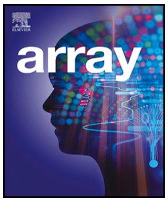
Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/locate/array



Minute-wise frost prediction: An approach of recurrent neural networks

Ian Zhou [a,b,*], Justin Lipman [a,b], Mehran Abolhasan [a], Negin Shariati [a,b]

[a] University of Technology Sydney, Australia
[b] Food Agility CRC Ltd, 81 Broadway, Ultimo, NSW, 2007, Australia

ARTICLE INFO

Keywords:
Frost prediction
Internet of Things
Machine learning
Recurrent neural network
Temporal prediction

ABSTRACT

Frost events incur substantial economic losses to farmers. These events could induce damage to plants and crops by damaging the cells. In this article, a recurrent neural network-based method, automating the frost prediction process, is proposed. The recurrent neural network-based models leveraged in this article include the standard recurrent neural network, long short-term memory, and gated recurrent unit. The proposed method aims to increase the prediction frequency from once per 12–24 h for the next day or night events to minute-wise predictions for the next hour events. To achieve this goal, datasets from NSW and ACT of Australia are obtained. The experiments are designed considering the scene of deploying the model to the Internet of Things systems. Factors such as model processing speed, long-term error and data availability are reviewed. After model construction, there are three experiments. The first experiment tests the errors between different model types. The second and third experiments test the effect of sequence length on error and performance for recurrent neural network-based models. All tests introduce artificial neural network models as the baseline. Also, all tests for model error are conducted in two rounds with testing datasets from the current year (2016) and next year (2017). As a result, recurrent neural network-based models are more suitable for short-term deployment with a smaller sequence length. In contrast, artificial neural network models demonstrate a lower error over the long term with faster processing time. With the results presented, the limitations of the proposed method are discussed.

1. Introduction

In the field of agriculture, frosts occur when ice crystals are formed within the plants and damage the cells [1]. As a result, frost could cause significant losses on the economy and ecosystem [2]. Currently, there are many active and real-time protection methods against frost, including heaters, sprinklers, artificial fog, and air disturbance technologies [3]. However, the frost prediction methods automating the activation of these protection methods can still be improved [3].

This article focuses on predicting the condition of future frost damage to plants. The potential of Recurrent Neural Networks (RNNs) in frost prediction is explored in this article. RNNs are a special form of artificial neural network (ANN) with recurrent connections, which provide the capability to recognize sequential patterns [4]. RNNs are different from the basic ANN that only accepts one input at a time. RNNs can accept several inputs in a sequence. In terms of time-series data, individual data points are processed at once in the sequence of time [4]. The output of the current time state is generated from the input of the current time state and the output of the previous time state, recursively [5]. The standard RNN has issues such as gradient explosion and gradient vanishing [4]. To address these issues, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are proposed as variants of the RNN [5]. This article leverages RNN, LSTM, and GRU models for frost prediction.

In recent years, the Internet of Things (IoT) technologies have been widely applied in the field of agriculture to provide real-time monitoring and actuation services [6]. There are also a few IoT-based frost protection systems. However, most of these frost protection systems rely on thresholds of real-time sensor readings to trigger the frost protection equipment [3]. The effect of these simple mechanisms is limited compared to the accuracy of the prediction algorithms [3]. Therefore, this article considers a few factors related to the future deployment of frost prediction algorithms. These factors include model processing speed, long-term accuracy and data availability. Since system resources are limited for IoT systems, the model should require a faster processing speed [7]. Also, IoT systems should eliminate extra human interventions [6]. Therefore, the deterioration of model accuracy over time should be minimum to ensure manual updates to IoT nodes are infrequent. Finally, as most frost prediction models depend on on-site

[*] Corresponding author at: University of Technology Sydney, Australia.
E-mail addresses: ian.zhou@student.uts.edu.au (I. Zhou), justin.lipman@uts.edu.au (J. Lipman), mehran.abolhasan@uts.edu.au (M. Abolhasan), negin.shariati@uts.edu.au (N. Shariati).

https://doi.org/10.1016/j.array.2022.100158
Received 7 October 2021; Received in revised form 4 March 2022; Accepted 1 April 2022
Available online 9 April 2022

historical data [3], data availability is important when creating these models. Since a large amount of historical data cannot be assumed to be available at all sites, our set scene assumes that only a small amount (three months) of data is available to minimize the data collection time for new models.

1.1. Related work

In [3], frost prediction methods are categorized as "classification methods" and "regression methods". Classification methods predict the occurrence of frost as a percentage in a future time, whereas regression methods predict the minimum temperature in a future period [3]. Both methods rely on climate data as the model input. Since the frost resistances for different species of plants are different [8], frost regression prediction methods are proposed in this article to provide the farmers future environmental insights and provide a more generalized solution avoiding the differences between individual plant species.

There are a few existing frost regression prediction methods. In [9–11], traditional machine learning methods are leveraged to predict temperature or minimum temperature in the next day or night. Random forest models are used in [11] to predict next-day minimum temperature with temperature and humidity inputs. Linear regression is used in both [9] and [10]. Environmental parameters such as temperature, dew point, and humidity are inserted as model inputs in [10]. On the other hand, to consider the effect of wind machines, the authors of [9] introduced the distance to wind machines along with elevation, time of local sunset, and radiation received during the previous day as input parameters.

Apart from the traditional machine learning models. ANN with fully connected layers, as a deep learning model, can also predict future minimum temperatures [12–14]. Models in [12–14], predict minimum temperature in the next 12–24 h as a numerical value. These three works all implement prediction models with air temperature, relative humidity, precipitation, wind direction and speed. However, [12] also includes daytime length, daytime maximum and minimum temperature to support night temperature predictions with a daytime baseline. In [13], precipitation, cloud cover, moisture, and pressure are included as model inputs. The authors also considered humidity and wind velocity at 19:00. The authors of [14] predicted next day minimum temperature with fewer input parameters, but introduced radiation to build their prediction ANN.

The above machine learning and deep learning models all predict frost conditions in the next 12–24 h [9–14]. Therefore, in extreme conditions, protection equipment might need to be switched on for 12–24 h to ensure zero frost damage when solely considering model predictions. However, by constant manual observations, the operational time of protection equipment could be reduced [3]. Hence, to reduce the operational time automatically, the major aim of this article is to implement minute-wise next hour minimum temperature prediction for frost prediction. Also, as mentioned in the above paragraphs, the potential of RNN-based models (RNN, LSTM, GRU) are explored to solve this prediction problem. The performance of different RNN-based models is also compared in this article. In conclusion, the major contributions of this article are presented as follows.

