Conference Series: Earth and Environmental Science*, Vol. 351. IOP Publishing, 012003.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[15] Jiu Gu, Yining Wang, Da Xie, and Yu Zhang. 2019. Wind Farm NWP Data Preprocessing Method Based on t-SNE. *Energies* 12, 19 (2019), 3622.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. 1026–1034.

[17] JW Hummel, KA Sudduth, and SE Hollinger. 2001. Soil moisture and organic matter prediction of surface and subsurface soils using an NIR soil sensor. *Computers and electronics in agriculture* 32, 2 (2001), 149–165.

[18] Rob J Hyndman and Anne B Koehler. 2006. Another look at measures of forecast accuracy. *International journal of forecasting* 22, 4 (2006), 679–688.

[19] MB Jones. 1993. Plant microclimate. In *Photosynthesis and production in a changing environment*. Springer, 47–64.

[20] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[21] Xuelong Li, Aihong Yuan, and Xiaoqiang Lu. 2019. Vision-to-language tasks based on attributes and attention mechanism. *IEEE transactions on cybernetics* (2019).

[22] Hui Liu, Xiwei Mi, and Yanfei Li. 2018. Smart deep learning based wind speed prediction model using wavelet packet decomposition, convolutional neural network and convolutional long short term memory network. *Energy conversion and management* 166 (2018), 120–131.

[23] Peter Lynch. 2008. The origins of computer weather prediction and climate modeling. *J. Comput. Phys.* 227, 7 (2008), 3431–3444.

[24] Spyros Makridakis, Steven C Wheelwright, and Rob J Hyndman. 2008. *Forecasting methods and applications*. John wiley & sons.

[25] Kathleen McLaughlin. 2017. Gaps in 4G network hinder high-tech agriculture: FCC prepares to release 500 million to improve coverage.

[26] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[27] Aekyeung Moon, Ki Young Moon, and Seung Woo Son. [n.d.]. Microcilmate-Based Predictive Weather Station Platform: A Case Study for Frost Forecast. ([n. d.]).

[28] K Papagiannaki, K Lagouvardos, V Kotroni, and G Papagiannakis. 2014. Agricultural losses related to frost events: use of the 850 hPa level temperature as an explanatory variable of the damage cost. *Natural Hazards and Earth System Sciences* 14, 9 (2014), 2375.

[29] K Ravikumar and S Tamilselvan. 2014. On the use of wavelets packet decomposition for time series prediction. *Appl. Math. Sci* 8, 58 (2014), 2847–2858.

[30] Norman J Rosenberg, Blaine L Blad, and Shashi B Verma. 1983. *Microclimate: the biological environment*. John Wiley & Sons.

[31] Mahesh Chand Singh, JP Singh, and KG Singh. 2018. Development of a microclimate model for prediction of temperatures inside a naturally ventilated greenhouse under cucumber crop in soilless media. *Computers and electronics in agriculture* 154 (2018), 227–238.

[32] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[33] T Vanwalleghem and RK Meentemeyer. 2009. Predicting forest microclimate in heterogeneous landscapes. *Ecosystems* 12, 7 (2009), 1158–1172.

[34] Deepak Vasisht, Zerina Kapetanovic, Jongho Won, Xinxin Jin, Ranveer Chandra, Sudipta Sinha, Ashish Kapoor, Madhusudhan Sudarshan, and Sean Stratman. 2017. Farmbeats: An iot platform for data-driven agriculture. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*. 515–529.

[35] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in neural information processing systems*. 2692–2700.

[36] Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer. 2019. Quant GANs: deep generation of financial time series. *arXiv preprint arXiv:1907.06673* (2019).

[37] Jun-Ichi Yano, Michał Z Ziemiański, Mike Cullen, Piet Termonia, Jeanette Onvlee, Lisa Bengtsson, Alberto Carrassi, Richard Davy, Anna Deluca, Suzanne L Gray, et al. 2018. Scientific challenges of convective-scale numerical weather prediction. *Bulletin of the American Meteorological Society* 99, 4 (2018), 699–710.

[38] Florian Zellweger, Pieter De Frenne, Jonathan Lenoir, Duccio Rocchini, and David Coomes. 2019. Advances in microclimate ecology arising from remote sensing. *Trends in ecology & evolution* (2019).

[39] Bangyou Zheng, Scott C Chapman, Jack T Christopher, Troy M Frederiks, and Karine Chenu. 2015. Frost trends and their estimated impact on yield in the Australian wheatbelt. *Journal of Experimental Botany* 66, 12 (2015), 3611–3623.

[40] Weidong Zou, Fenxi Yao, Baihai Zhang, Chaoxing He, and Zixiao Guan. 2017. Verification and predicting temperature and humidity in a solar greenhouse based on convex bidirectional extreme learning machine algorithm. *Neurocomputing* 249 (2017), 72–85.

SUPPLEMENTARY MATERIAL

A DEEPMC IMPLEMENTATION DETAILS

This section describes some of the implementation details of the DeepMC architecture to aid reproducibility.

