Forecasting microclimates with a low-cost LoRa enabled weather station network

1st Adam Sidnell
Computer Science MSc
University of Bristol
Bristol, United Kingdom
adam.sidnell@gmail.com

2nd Ruzanna Chitchyan
Professor in Computer Science
University of Bristol
Bristol, United Kingdom
r.chitchyan@bristol.ac.uk

***Abstract*—General weather forecasts are often too broad to accurately capture the small climatic variation that exists on farms. This can lead to sub-optimal decisions on tasks such as crop spraying, frost prevention or irrigation. While commercial weather stations exist, there is often a trade-off between their cost and transmission range. Additionally, these hardware products do not attempt to use their data to offer farmers improved forecasting on their fields. This paper presents the prototype design and evaluation for Agriscanner: An end-to-end, low-cost IoT (Internet of Things) weather station network that incorporates machine learning (LightGBM) to predict future weather within a field. Data is streamed to a web application for live visualisation. Evaluation shows the hardware system achieves a 1,200m range. With the use of a repeater node, this allows for an effective range of 2,400m in non-ideal conditions. This was achieved at a cost of £520 for the complete system (including two sensor nodes, a repeater, and a WiFi gateway) while outperforming commercial alternatives in range and cost. The machine learning model showed mixed results, outperforming general forecasts for certain climate factors (wind-speed) but highlighting the need for a larger, more diverse training dataset.**

***Index Terms*—machine learning, smart farming, forecast, microclimate, IoT, LoRa, LightGBM**

I. Introduction

Accurate weather data is critical for decision making in agriculture. Farmers rely on forecasts for various daily farming functions with significant financial and environmental costs attached. For example, it is required by law that pesticide spraying must be done in low wind speed to prevent spray drift which can damage nearby ecosystems [1]. Similarly, spring frosts pose a significant risk to crop yields; a single frost night can reduce crop yields by as much as 24% [2], highlighting the need for accurate prediction that allows farmers to take appropriate precautions.

However, weather forecasts, as provided by the MetOffice in the UK, are based on macro-scale models which can only give a more general view of weather in a region and not smaller distinct climate variations that exist within this, known as microclimates. The models used by weather forecasts are not suitable for microclimate prediction as they are designed for wide area predictions based on planetary movements of wind and moisture.

This project is funded by the UK Research Institute

Farmers rank the accuracy and reliability of weather forecasts as one of the largest limiting factors when making decisions in growing season [3]. Microclimates can occupy an area of anywhere from less than a metre across to several hundred metres [4], thus multiple different microclimates can exist within a single field. This need for hyper-local data was confirmed in preliminary interviews; one farmer noted that a single orchard, being situated on a large slope, experienced significantly different wind speeds on one side compared to the other.

To address this issue, this paper presents a complete end-to-end system called Agriscanner. While commercial IoT weather stations exist, simply viewing current and historical data is insufficient for effective planning. The primary contribution of this work is a deployable system that applies an open-source machine learning procedure using locally collected sensor data to forecast the microclimate.

This approach is distinct from related local forecasting work in three key areas. First, unlike studies relying on public datasets or existing sensor networks, we designed and built the entire low-cost hardware network required to collect data from remote fields. Second, while other systems focus on complex deep learning models, we demonstrate the application of an efficient, tree-based machine learning model (LightGBM) as a more accessible alternative. Third, this model is applied in a challenging open air environment, as opposed to more controlled contexts such as greenhouses.

II. Related Work

IoT devices have seen widespread adoption in agriculture, with digital solutions offering the potential to improve yields even in remote areas. IoT systems require a network communication layer to allow either inter-device or device-cloud communication. There are a number of alternative communication protocols used in IoT architecture, each with distinct trade-offs. The following section reviews literature on common protocols used in agriculture (i.e. Bluetooth and Wi-Fi), contrasting them with LoRa which is the communication protocol used for this project Add some related work on non-LoRa IoT, then contrast that to LoRA

A. Non-LoRa IoT in agriculture

One common communication standard used in IoT systems is Wi-Fi. In [5], the authors developed a smart wireless sensor network using 802.11 Wi-Fi modules (WSN802G) to monitor agricultural parameters such as temperature, air pressure and soil moisture. Readings from sensors were transmitted to a standard router and uploaded to the cloud for storage and retrieval. While the system successfully enabled remote monitoring, Wi-Fi has relatively high power consumption. While the authors aimed to move to a solar-battery set up as part of future work future, this was not attempted during the study.