1. Application of an RNN-based frost prediction method.
2. Increasing the prediction frequency from once per 12–24 h for the next day or night events to minute-wise predictions for the next hour events.
3. Evaluated the limitations of RNN-based frost prediction.

The rest of this paper is arranged as follows. Section 2 describes the methodology and experiment settings with the study area, data processing procedures, and experiments. Experiments include comparing temperature prediction models (ANN, RNN, LSTM, GRU), analyzing different RNN model settings, and the performance of predicting minimum temperature with other frost-related parameters. Then, the experimental results are discussed in Section 3 and lead towards limitations with open challenges. In the end, Section 4 concludes the whole article.

2. Methodology

This section explains the methodology and settings of the experiments. The study area is firstly explained, followed by the data preprocessing procedures. Then, model construction and testing leveraging the preprocessed data are described. Finally, the experiment processes are summarized.

2.1. Study area

The study area is located in New South Wales and Australian Capital Territory of Australia. In the study area, datasets from 30 different weather stations are obtained. Fig. 1 is a map of the study area with the weather station locations and IDs. Also, a list of location coordinates of the weather stations are presented by Tables A.7 and A.8 in the Appendix [15]. All these raw datasets used in this work can be acquired from the public weather station directory service hosted by the Bureau of Meteorology (BOM) of Australia [15].

This study is focused on the months June, July, and August of winter [16,17]. Minute-wise climate data of these winter months in years 2016 and 2017 are extracted from the 30 weather stations in the study area. After extracting these data, they are further processed to be prepared for model construction and testing. These procedures are discussed in the next subsection.

2.2. Data preprocessing

The raw data needs to be preprocessed before model construction. After extracting the required columns (Timestamp, Air Temperature, Dew Point, Relative Humidity, Wind Speed, Wind Direction) from the raw datasets, seven more steps of data preprocessing are performed. These seven steps belong to two phases. In phase 1, the raw data is processed to transform useful features and handle empty values. The outputs of phase 1 are saved for reuse in the next phase. In phase 2, the outputs from phase 1 are inputted and modified to fit a specific target model and model setting. As an example, Fig. 2 outlines the changes of data structure during data preprocessing, using an RNN with a sequence length of 20 as an example. The red figures indicate a change to that data in that step.

Phase 1 consists of five steps and starts with transforming wind speed and direction into features of "N-wind" and "E-wind". These two new features represent wind speeds towards the north (N-wind) and east (E-wind) directions. This transformation is done to prevent the possible errors generated by the wind direction values (0–359) [18]. The original wind direction data from BOM is the bearing of the direction the wind is originated [19]. Therefore, to begin the conversion, the original wind direction (met) is changed to the wind blowing direction (deg) by reversing the direction (Eq. (1)).

$$deg = \begin{cases} met + 180^\circ, & \text{if } met < 180^\circ \\ met - 180^\circ, & \text{if } met \geq 180^\circ \end{cases} \tag{1}$$

Then, with the wind speed (v) and the wind blowing direction (deg), magnitudes of wind vector towards north (v_N) and east (v_E) are obtained through Eq. (2) [18].

$$v_E, v_N = v \times \sin(deg), v \times \cos(deg) \tag{2}$$

In the second step of data preprocessing, the minimum temperature is constructed as the prediction target. The minimum temperature is simply constructed by getting the minimum temperature value in the next 60 min. As the recording of data is done once per minute, the minimum temperature of the current data entry is the minimum value of the next 60 temperature values.

To fit the data structure for RNN, LSTM, and GRU models, sequences of data are created in step 3. A sequence contains all the features (Air Temperature, Dew Point, Relative Humidity, Wind Speed, Wind Direction). For every data entry, the current sequence is defined as

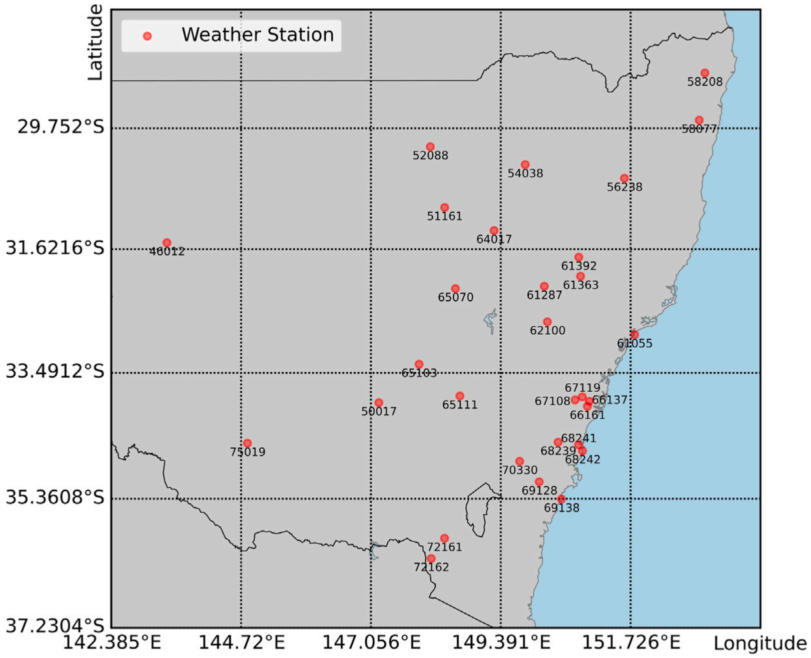


Fig. 1. Weather stations with ID.

sequence t, the sequence from one minute before is added to the current entry and defined as sequence $t-1$. Similarly, the sequence from two minutes before is added and defined as sequence $t-2$. Since the maximum sequence length of the experiments is 120, sequences from previous entries are added to the current entry from one minute before until 119 min before. Therefore, each entry includes 120 sequences from sequence $t-119$ to sequence t.

The timestamp column is a tool to help extract the data from the desired time period. Now, as all of the features and sequences are generated, the timestamp column can be removed in step 4. Then, in the final step of phase 1, the listwise deletion data imputation technique is applied to eliminate data entries with empty features [20]. This also includes the removal of data sequences with missing features to preserve the time differences between observations. The final product of phase 1 is output to hard disk to be used in phase 2. For every weather station dataset, the steps in phase 1 are conducted for data in years 2016 and 2017.