A.1 Preprocessing of sensor data:

As mentioned in Section 6.1 we use the ARIMA forecasting model to fill in missing data. We use the python module `statsmodels`[8] with parameter values: Number of time lags of the autoregressive model, $p = 5$; Degree of differencing, $d = 1$ and; Order of the moving-average model, $q = 0$.

A.2 Forecast Error Computation:

Using the notations defined in Section 6.2 we use various values for L and L' depending on the problem of interest. Typically $L' \geq L$ and in the range of $L, L' \sim 24$ which can signify 24-hour retrospective and predictive interval with 1-hour resolution or 3-day retrospective and predictive interval with 6-hour resolution or any such combination.

A.3 Wavelet Packet Decomposition

Wavelet Packet Decomposition is described in Section 6.3. We use a 5 level decomposition using Daubechies wavelet function.

A.4 Multi Scale Deep Learning

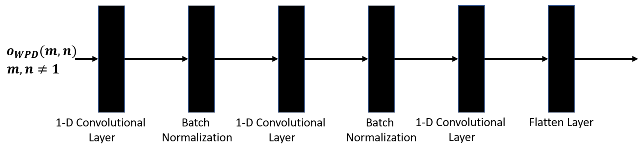

Figure 10: CNN stack

The architecture described in Figure 1 and Section 6.4 contains a CNN stack and a CNN-LSTM stack. The CNN stack (Figure 10) uses three 1-D convolutional layers with filter sizes of 4 and ReLU activation function for each layer. Each layer is batch normalized before passing onto the next layer. The last layer flattens the output. The CNN-LSTM (Figure 11) stack uses two 1-D convolutional layers with filter sizes of 4 and ReLU activation function for each layer. Each layer is batch normalized before passing onto the next layer. The output of the last layer is passed through an LSTM layer with ReLU activation function and a dropout rate of 20%. The convolutional layers use a He Normal [16] Initialization.

A.5 Decoder

The decoder described in Section 6.6 uses a 20 node LSTM layer with the ReLU activation function. Additionally, the decoder also uses ReLU activation for the first dense layer and a linear activation function for the second dense layer. The first dense layer has 50 nodes for each of the time series steps and the second dense layer has 1 node for each of the time series steps.

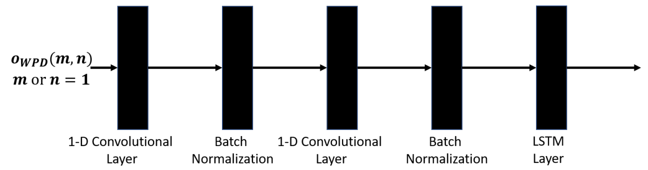

Figure 11: CNN-LSTM stack

The entire model, as summarized in Figure 1, is trained using the mean squared loss function with Adam optimizer [20].

B ENGINEERING A REAL-WORLD MICRO-CLIMATE PREDICTION FRAMEWORK

In addition to the points mentioned in Section 4, there are a few other challenges to overcome for a micro-climate prediction framework to be feasible. Firstly, in most live deployments we observed that the amount of data is quite less, and there is a general expectation for the model to learn quickly with less data. To overcome this problem we implemented a GAN based solution to transfer models learned at another site. The details are provided in Subsection B.1. Additionally, most often frameworks like DeepMC are used by domain experts such as Agronomists, Energy researchers, etc. In many cases, the domain experts have a specialized model for forecasting a specific micro-climate parameter. Although, these models lack in their generalizability and fail to capture short-scale dynamics but they do provide precise modeling of the specific use case and a good understanding of long-range dynamics. With the scale-based approach (with emphasis on attention mechanisms) for DeepMC, we observed that these 3rd party models can easily be combined into the base DeepMC framework. This enables DeepMC to be used as Platform-as-a-Service (PaaS), which allows for scalability and ease of use. Figure 12 shows how 3rd party scales and models can be combined with the base DeepMC framework. Once the specific model or scale is ingested then the entire network is retrained end-to-end. We tested this approach by combing ARIMA outputs with the base DeepMC for the micro-radiation prediction described in Section 7.2. The accuracy increased by a promoting percentages of $P_{RMSE} = 12.9\%$, $P_{MASE} = 12.7\%$ and $P_{MAE} = 9.7\%$ with the new RMSE = 107.58, MASE = 1.73 and MAE = 60.37.

B.1 Transfer Learning using GAN

We use GAN [14] to transfer models learned on one domain to sensors where a sufficient paired labeled dataset is not available. GANs have been extensively used in image-based tasks but their exploration in time series data is limited and developing [10, 36]. In the context of microclimate time series data we interpret generator as the microclimate predictor, while the discriminator is a binary time series classifier, which discriminates between the predicted microclimate parameter ($\hat{y_{t+\ell}}$) and actual observations ($y_{t+\ell}$). The generator is the DeepMC model, while the discriminator uses the InceptionTime model by Fawaz et al. [9]. Figure 13 shows the gan architecture used for transfer learning. The GAN archtecture uses the InceptionTime [9] model for the discriminator. The model is

[8] https://www.statsmodels.org/stable/generated/statsmodels.tsa.arima_model.ARIMA.html