Similarly, studies have investigated Bluetooth for agricultural IoT. In [6], the authors explored an IoT sensor network based on Bluetooth Low Energy (BLE) for greenhouse monitoring. The primary advantage of this technology is very high power efficiency with the authors estimating a battery life of over 8 years of continuous transmission. Their prototype connected ambient light and temperature sensors to a BLE transmitter. The transmission was then received by a BLE compatible router and data uploaded to a web API via WiFi. While this solution offers superior energy efficiency compared to WiFi, the authors noted that BLE has limited range of 30-100m. This would make make Bluetooth unsuitable for open-field contexts that lack nearby network connectivity.

Research has also examined low-rate wireless personal area networks (LR-WPAN) for agricultural monitoring. In [7], the authors developed a Wireless Sensor Network (WSN) using XBee hardware modules operating within the 802.15.4 communication standard to monitor soil moisture and temperature. Their system utilized a mesh topology and consisted of sensor nodes, routers, and a central coordinator to facilitate communication. While the study demonstrated the low power consumption (62 days) and reliability even when deployed in open air conditions, the technology's shorter transmission range necessitates a dense multi-node network architecture to cover typical farm fields effectively, which adds to cost.

These non-LoRa technologies present certain advantages over LoRa but also come with significant disadvantages in an off-grid context. While Wi-Fi provides high bandwidth, its high power consumption and short range restrict off-grid viability. While BLE and LR-WPAN offer excellent energy efficiency, their limited transmission ranges require dense mesh networks using multiple nodes to to cover typical agricultural areas and must be within close reach of an internet gateway connection. In contrast, LoRa addresses these specific constraints by offering a balance of long-range communication and low power consumption without the need for complex intermediate infrastructure.

B. LoRa IoT in agriculture

LoRa (Long Range) is a radio modulation technique that allows for the transmission of data over very long distances (over 4000 times greater than WiFi [8]) while using very little power. This makes it a preferred technology for the remote, off-grid applications. LoRa receiver can aslo to distinguish signals even when background noise is "louder" than the LoRa signal [9]. The main disadvantage of LoRa is a comparatively low data throughput - around 5 kbps in the configuration used here.

Papers examining the effectiveness of LoRa in agricultural applications include [10] where LoRa was used in an edge computing exercise. In this study the authors used CNN machine learning to create a compressed image that holds thousands of simulated climate readings. This image can then be sent over LoRa to a receiver node which can infer the readings of each node from this single image. While only one sensor node was created for the exercise they also tested the range of this device at a distance of 200m. This system would be useful in particularly large networks of LoRa devices where the low data transfer speed of LoRa would start to be a limiting factor.

The authors in [11] implement a LoRa based weather station prototype in India. The authors create a node that measures temperature, humidity and soil moisture in an experimental setting with no field deployment. Readings are then sent via LoRa to a receiver and can be read manually from the device's screen or viewed on an IBM dashboard.

Conclude by saying that LoRa is particularly relevant in fields, with no wifi...

I've added some commentary on this to the non-lora section

C. Weather forecasting microclimates with machine learning

General weather models operate at magnitudes between 1 and 10 km and microclimate predictions require models that operate at scales of roughly 100m or less. Using existing general forecast models for micro-scale predictions is computationally expensive [12], and these models have lower accuracy rates than predictions using machine learning due to the inherent complexity and non-linear nature of microclimates. Therefore a number of studies have focused on building bespoke models to predict very local forecasts using machine learning processes.

A 2021 study by Kumar et al [13] developed an ML framework called DeepMC as a part of a Microsoft Research initiative. Their model is able to predict a variety of climatic variables such as soil moisture, wind speed and temperature using inputs from weather station forecasts and IoT sensors. They were able to get up to 90% accuracy with a 12-120 hour forecast range.

Zanchi et al [14] used physical modelling of local terrain combined with deep learning (DL) to forecast the microclimate in the foothills of Lombardy. The objective was to predict the local conditions at the meter-scale as opposed to the 10km+ scale of regional and global weather forecasts. The initial model combined data about the morphology of the local terrain and weather forecast data to provide the input data for two feed-forward neural networks. These neural networks were trained to predict the local weather variables using data from 25 sensors deployed in the region being studied. The study demonstrated that local predictions were more accurate when using forecast data from local weather stations as opposed to global climate datasets, but accuracy was high in both cases.