In phase 2 of data preprocessing, the results from phase 1 are transformed to the required form for different models with different settings. After reading an output from phase 1, step 6 of data preprocessing removes excess data sequences. For ANN models, only one sequence is required. Therefore, sequences $t-119$ to $t-2$ should be removed. For RNN, LSTM, and GRU models, sequences are removed according to the sequence length of the target model. For example, with a desired sequence length of 20, sequences $t-119$ to $t-20$ are removed, leaving only 20 data sequences (Fig. 2).

Step 7 of data preprocessing is only executed to prepare the data for RNNs and their variants. Every entry in the dataset is converted to a 2D structure. Each row of this 2D structure represents a sequence of a specific time. The rows are structured top to bottom from an earlier time to a more recent time. At this stage, the data can proceed to model construction.

Table 1 describes the features at the end of data preprocessing. Most features demonstrate a normal distribution. The relative humidity is the only skewed-left feature. Table 2 shows the Pearson correlation matrix of the processed features. Dew point and relative humidity show strong correlations to temperature. Other features are relatively independent of each other.

2.3. Model construction and testing

In this subsection, information of model construction and testing is provided. The computation environment is firstly described, followed by the usage of datasets. Next, the model structures and hyperparameters are clarified. Finally, the two stages of model construction are revealed.

All models in this article are constructed on a desktop computer using an Intel i7-8700K 3.70 GHz processor, equipped with 32 GB RAM. The graphics processing unit is a Nvidia RTX 2080 graphics card. The deep learning framework for model construction and testing is TensorFlow 2.3.0.

The data preprocessing processes generate two datasets from 2016 and 2017 for every weather station. In the experiment scene, models should be built in the current year and deployed in the next year. The year 2016 has been set as the “current” year and 2017 is the “next” year. Therefore, datasets from 2016 are used for model construction and datasets from 2017 are used for final testing.

It is assumed that during the process of model construction only the “current” year data is available. The datasets from year 2016 are split for model training, validation, and testing. For every dataset from different weather stations, 80% of the data are randomly allocated to the training dataset and the other 20% to the testing dataset. From the training dataset, a further split of 20% of the data forms the validation dataset. The training dataset, which contains most of the data, is used to fit the parameters of the models. During the training process, the validation dataset helps to tune the hyperparameters. After all training, hyperparameter tuning, and validation are completed, test predictions are conducted by the model with the testing dataset. This provides an error metric of the model. However, as the model should be deployed in the “next” year, datasets from 2017 are used as extra and final testing datasets. In the experiments, the errors generated from the 2017 testing datasets are also compared.

The model structure for all models is defined as a three-layer model. The first layer has five neurons. The second has seven and the output layer has one neuron. The first two layers change according to the target model. For example, if it is an ANN model, the cells in these two layers are ANN cells with Rectified Linear Unit (ReLU) activation functions. For RNN and its variants, the cells use the tanh activation

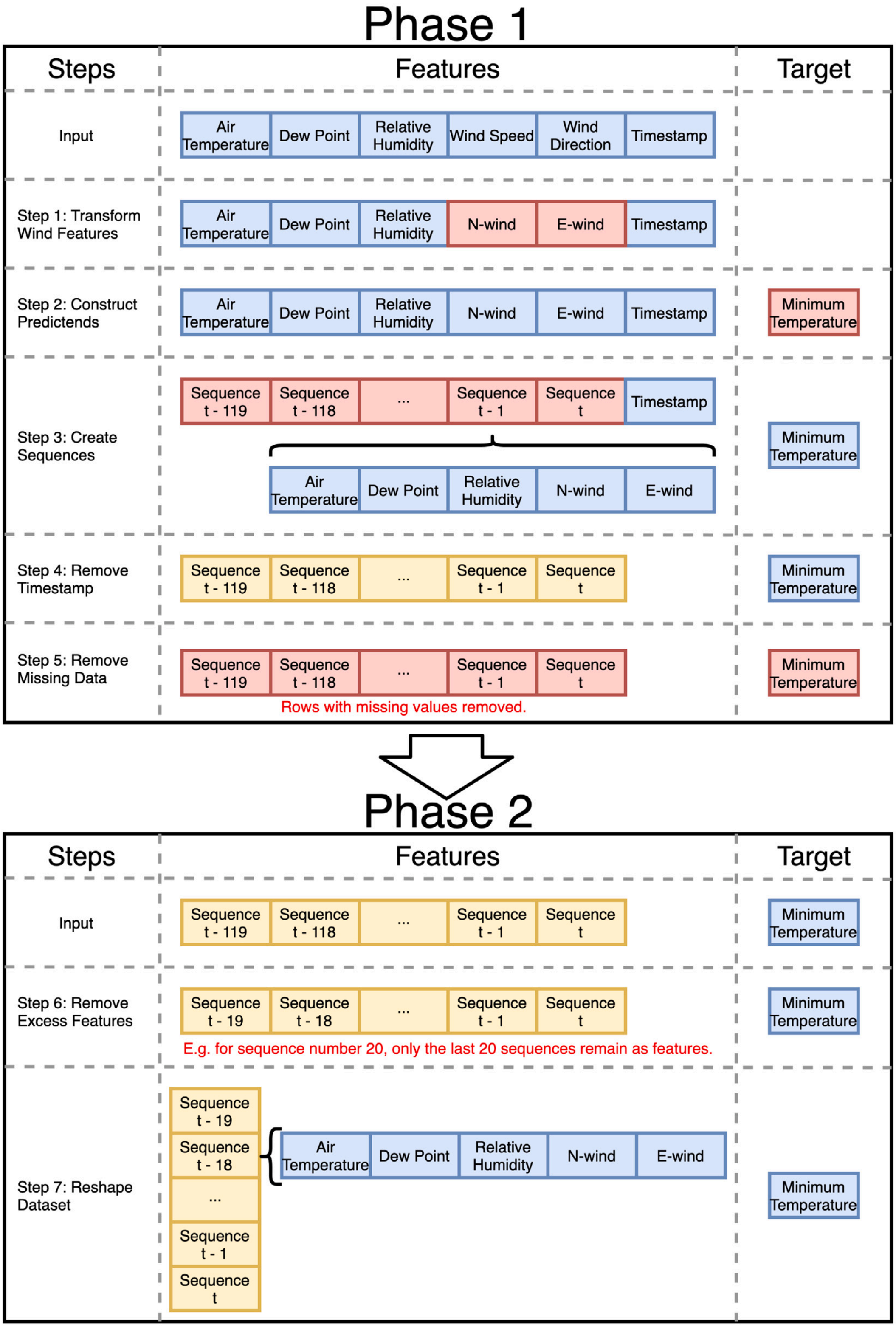


Fig. 2. Data preprocessing steps. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

functions with relevant cells to the models. Also, for the first layer of RNN, LSTM, and GRU models, the hidden state of the cells is output as sequential inputs of the second layer. The third layer only consists of a single linear cell to output the result.