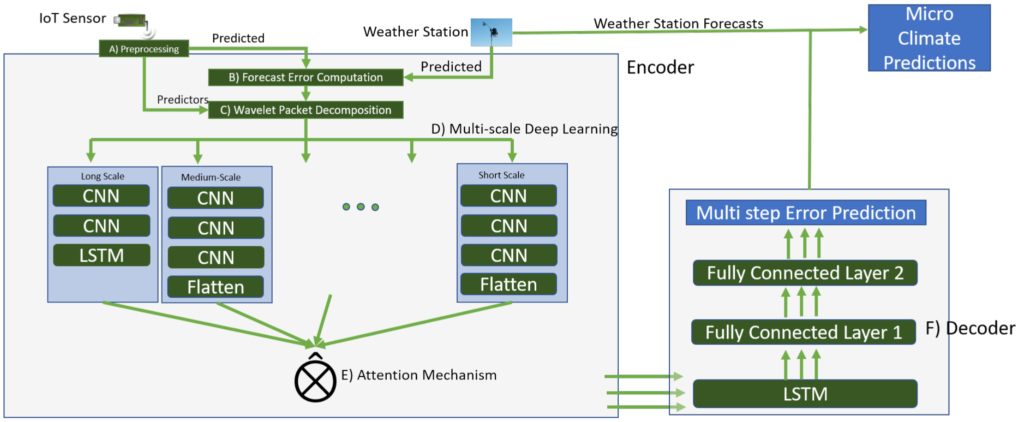

Figure 12: DeepMC PaaS

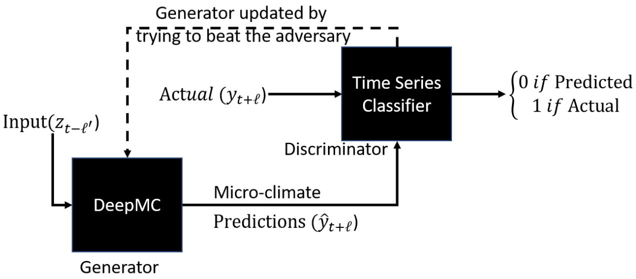
Generator updated by trying to beat the adversary
Actual ($y_{t+\ell}$)
Time Series Classifier
$\begin{cases} 0 \text{ if Predicted} \\ 1 \text{ if Actual} \end{cases}$
Input($z_{t-\ell'}$)
Discriminator
DeepMC
Micro-climate Predictions ($\hat{y}_{t+\ell}$)
Generator

Figure 13: GAN architecture for Transfer Learning. DeepMC is described in Figure 1 and Time Series Classifier is sketched in Figure 14.

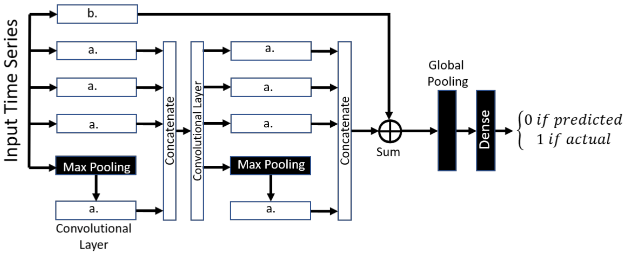

Figure 14: GAN Discriminator: InceptionTime model which classifies between time series predicted from DeepMC model and actual observations.

plotted in Figure 14. This model uses various stacks of convolutional layers each which has ReLU as the activation function. The convolutional layers marked *a.* are all similar with 32 filters each of 3, 5 and 8 kernel size. The convolutional layer marked *b.* uses 128 filters with kernel size 1. The global pooling layer pools and averages the input across time resulting in a vector of dimension 128 which is fed into the fully connected layer with 1 node and sigmoid activation function. The discriminator is trained using the binary cross entropy with Adam optimizer. During transfer learning, the DeepMC model is updated while trying to beat the adversary which discriminates between the predicted results and the actual observation on the target domain. The algorithm is presented in Algorithm 1.

Additional note: The discriminator, as it is trained, provides high accuracy ($>$ 99%+ accuracy) for time series classification. Although, when the generator is pitted against the adversary (the discriminator model) and updated, in some cases the generator model is observed to settle in a local optimum where it learns just enough basic characteristics to fool the discriminator but the accuracy of the generator on the new domain is not greatly improved. It will be interesting to explore new methods to address these problems. Similar issues are observed in generative image-based tasks and a modified GAN called CGAN was proposed by Mirza et al. [26] to address similar problems. We are experimenting with a modified conditioned GANs for time series generative tasks such as micro-climate prediction described in this paper. In this case, the new generator is a modified DeepMC framework conditioned on the actual weather output.

Algorithm 1: DeepMC GAN model for transfer learning

```
Result: Updated DeepMC model
for Number of training iterations do
    for Number of training iterations do
        Sample minibatch of training samples for the
          discriminator model;
        Train\Update the discriminator model;
    end
    Freeze the discriminator weights from updating;
    Sample minibatch of training sample for the combined
      DeepMC and discriminator model;
    Update the generator DeepMC model (while pitted
      against the adversary);
end
```