It was notable that in this study only 4 of the 25 sensors used ran without failure.

Blunn et al [12] ran a study focussed on predicting temperatures in urban environments during heatwaves, using data from eight heatwaves in London, UK. They used data from the UKV - a high-resolution weather forecasting model - and from citizen weather stations (CWS). The authors used a similar model training design to that in this study. A number of ML models were trained on UKV variables (i.e. a general forecast) and CWS variables (local sensor data) to bias correct the UKV readings and create a forecast prediction model that could predict the CWS readings accurately (mean average error: 0.12°C) compared with the general weather readings from UKV (mean average error: 0.64°C). The main points of difference to this paper are the use of only temperature versus a wider range of variables in this study, along with the use of custom weather stations here compared to public weather data.

A recent paper from Abdelmadjid et al 2025 [15] used online datasets from Kaggle (a public repository of various datasets) to develop an ML tool to predict changes in temperature and humidity within greenhouses in response to changes to external weather conditions. They used these data to test three ML models and three DL models and selected the LightGBM ML model and the LSTM DL model as the best performing models for prediction. The overall system design consisted of four LSTM models feeding into the LightGBM model. This design resulted in 98.45% accuracy for temperature predictions and 99.61% accuracy for humidity predictions. Due to these results LightGBM was also chosen for this experiment.

Note cold start challenge, and cost of equipment and wifi needs - and conclude that these challenges are not addressed in related work and are addressed in this paper.

There are still challenges that these studies do not address. First, many studies do not provide specific detail on the costs of their sensors and in some cases rely on high-cost proprietary hardware as in [13]. While the authors in [15] give a comprehensive overview of many ML models in forecast prediction their data come from pre-existing public datasets, ignoring the practical challenge of deployment. Studies that do deploy actual hardware such as in [14] often face reliability issues, in that study the authors had a high failure rate in their sensors. This paper addresses these gaps by presenting a complete solution from the hardware to the software level. The use of LoRa allows this solution to work in the open-air environment over greater distances, making the use of ML based forecasting more practical for food producers.

III. Methodology

methodology is Design Science: Analyse, design, deploy, evaluate improve, repeat. I think we had two cycles, the first resulted in battery doubling and box re-design with shading as well. The 3rd is in progress (deploying at Langford). Add this as the overview part, then the sub-sections are on specific design choices/solutions.

The methodology for this project broadly aligns with the concept of an iterative design and engineering cycle proposed by [16]. This involved an analysis of requirements; designing the software and hardware; deployment of hardware; and then an evaluation of the solution. There were two distinct development cycles; in the first cycle issues with battery power and solar heating were identified that were then improved upon in the second cycle.

Presently, a third development cycle is underway. In this cycle, the weather stations are being modified to include wind direction measurements and gas detection (carbon dioxide, ammonia, methane and nitrogen dioxide). On completion, the stations will be deployed to Wyndhurt Farm in South West England to measure climate conditions and gas emissions from cattle. However, as this work is ongoing, it remains outside the scope of this paper.

A. IoT hardware overview

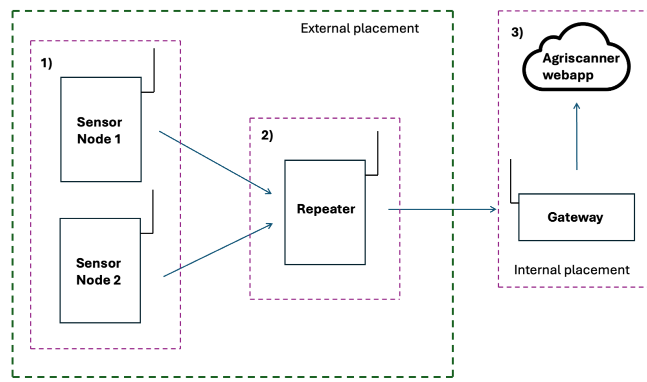

Fig. 1. Network diagram of the system

The design of the hardware consisted of two sensor nodes, a repeater and a gateway. The purpose for each is outlined below:

1) Sensor node: Collects temperature, humidity, wind speed and soil moisture data every 6 seconds. These readings are then averaged and sent as a single packet each minute to the repeater. Figure 2 shows one of these nodes.
2) Repeater: Receives LoRa signals from the sensor nodes and then immediately re-transmits these to boost range.
3) Gateway: A hub that receives LoRa signals and then transmits weather data using WiFi.