Adam is the optimizer used during training. Learning rate, β_1, β_2, and ϵ, are the hyperparameters required for Adam [21]. In the experiments, learning rate is set to the "good default settings" as 0.001, β_1 as 0.9, β_2 as 0.999, and ϵ as 10^{-7}. Another hyperparameter, batch size is configured to 64. With these settings, all models are trained for 100 epochs with a Mean-Square Error (MSE) loss function.

Models are constructed in two groups. The groups are separated by different model structures and settings. In the first group, one ANN model is constructed for each of the 30 weather stations. In the second group, RNN, LSTM, and GRU models with different sequence lengths

Table 1
Description of the processed training features.

Item	Feature				
	Temperature (°C)	Dew point (°C)	Relative humidity (%)	N-wind (km/h)	E-wind (km/h)
Min	−5.4000	−37.8000	2.0000	−86.2608	−78.0689
Mean	10.4629	6.3262	78.1665	−1.9071	4.8393
Max	31.0000	23.2000	100.0000	81.8329	95.3837
Std	5.0328	4.2037	18.0907	10.7936	11.3289
Distribution	Normal	Normal	Skewed-left	Normal	Normal

Table 2
Pearson correlation matrix of the processed training features.

	Temperature	Dew point	Relative humidity	N-wind	E-wind
Temperature	1.0000	0.6391	−0.5744	−0.1330	0.0181
Dew point	0.6391	1.0000	0.2474	−0.0946	−0.1613
Relative humidity	−0.5744	0.2474	1.0000	0.0659	−0.2059
N-wind	−0.1330	−0.0946	0.0659	1.0000	−0.1814
E-wind	0.0181	−0.1613	−0.2059	−0.1814	1.0000

are trained for all 30 weather stations. There are 6 different sequence length settings (20, 40, 60, 80, 100, 120). Altogether, the second group outputs 540 models. Overall, there are 570 models constructed for this article. The usage of these models in experiments is explained in the next subsection.

2.4. *Experiments*

There are three experiments conducted to test the model error and performance of models. In the first experiment, the errors of different model types (ANN, RNN, LSTM, GRU) are compared. ANN models from the first model group are the baseline of this experiment. For each weather station, there are eight conducted tests to measure the losses from the current year and next year data for the four model types. RNN models and their variants are tested with a sequence length of 120. Then, in the second experiment, the effect of the sequence length of RNN, LSTM, and GRU models are compared. Additional to the results obtained in the first experiment, 30 more tests are conducted for each weather station to obtain the results of the three RNN-based model types with five sequence number settings, and tested with the current year and next year testing datasets. Model errors of all RNN-based model types are compared with different sequence length settings against the ANN baseline. Similarly, in the final experiment, the training time and inference time of different sequence length settings are compared against the ANN baseline.

3. Results and discussions

To validate and compare different model settings for next hour frost prediction models, three experiments are conducted. The first experiment compares the MSE between ANN and RNN-based model types. Then, in the second experiment, MSEs of RNN-based models with different sequence lengths are assessed. Finally, processing time related factors are also analyzed with different sequence lengths. This provides an overview of different models' real-time computation abilities.

3.1. *Model error*

In this experiment, the model errors of RNN-based models with a sequence length of 120 are evaluated with ANN models. Fig. 3 shows that when testing with testing datasets derived from the same year when the training datasets is collected, LSTM seems to perform with the best accuracy with the lowest MSE loss. LSTM is also the only RNN-based model type to exceed the accuracy of ANN models. This result is also confirmed with one-sided paired T-tests. From the p-values (RNN: 0.1544; LSTM: 6.1225e−12; GRU: 0.2644), LSTM is the only model type that the p-value is smaller than the 0.05 α value. This means the null hypothesis is rejected and ANN is likely to produce outputs with a greater loss than LSTM. LSTM models have the highest accuracy among other RNN-based models due to the extra gates to memorize sequence patterns [5].

An assumption made for the models of this paper is that the models are constructed using data from "this" year and deployed in the "next" year. Therefore, models are also tested with testing datasets obtained one year after the training datasets. The results are significantly different compare to the results from the "current" year testing datasets (Fig. 3). ANN shows the lowest MSE loss, which indicates the highest accuracy. On the other hand, LSTM models have the lowest accuracy. The results of the one-sided paired T-tests demonstrate that LSTM and GRU models have p-values (RNN: 0.1350; LSTM: 0432; GRU: 0.0027) less than the α threshold. Therefore, it is likely that LSTM and GRU models all have a significantly higher loss (less accurate) than ANN models.

As all models are trained with the current year data, LSTM models with more parameters and gates [22] fit closer to the current year testing datasets. On the other hand, RNN-based models are constructed through learning the sequence patterns [4,5]. Thus, these models are sensitive to the change of sequence patterns. In [23,24], the global climate change induces an increase of instability in weather patterns over time. As a result, the accuracy of RNN-based models deteriorates when tested with the next year testing datasets. Compared to the baseline, LSTM and GRU models with more parameters [22] tend to "overfit" more to the current year pattern and are vulnerable to the changed next year pattern. However, the exact extent of accuracy reduction is unknown as the change of climate patterns in the future is also unknown.

3.2. *Effect of sequence length on model error*

In this experiment, the effect of sequence length on model error for RNN-based models is inspected. Fig. 4 shows the average MSEs of RNN-based models tested by current year datasets. Overall, the increase of sequence length does not reduce the average loss. Only LSTM shows a decreasing trend of losses with the increase of sequence length. However, this change is not very significant. When compared to ANN, LSTM and GRU with some settings (sequence length = 20, 40, 80, 100) seem to have a smaller loss than ANN (0.4550). This is confirmed by the one-sided paired T-tests results on Table 3. Only p-values of LSTM, and GRU models with the sequence lengths of 20, 40, 80, and 100 are smaller than the α. This reveals that when testing with the current year datasets. This demonstrates the higher likelihood that LSTM and GRU (sequence length = 20, 40, 80, 100) models are more accurate than ANN models when tested with current year testing datasets.