All components were commercially available, and assembling the final hardware required only basic tools. Each device used an RP2040 based microcontroller (iLabs Challenger LoRa), which provided the necessary computing power and included built-in LoRa capability for wireless communication. The gateway node was additionally equipped with a Raspberry Pi to enable WiFi connectivity and remote access via the VNC Viewer application.

The microcontrollers were programmed using a distribution of Python for low-power devices called CircuitPython [17].

Fig. 2. Sensor node

B. Solar power and battery life

While developing the nodes, the ability for them to operate without mains power supply was important to ensure they could be deployed in remote applications. The electronics were initially tested in a controlled environment on a fully charged single 2500 mAh 18650 battery. In this configuration, they lasted 19 hours before the battery was entirely drained. However after being tested in the field over a week, this capacity proved insufficient. During periods of low sunlight, nodes frequently stopped transmitting at approximately 06:00. To address this in the second development cycle, a second battery was added in parallel, doubling the capacity to 5000 mAh. Subsequent monitoring confirmed this was sufficient to maintain power through the night and recharge fully during the day during the August data collection period.

C. Weatherproofing

All sensitive electronics were housed in an IP65-rated waterproof junction box. Most external components were water resistant by design with the exception of the soil moisture sensor: In this case the electronics were coated in non-conductive nail varnish and sealed with heatshrink. All cable entry points to the main box were sealed with silicone to prevent water ingress.

D. Solar heating issue

In the first development cycle, the node was constructed with the temperature and humidity sensor housed in a small IP55 junction box painted white. However, data from the initial deployment revealed a significant temperature discrepancy between shaded areas and sun exposed areas. The sensor readings showed a delta of approximately $5°C$ compared to local forecast air temperature, indicating that the white paint was not sufficient to prevent solar radiation affecting the accuracy of readings.

To address this in the second development cycle, a custom solar shield was created using a reshaped aluminium can mounted on a wooden frame. Both the can and frame were also painted white to improve reflection of solar energy. This shield was installed to surround the sensor housing, blocking direct solar radiation while maintaining good airflow over the sensor. After the installation, readings in sunlight were in line with shaded readings.

E. Deployment and data collection period

The nodes were completed and installed in August 2025 in a private garden for a test deployment collecting local readings from two different locations within the garden. This helped facilitate easy repairs and updates to the devices.

The system ran continuously in this location from the 15th of August, 2025, which provided the data used for the machine learning model's training and evaluation later that month.

F. Webapp design

Weather data was sent from the gateway to a webapp called Agriscanner. This webapp allowed for the displaying of current, historic and future (predicted) weather.

The dashboard (Figure 3) displayed live weather data from the nodes with a 1 minute update frequency. Clicking on a particular data point allows the user to see a graph of past and future weather data in a way that is familiar to any user of standard forecasting apps (Figure 4).

Fig. 3. Webapp main dashboard

G. Training and deployment of Machine learning algorithm

To forecast microclimate date up to 48 hours in advance, we trained 10 separate machine learning models using the LightGBM algorithm. One model was created for each of the five sensor variables (temperature, humidity, wind speed, gust

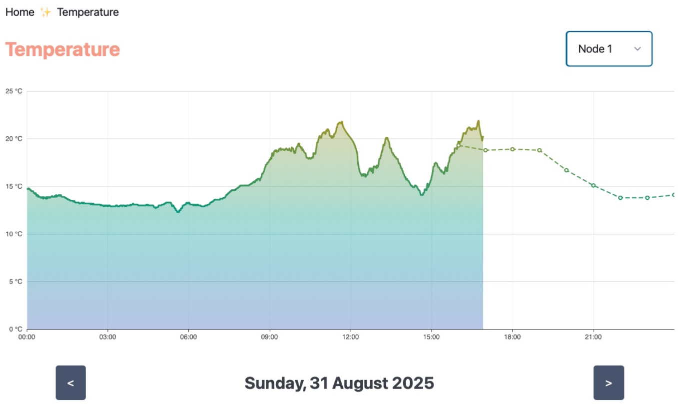

Fig. 4. Webapp temperature page showing current and predicted weather

speed and soil moisture) for each of the two nodes. LightGBM was selected for its high performance on tabular climate data as the authors in [15] show. An additional benefit of this set up is that LightGBM is compatible with the m2cgen library which allows the conversion of the final models to individual JavaScript files.

1) General forecast source: The model was trained using both historical sensor readings collected in the field and general forecast data taken from OpenWeather's *One Call API 3.0* [18].

This weather API was selected as it was free to use and offered a high resolution of forecast readings, with new current and future weather data produced every 10 minutes.

2) Training procedure: The final models would take future general weather forecast data as input and correct for any bias compared to the sensor data. The output was then a machine learning corrected forecast that was predicted to give a more accurate forecast of the conditions in the microclimate. This was predicted to improve forecasting accuracy.

The following steps were followed to train each of the ten models, also shown visually in Figure 5

1) **Dataset prepared:** A single cleaned dataset was created by matching timestamps between the api weather data and the sensor node data. As API readings are taken every 10 minutes and node readings every 1 minute, this meant that 9/10 node readings were discarded. The final dataset was roughly 1,400 rows. The data used for training spanned the period 15 - 27 August. Provide dataset as supplementary material. Made note to add when submitting
2) **Feature set and target data defined:** The feature set from the weather API and targets from the node data were defined, and unnecessary columns discarded. The database timestamp field was transformed into sine and cosine representations of day and year. This is necessary when training on a time-series data set as the algorithm must be able to understand the cyclical nature of time. For example, using raw timestamps would incorrectly suggest to the algorithm that the times of 23:00 on day 1 and 00:00 on day 2 are not closely related.
3) **Dataset split into training data (80%) and validation data (20%):** The data are split by time so the training data consists of the first 80% of the rows and the validation data the last 20%. These data are then supplied to the model.
4) **Iterative training:** For each iteration, the model looks at the inputs (training features) and the correct answers (training target) of the training rows, and determines where it is getting incorrect outputs. It builds a small decision tree that specifically aims to correct those mistakes on the training rows and adds that tree into itself so its predictions change a little. It then applies the updated model to the validation inputs (validation features) and compares those predictions to the validation answers (validation target) —to see how well the model would do on new "unseen" data. The validation data are never used to build the tree; they are only used to check the accuracy of the model. If the validation check shows no improvement after a number of iterations, the training stops and the model keeps the version that performed best on validation. The process will perform a minimum of 50 iterations. I set the maximum number of iterations to 250 to prevent the models getting too large, as each iteration increases the model size substantially (The humidity model is over 40,000 lines long in JavaScript format for example).

Once the ten models had been trained, they were uploaded in JavaScript format to the web server. An automated function in the webapp provides the models with data from general forecast hourly up to 48 hours in advance. This general forecast data was then fed through each of the models to provide a 48 hour prediction of each sensor variable (temperature, humidity etc.). These predictions could then be requested by the front end in JSON format to display a line graph of predicted values for the next 48 hours (see dotted line in Figure 4).

IV. Results

The system was evaluated in two key areas. First, we evaluated the performance of the machine learning forecasting. Then we evaluated the custom hardware by quantitatively comparing it with commercial alternatives.

The results for machine learning are limited by a small training data set, with only two weeks of sensor data captured before training. This limitation was compounded by the fact that the chosen period had virtually no precipitation. This factor limited the accuracy of the ML model during the evaluation period, as there was significant rainfall at that time. Therefore, predictions of humidity and soil moisture were generally less accurate than expected.

A. Machine learning performance

The accuracy of the machine learning prediction was evaluated over a period of 48 hours from 12am on 2025/08/30 to 12am on 2025/09/01. This was the period following the training of the model so was not included in the training data.