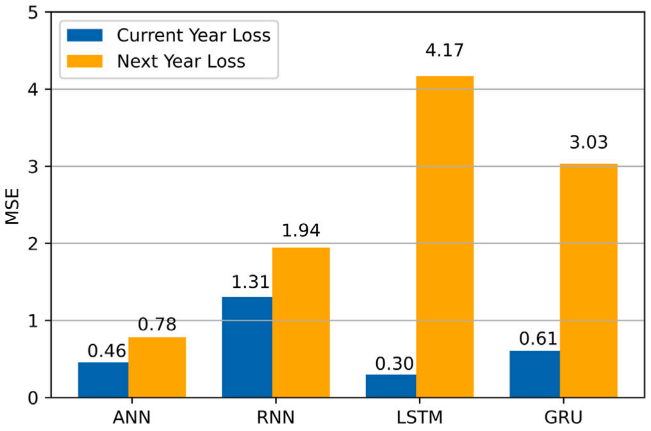


Fig. 3. Average MSE tested with current and next year datasets.

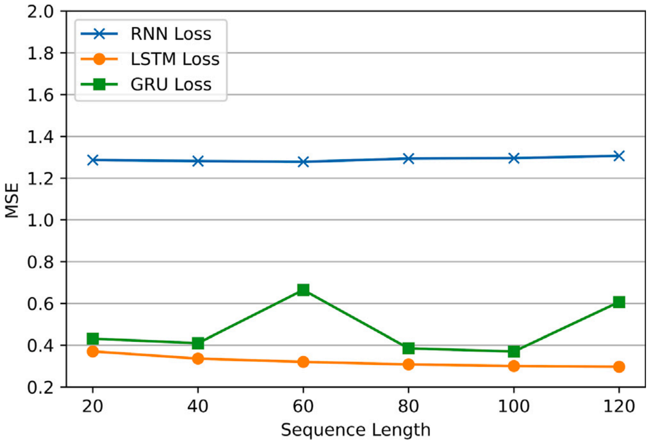


Fig. 4. Average MSE tested with current year datasets for different sequence lengths.

Table 3
P-values of comparing average MSEs between ANN and RNN-based models with different model sequence lengths (current year).

Sequence length	Model type		
	RNN	LSTM	GRU
20	0.1603	3.4979e−10	0.0032
40	0.1617	6.9911e−11	1.3249e−5
60	0.1628	1.0544e−10	0.1591
80	0.1581	3.0000e−11	4.2542e−7
100	0.1576	2.1009e−11	4.2930e−7
120	0.1544	6.1225e−12	0.2644

Fig. 5 is the average MSEs for RNN-based models with different sequence lengths tested by the next year testing datasets. Similar to the reverse of results in Experiment 1, all RNN-based models have a higher loss than the ANN MSE (0.7813). Table 4 is the p-values obtained from one-sided paired T-tests with an alternative hypothesis that each tested model has a greater MSE than the ANN baseline. The alternative hypothesis is in favor of LSTM and GRU models as their p-values are less than α. This means LSTM and GRU models are likely to perform with higher errors than ANN models in the next year. Also, as explained in Experiment 1, the change of climate patterns in the future is unknown. This could be the reason of the additional noise in Fig. 5, compared to Fig. 4.

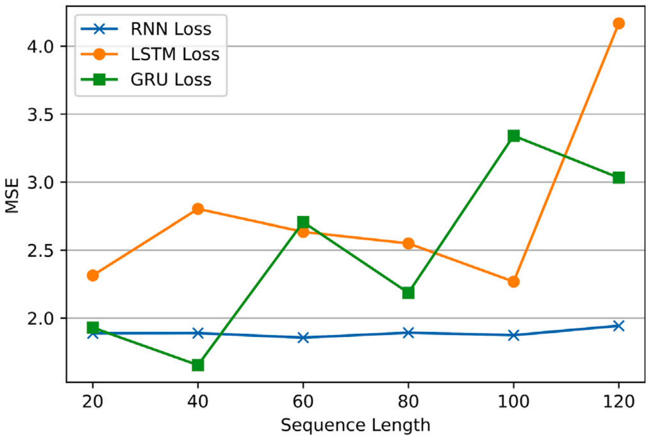


Fig. 5. Average MSE tested with next year datasets for different sequence lengths.

Table 4
P-values of comparing average MSEs between ANN and RNN-based models with different model sequence lengths (next year).

Sequence length	Model type		
	RNN	LSTM	GRU
20	0.1466	0.0013	0.0109
40	0.1463	0.0040	0.0249
60	0.1538	0.0028	0.0156
80	0.1457	0.0072	0.0089
100	0.1497	0.0048	0.0294
120	0.1350	0.0433	0.0028

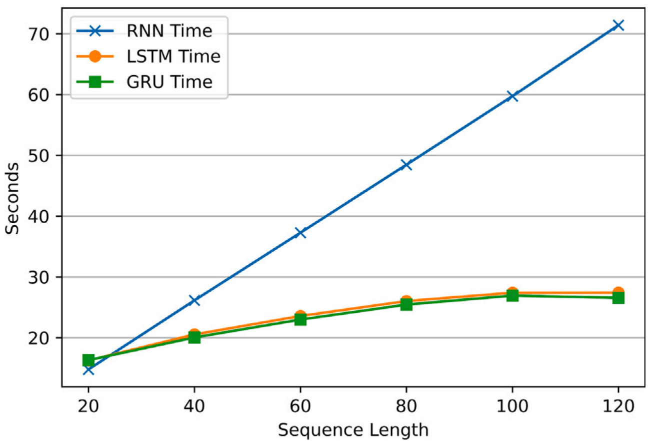


Fig. 6. Average training time per epoch for different sequence lengths.

3.3. Effect of sequence length on processing time

In the final experiment, the training time per epoch and inference time per input are tested and compared between ANN models and RNN-based models with different sequence lengths. Fig. 6 demonstrates that the training time per epoch for RNN, LSTM, and GRU models increases as the sequence length increases. This is due to the additional parameters for training as the sequence length increases. LSTM and GRU models have overlapping curves of training times. RNN training times are greater than the prior two models. This is because, in TensorFlow 2.3.0, only LSTM and GRU model training processes are optimized by cuDNN for faster training speeds [25,26]. However, whether optimized or not, compared to the ANN training time of 1.070 s, as the p-values are smaller than the 0.05 α, Table 5 indicates that the training time of ANN models per epoch is more likely to be smaller than all settings of RNN-based models.