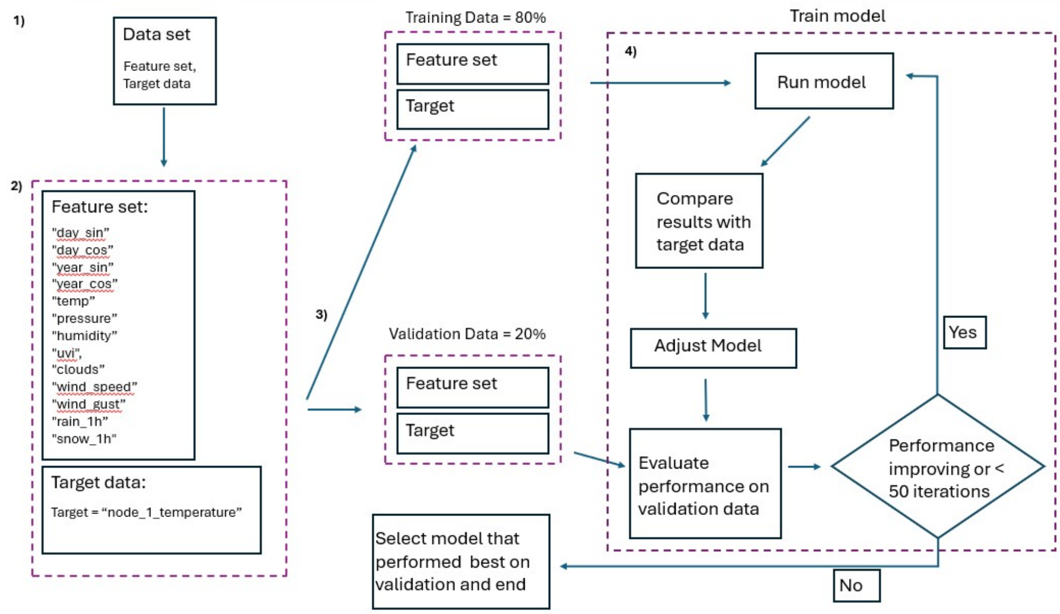

Fig. 5. Infographic showing training steps for training with LightGBM

The performance of the model was compared to two alternatives:

1) **Raw general forecast from OpenWeather:** This was the input to the machine learning model and would give an indication of whether the accuracy could be improved.
2) **Alternative mean average adjustment:** This simple alternative model applied a constant mean average adjustment to the above raw forecast. This would help identify if a simpler model than the LightGBM process could be used instead.

Figure 6 shows the machine learning model performance had mixed results:

1) Wind and gust speed: One relative success of the model was in the prediction of wind speed. The general forecast and alternative model were completely unusable in this regard with MAE of around 200 to 400%. This overestimate is likely due to the macro-scale forecast using wind speed data from much higher positions (often 10m) where wind speeds are much higher. Even the adjusted model does not correct this enough to come close to the true figure with an MAE of 232%. The machine learning model achieved a dramatically more accurate prediction with an MAE of 44%. While still a notable error, this represents a large improvement over the general forecast, which helps to validate this localised forecasting approach.

2) Temperature and humidity: For temperature and humidity, the ML model did not perform as well. The performance differences for temperature were minor overall, with all models achieving a low MAE. The alternative prediction achieved the lowest error (1.3% MAE), followed by the ML prediction (2.9% MAE) and general forecast at (3.4%). This suggests that temperature did not differ significantly in the deployed microclimate compared to the macro-climate, and a simple static bias correction is quite effective for this prediction.

The humidity forecast however showed a distinctly worse result for machine learning (14.9% MAE) compared to both the general forecast (5.5% MAE) and the alternative prediction (11.1%). However in RMSE the three predictions are more tightly grouped; RMSE is more sensitive to large outliers suggesting the general forecast had more occasional but large errors.

The poorer humidity performance is an indicator that the model overfit to the small training sample. With no precipitation in training the model could not accurately predict how rain would affect the local humidity.

3) Soil moisture: A key benefit of the ML model approach is that soil moisture can be predicted, which is not usually reported in general weather forecasts. Therefore the machine learning model is the only model with any prediction for this variable.

The high error for soil moisture (162% MAE) is a predictable consequence of the lack of precipitation in the training data. When it rained before and during the evaluation period the model had not learned the relationship between general forecasted rain and an increase in soil moisture. This meant the model incorrectly predicted dry soil for the entire evaluation period.

B. Hardware evaluation

The other key focus of this work was to create a prototype hardware network that was both low-cost and robust enough for the farm environment.

1) Range: The LoRa modules were tested in an urban park in Bristol. While relatively flat, the park topography still

	Average MAE (%)					Average RMSE (%)				
	temperature	humidity	wind speed	gust speed	soil moisture	temperature	humidity	wind speed	gust speed	soil moisture
General forecast	3.4%	5.5%	333.6%	381.9%	N/A	9.8%	14.7%	348.8%	398.6%	N/A
ML prediction	2.9%	14.9%	43.6%	37.6%	162.0%	13%	18%	58%	47%	162%
Alternative prediction	1.3%	11.1%	232.4%	264.0%	N/A	9.2%	17.5%	253.7%	287.6%	N/A

Fig. 6. Heatmap results showing relative MAE and RMSE averaged over both nodes

resulted in broken line of sight between the LoRa modules at roughly the 1,000m mark so communication was not expected beyond this point.