Table 5
P-values of comparing average training time per epoch between ANN and RNN-based models with different model sequence lengths.

	Model type		
Sequence length	RNN	LSTM	GRU
20	5.3068e−32	1.4421e−30	7.5427e−31
40	1.4240e−31	1.5839e−31	2.1966e−31
60	1.5350e−31	2.2799e−32	1.3996e−31
80	4.3655e−32	5.7483e−32	1.4185e−31
100	7.6250e−32	2.8775e−31	6.2908e−31
120	1.2237e−31	5.4545e−32	2.6502e−31

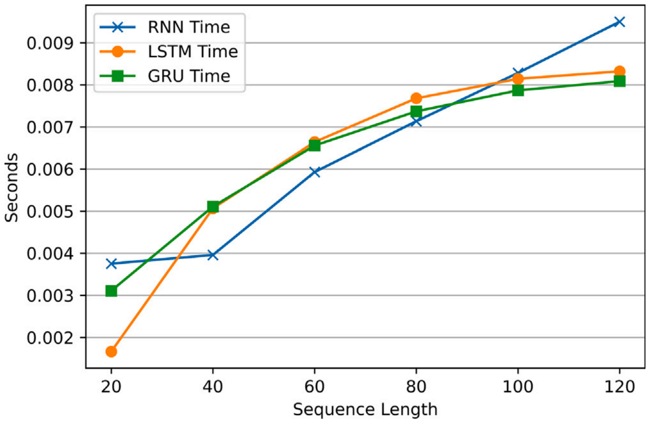


Fig. 7. Average inference time per input for different sequence lengths.

Table 6
P-values of comparing average inference time per input between ANN and RNN-based models with different model sequence lengths.

	Model type		
Sequence length	RNN	LSTM	GRU
20	2.5574e−45	1.0941e−17	1.7544e−35
40	5.0297e−49	2.1683e−37	2.6969e−42
60	6.7674e−62	3.7983e−42	2.6171e−42
80	2.1246e−60	1.0625e−44	1.5319e−42
100	1.9803e−54	2.7216e−46	1.2132e−37
120	1.0382e−58	5.9795e−48	1.6906e−46

Fig. 7 demonstrates the average inference time per input for RNN-based models with different sequence lengths. There is a trend of increase in inference time, along with the increase of sequence length. Similar to training time, a higher sequence length of RNN-based models implies a larger input sequence and a model structure with more parameters. Thus, the inference time increases with the sequence length. Also, the inference time for all RNN-based settings is significantly larger than ANN inference time (4.5214e−4 s). This statement is supported with the one-sided paired T-test results as all p-values are less than the α (Table 6).

3.4. Limitations and open challenges

Experiment 1 shows that RNN-based models have lower model errors than ANN models when tested with current year datasets. LSTM models have the lowest errors and highest accuracy. However, RNN-based models' accuracy declines and is exceeded by ANN when tested with next year datasets. This decline is likely to be caused by climate pattern change over time [4,5,23,24]. Also, the models in this article are constructed with one year of data. From [27], RNN models often require data from more years to fully learn the seasonality patterns. However, this would increase the dependency on historical data, which is a limitation mentioned in the next subsection. With both sources of accuracy deterioration, RNN-based models are only suitable for a model deployed in the short term. In Experiment 2, RNN-based models do not present a significant reduction in model error as the sequence length increases. Therefore, considering the results from Experiment 3, RNN-based models with a short sequence can be deployed for short times to provide predictions with higher accuracy and performance. On the other hand, ANN models can be deployed over longer time spans without much accuracy deterioration. Both training and inference time is significantly lower than RNN-based models. Overall, ANNs might still be more suitable than RNN-based models for frost prediction in a long-term scenario with minimal system maintenance and update because of their higher accuracy and performance over the long term. After constructing the above different frost prediction models and analyzing their performance, the limitations presented in this subsection are discovered. Limitations create new challenges and lead towards future directions to frost prediction research.

3.4.1. Model accuracy requirements are not specified

Different plants have different frost tolerance and sensitivity [8]. Therefore, model accuracy requirements may vary between different plants. There are many studies on plant frost tolerance. However, the sensitivity of plants to individual frost factors is not fully revealed [28]. As a future direction, the sensitivity to different frost factors such as temperature, humidity, dew point, cloud coverage, solar radiation and wind speed should be tested with high precision for individual species of plants in controlled environments. With enough plant species studied, a model accuracy threshold can be set for frost prediction models.

3.4.2. Lack of standard datasets

The models and experiment results are obtained from a public dataset that represents local climate conditions of our study sites in Australia. Therefore, the accuracy of the models at other sites with different climate patterns is questionable. To the best of our knowledge, prior research has been based upon locally obtained private datasets [9–14]. As a result, model results could be biased. There is a need of a standard dataset for frost prediction models that includes data entries from different locations.

3.4.3. The RNN input data format affects the system energy efficiency

The RNN-based models in this article require a sequence of climate data as model inputs. Each sequence of data contains sensor readings collected every minute over a long time span. This places a restriction on node duty cycles. Sensor readings must be collected per minute to satisfy the model input. For common agricultural IoT systems, sensor readings are transmitted through the radio from a node to a central processing node for further processing and data analytics [29]. In [30], radio power consumption increases as the time interval of radio transmission reports decreases. Even radio transmissions between 10-minute periods consume a significant amount of the system energy [30]. Therefore, if sensor readings are reported every minute, higher energy consumption will be placed on the whole system. This issue could be mitigated by aggregating a few minute-wise sensor readings into one radio transmission. Also, inference on edge with edge computing can also reduce the number of radio transmissions. A model can be deployed on edge devices and only transmit the predicted outcome when it triggers a preset condition. However, even there are mitigation plans, RNN-based models still limit the design of IoT systems with potential high energy impact. On the other hand, each ANN model inference only requires a single set of climate data as input. ANN-based frost prediction models can be applied to systems with different time intervals to obtain sensor data. Radio transmission intervals can be adjusted according to system requirements. Therefore, ANN models may still be more suitable for frost prediction systems.

3.4.4. Models depend on previous year data

All models constructed for this article depend on previous years of data. Therefore, model accuracy is dependent on the quality of historical data. This is a limitation on all machine learning models [31]. As more recent researches are site-specific, sites without any record of historical climate data require an IoT data collection system to be deployed for data collection. To ensure similar performance to the results of this article, the system must be deployed at least a year prior to produce a prediction model and provide frost prediction services. This increases the deployment time of the system. A possible solution is to explore the performance of models based on previous months of data. This solution could substantially reduce development time. However, it cannot eliminate this excess system development time for data collection. Methods eliminating the requirement of on-site historical data need to be developed. Another possible direction is to explore the generalization or transfer of models to similar locations.

3.4.5. Lack of stopping conditions

The frost prediction models constructed in this article only predict the start of a frost event with real-time sensor readings. This prediction could be the trigger of a frost protection mechanism. However, there are limited mentions of predicting the end of a frost event to switch off any protection mechanisms. As most frost protection systems rely only on a single sensor node [3], activation of a nearby protection mechanism might affect the sensor readings and contaminate the prediction outcomes of frost prediction models. Therefore, future prediction models could be developed to eliminate the effect of frost protection mechanisms. As a possible benefit, frost protection mechanisms could be switched off earlier to reduce the operational cost.

4. Conclusion

The primary aim is to increase the prediction frequency from once per 12–24 h for the next day or night events to minute-wise predictions for the next hour events. RNN-based models are selected to learn the sequence pattern of historical data. ANN models are used as a baseline. Datasets from weather stations in the NSW and ACT areas of Australia are obtained. These datasets are recorded during the years 2016 and 2017. With these datasets, it is assumed that our models are built during the year 2016 (current year) and deployed in year 2017 (next year). Therefore, datasets from 2016 are used for model construction and preliminary testing. Datasets from 2017 are used for final testing. After constructing the models, there are three experiments testing the model errors, also the effect of sequence lengths on errors and processing time for RNN-based models. The errors of models is tested with both the current and next year datasets. LSTM seems to have the highest accuracy when tested with the current year testing datasets. However, the accuracy for all RNN-based models reduces when tested with the next year testing datasets. ANN models have the highest accuracy with the next year testing datasets. When testing RNN-based models with different sequence lengths, it seems that sequence lengths cannot affect the accuracy of models significantly. However, training and inference time increases with the sequence length. Therefore, RNN-based models should be used for short-term deployments with a shorter sequence length to ensure accuracy and performance. On the other hand, ANN models demonstrate the lowest error when tested with next year datasets. Also, the training and inference speeds of ANN models are faster than RNN-based models. Therefore, in the long term, ANN models are more suitable than RNN-based models due to better accuracy and performance.

There are limitations determined. Firstly, the model accuracy requirements are not specified in this article due to a lack of studies on precise frost sensitivities to individual frost factors. Secondly, the current model and most previous models are constructed with local data. The lack of standard datasets limits unbiased comparisons between different models. Thirdly, the RNN input data format affects the system energy efficiency. The limitations on energy efficiency could be avoided by the application of edge computing or message aggregation. On the other hand, ANN models do not have such limitations. The next limitation is related to the construction of models. All of these models depend on previous year data. For sites without data records, this limitation requires extra development and deployment time. Therefore, models that decouple from historical data could be developed to eliminate this restriction completely. The final limitation is that prediction models lack stopping conditions. As most systems read climate data from one sensor node, the activation of protection gear could contaminate the input data. Thus, it contaminates the prediction outcomes. A future model could be developed to remove this effect of data contamination and provide a model-based automatic stopping mechanism to reduce operational costs of frost protection systems.

CRediT authorship contribution statement

Ian Zhou: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Justin Lipman:** Conceptualization, Supervision, Writing – review & editing. **Mehran Abolhasan:** Methodology, Validation, Writing – review & editing. **Negin Shariati:** Resources, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

We would like to acknowledge the support of Food Agility CRC Ltd, 81 Broadway, Ultimo, NSW, 2007, Australia, funded under the Commonwealth Government CRC Program. The CRC Program supports industry-led collaborations between industry, researchers and the community.

The authors are with Radio Frequency and Communication Technologies (RFCT) research laboratory, Faculty of Engineering and IT, University of Technology Sydney, Ultimo, NSW 2007, Australia.

Appendix. Weather station location coordinates

See Tables A.7 and A.8.

Table A.7
Weather station location coordinates.

Station ID	Latitude (degrees)	Longitude (degrees)
69138	−35.3635	150.4828
62100	−32.7244	150.2290
75019	−34.5412	144.8345
61363	−32.0335	150.8264
46012	−31.5194	143.3850
67119	−33.8510	150.8567
50017	−33.9382	147.1962
54038	−30.3154	149.8302
72162	−36.2304	148.1405
58208	−28.8824	153.0618
67108	−33.8969	150.7281
61287	−32.1852	150.1737
65070	−32.2206	148.5753
68242	−34.6532	150.8609
61392	−31.7416	150.7937

Table A.8
Weather station location coordinates continue.

Station ID	Latitude (degrees)	Longitude (degrees)
72161	−35.9371	148.3779
51161	−30.9776	148.3798
70330	−34.8085	149.7311
61055	−32.9184	151.7985
66137	−33.9176	150.9837
64017	−31.3330	149.2699
68239	−34.5253	150.4217
66161	−33.9925	150.9489
68241	−34.5638	150.7900
58077	−29.6224	152.9605
69128	−35.1103	150.0826
65111	−33.8382	148.6540
56238	−30.5273	151.6158
65103	−33.3627	147.9205
52088	−30.0372	148.1223

References

[1] Webb L, Snyder RL. Frost hazard. In: Bobrowsky PT, editor. Encyclopedia of natural hazards. Dordrecht: Springer Netherlands; 2013, p. 363–6. http://dx.doi.org/10.1007/978-1-4020-4399-4_148.

[2] Ma Q, Huang J-G, Hänninen H, Berninger F. Divergent trends in the risk of spring frost damage to trees in Europe with recent warming. Global Change Biol 2019;25(1):351–60. http://dx.doi.org/10.1111/gcb.14479.

[3] Zhou I, Lipman J, Abolhasan M, Shariati N, Lamb DW. Frost monitoring cyber–physical system: A survey on prediction and active protection methods. IEEE Internet Things J 2020;7(7):6514–27. http://dx.doi.org/10.1109/JIOT.2020.2972936.