Fig. 7. Annotated satellite image of range test location with 200m markers

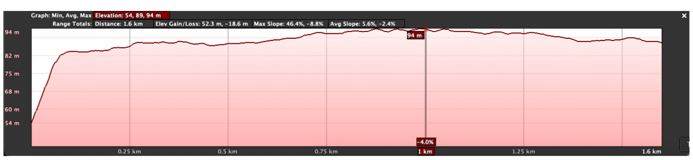

Fig. 8. Elevation of test area showing highest point at 1,000m

Despite the lack of line-of-sight, a final range of 1,200m was achieved. With the inclusion of the repeater, this gives the system a minimum effective range of 2,400m. In a real agricultural deployment, the repeater could be placed on a hill, overcoming line-of-sight issues and enabling a range likely multiples greater.

2) Cost: The final cost of the hardware was £520. This makes the system more cost effective than commercial alternatives; many professional grade weather stations are over 10 times this price which makes them inaccessible for many smaller farms.

Table I presents a range of similar stations on the market. Our system is £200 cheaper than the nearest competitor (SenseCAP S2120) which also uses LoRa for transmission. Due to the use of a repeater the estimated range would also double compared to the other LoRa stations, giving much better range. Additionally, as more sensor nodes are added this cost advantage becomes even greater, meaning a larger deployment with many nodes would see even greater savings. The use of LoRa also results in no ongoing mobile sim card fees as in the case with the HOBO system.

However, it should be mentioned that while cheaper there are some specific areas where the alternative models have an advantage. The most striking is the relative lack of sensors on the Agriscanner network, only four sensors vs around 7-11 in other systems. Additionally, the use of hobbyist grade components means precision is lower in Agriscanner; for example the DHT11 sensor used for temperature measurements in this prototype is rated for accuracy of $\pm 2.0°C$, whereas the Decentlab system is $\pm 0.6°C$. There are also intangible benefits such as long-term support and warranties.

C. Software evaluation

The usability of the webapp was also evaluated using a System Usability Scale [23]. 15 participants were tasked with performing common basic tasks on the webapp, such as reading the temperature on a particular date. Then participants filled out a Likert scale questionnaire of 10 SUS questions.

The application achieved a mean SUS score of 87.7, which exceeds the common benchmark of 68 and indicates a high degree of usability. Users also had a successful task completion rate of 92%, which further supports this conclusion. Furthermore, statistical analysis using a Mann-Whitney U test revealed no significant difference in satisfaction between mobile and desktop users ($p > 0.05$), suggesting the mobile design was just as usable as the desktop interface. However, as the sample of subjects was small these results should only be taken as an indication of the webapp's usability and not conclusive evidence.

V. Discussion

issues to discuss?: e.g, The 'cold start' challange? Deployment of many micro-stations? Relevance of the forecast (cost/benefit)? etc.

A. Current issues and future development cycles

In addition to the mentioned lack of data for machine learning purposes, another problem identified was the restart behaviour of the nodes when power was cut out and then restored. If this happened the device should have entered into CircuitPython's "safe mode" which was programmed to restart data collection after a black out or brown out (volatage dips). However the device did not seem to reliably restart as expected

	Agriscanner Network	SenseCAP S2120 [19]	Decentlab Eleven Parameter [20]	HOBO weather station kit [21]	SparkFun Arduino weather kit [22]
Number of sensors	4	8	11*	6	7
Sensor accuracy	Hobbyist	Hobbyist	Professional	Professional	Hobbyist
Communication type	LoRa	LoRa	LoRa	Mobile network	WiFi
Update frequency	1 minute	1 hour	10 minutes	1 hour	1 minute
Readings per hour	60	1	6	10	60
Power source included	Yes	Yes	Yes	Yes	No
Power source	Solar	Solar	Solar	Solar	–
Batteries recharge?	Yes	No	No	Yes	–
Reported battery life	replace $\sim$ 3 years	154 days	Several months	replace 3–5 years	–
Reported range	–	2–10 km	2–10 km	Anywhere with 4G	–
Estimated range	2.4–20 km	1.2–10 km	1.2–10 km	–	10–50 m
IP rating	$\sim$ IP65	IPX6	IP66	IP66	None
Ongoing payment?	–	–	–	Yes – mobile plan	–
Ongoing costs p.a	£0	£0	£0	£132	£0
Cost per sensor node	£177	£287	£3,272	£4,138	£130
Cost per repeater	£93	£0	£0	£0	£0
Cost per gateway**	£66	£122	£122	£0	£0
Battery cost p.a.	£7	£10	£30	£20	£0
Total cost* **	**£520**	**£706**	**£6,696**	**£8,418**	**£260**