[4] Goodfellow I, Bengio Y, Courville A. Sequence modeling: Recurrent and recursive nets. In: Deep learning. MIT Press; 2016, p. 363–408.

[5] Salehinejad H, Sankar S, Barfett J, Colak E, Valaee S. Recent advances in recurrent neural networks. 2017, arXiv preprint arXiv:1801.01078.

[6] Farooq MS, Riaz S, Abid A, Abid K, Naeem MA. A survey on the role of IoT in agriculture for the implementation of smart farming. IEEE Access 2019;7:156237–71. http://dx.doi.org/10.1109/ACCESS.2019.2949703.

[7] Sarker VK, Gia TN, Tenhunen H, Westerlund T. Lightweight security algorithms for resource-constrained IoT-based sensor nodes. In: ICC 2020 - 2020 IEEE international conference on communications. 2020. p. 1–7. Dublin, Ireland. http://dx.doi.org/10.1109/ICC40277.2020.9149359.

[8] di Francescantonio D, Villagra M, Goldstein G, Campanello PI. Drought and frost resistance vary between evergreen and deciduous Atlantic forest canopy trees. Funct Plant Biol 2020. http://dx.doi.org/10.1071/FP19282.

[9] Halley V, Eriksson M, Nunez M. Frost prevention and prediction of temperatures and cooling rates using GIS. Aust Geogr Stud 2003;41(3):287–302. http://dx.doi.org/10.1046/j.1467-8470.2003.00235.x.

[10] Iacono LE, Vázquez Poletti JL, García Garino C, Llorente IM. Performance models for frost prediction in public cloud infrastructures. Comput Inform 2018;37(4):815–37.

[11] Diedrichs AL, Bromberg F, Dujovne D, Brun-Laguna K, Watteyne T. Prediction of frost events using machine learning and IoT sensing devices. IEEE Internet Things J 2018;5(6):4589–97. http://dx.doi.org/10.1109/JIOT.2018.2867333.

[12] Ghielmi L, Eccel E. Descriptive models and artificial neural networks for spring frost prediction in an agricultural mountain area. Comput Electron Agric 2006;54(2):101–14. http://dx.doi.org/10.1016/j.compag.2006.09.001.

[13] Zeng W, Zhang Z, Gao C. A Levenberg-Marquardt neural network model with rough set for protecting citrus from frost damage. In: Proc. 2012 eighth international conference on semantics, knowledge and grids. 2012. p. 193–196. Beijing, China. http://dx.doi.org/10.1109/SKG.2012.4.

[14] Fuentes M, Campos C, García-Loyola S. Application of artificial neural networks to frost detection in central Chile using the next day minimum air temperature forecast. Chil J Agric Res 2018;78(3):327–38. http://dx.doi.org/10.4067/S0718-58392018000300327.

[15] of Meteorology B. Weather Station Directory. Bureau of Meteorology; 2020, URL http://www.bom.gov.au/climate/data/stations/. [Accessed 04 April 202].

[16] of Meteorology B. Australia in winter 2016. 2016, URL http://www.bom.gov.au/climate/current/season/aus/archive/201608.summary.shtml. [Accessed 04 April 2020].

[17] of Meteorology B. Australia in winter 2017. Bureau of Meteorology; 2017, URL http://www.bom.gov.au/climate/current/season/aus/archive/201708.summary.shtml. [Accessed 04 April 2020].

[18] Lee M, Moon S, Kim Y, Moon B. Correcting abnormalities in meteorological data by machine learning. In: 2014 IEEE international conference on systems, man, and cybernetics. San Diego, CA, USA; 2014, p. 888–93. http://dx.doi.org/10.1109/SMC.2014.6974024.

[19] of Meteorology B. Wind. Bureau of Meteorology; 2020, URL http://www.bom.gov.au/marine/knowledge-centre/reference/wind.shtml. [Accessed 04 April 2020].

[20] Osborne JW. Six: Dealing with missing or incomplete data: Debunking the myth of emptiness. In: Best practices in data cleaning: A complete guide to everything you need to do before and after collecting your data. SAGE Publications, Inc; 2013, p. 105–38. http://dx.doi.org/10.4135/9781452269948.n6.

[21] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2017, arXiv preprint arXiv:1412.6980.

[22] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. 2014, arXiv preprint arXiv:1412.3555.

[23] Augspurger CK. Reconstructing patterns of temperature, phenology, and frost damage over 124 years: Spring damage risk is increasing. Ecology 2013;94(1):41–50. http://dx.doi.org/10.1890/12-0200.1.

[24] Hänninen H. Does climatic warming increase the risk of frost damage in northern trees? Plant Cell Environ 1991;14(5):449–54. http://dx.doi.org/10.1111/j.1365-3040.1991.tb01514.x.

[25] Google. tf.keras.layers.LSTM. Google; 2020, URL https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM. [Accessed 12 October 2020].

[26] Google. tf.keras.layers.GRU. Google; 2020, URL https://www.tensorflow.org/api_docs/python/tf/keras/layers/GRU. [Accessed 12 October 2020].

[27] Muzaffar S, Afshari A. Short-term load forecasts using LSTM networks. Energy Procedia 2019;158:2922–7. http://dx.doi.org/10.1016/j.egypro.2019.01.952, Innovative Solutions for Energy Transitions.

[28] Barlow KM, Christy BP, O'Leary GJ, Riffkin PA, Nuttall JG. Simulating the impact of extreme heat and frost events on wheat crop production: A review. Field Crops Res 2015;171:109–19. http://dx.doi.org/10.1016/j.fcr.2014.11.010.

[29] Deepika G, Rajapirian P. Wireless sensor network in precision agriculture: A survey. In: 2016 international conference on emerging trends in engineering, technology and science. 2016. p. 1–4. Pudukkottai, India. http://dx.doi.org/10.1109/ICETETS.2016.7603070.

[30] Martinez B, Montón M, Vilajosana I, Prades JD. The power of models: Modeling power consumption for IoT devices. IEEE Sens J 2015;15(10):5777–89. http://dx.doi.org/10.1109/JSEN.2015.2445094.

[31] Mohammadi M, Al-Fuqaha A, Sorour S, Guizani M. Deep learning for IoT big data and streaming analytics: A survey. IEEE Commun Surv Tutor 2018;20(4):2923–60. http://dx.doi.org/10.1109/COMST.2018.2844341.