** Sensors missing from Agriscanner: Solar radiation, rainfall, barometric pressure, vapor pressure, dew point, wind direction, tilt sensor, lightning strike count / distance*

*** For SenseCAP and Decentlab models the lowest cost gateway available is sensecap m2 at £122*

**** Includes sensors, repeater, gateway, and estimated first-year battery + ongoing costs.*

TABLE I

COMPARISON OF WEATHER-STATION OPTIONS.

and would often need a manual reset if batteries drained - which became more frequent in the Autumn months. In future development cycles, the software on the devices will be re-written in another Python distribution called MicroPython to see if this issue is resolved.

B. Relevance of agricultural forecasting

The study confirms that general macro-scale forecasts often fail to capture critical local variations. For instance, the general forecast overestimated local wind speeds with a relative error of over 300%, likely due to measuring at higher altitudes. In contrast, the locally trained LightGBM model reduced this error to 43%, providing data that is more relevant for precision tasks such as pesticide spraying.

Improved forecasting accuracy would have clear financial value for farmers. Beyond wind speed, accurate local temperature data can help predict overnight frost or pest incubation periods (see [24]). Because the sensor nodes are lower cost than alternatives (£177), a farmer could afford to install more of them across a field to capture climate variation . This gives a much more useful picture of the farm's weather compared to a single expensive commercial station, ultimately helping to save money by optimising when to spray crops or protect them from frost.

VI. CONCLUSION

This paper presented a methodology for forecasting local microclimates in their farms integrated into an end-to-end IoT weather station system. This was designed to provide farmers with the a super-local forecast that macro-scale "general" forecasts are unable to capture.

Our evaluation confirmed that our hardware is a viable cost-effective solution offering superior range to commercial alternatives. More significantly we trained a LightGBM model to predict the error between general forecasts and local sensor data to create a localised forecast. This was most effective in the prediction of wind speed where the model reduced the relative MAE from 333% to 43%. Performance in humidity and temperature was not clearly better than the general forecast but this is almost certainly down to a lack of training data in the models - particularly the lack of precipitation.

Future work will focus on generating a larger multi-season dataset with the aim of building a more accurate model.

REFERENCES

[1] Department for Environment, Food and Rural Affairs (Defra), Health and Safety Commission (HSC), and National Assembly for Wales Environment, Planning and Countryside Department, *Code of Practice for Using Plant Protection Products*, Department for Environment, Food and Rural Affairs (Defra), London, UK, 2006, [Online; accessed 2-November-2025]. [Online]. Available: https://www.hse.gov.uk/pesticides/assets/docs/Code_of_Practice_for_using_Plant_Protection_Products_-_Complete20Code.pdf

[2] B. Drepper, B. Bamps, A. Gobin *et al.*, "Strategies for managing spring frost risks in orchards: effectiveness and conditionality—a systematic review," *Environmental Evidence*, vol. 11, no. 1, p. 29, 2022. [Online]. Available: https://doi.org/10.1186/s13750-022-00281-z

[3] Q. Hu, L. M. P. Zillig, G. D. Lynne, A. J. Tomkins, W. J. Waltman, M. J. Hayes, K. G. Hubbard, I. Artikov, S. J. Hoffman, and D. A. Wilhite, "Understanding farmers' forecast use from their beliefs, values, social norms, and perceived obstacles," *Journal of applied meteorology and climatology*, vol. 45, no. 9, pp. 1190–1201, 2006.

[4] World Meteorological Organization (WMO), *Guide to Instruments and Methods of Observation: Volume III – Observing Systems*, 2024th ed. Geneva: World Meteorological Organization (WMO), 2025. [Online]. Available: https://library.wmo.int/idurl/4/68661

[5] G. R. Mendez, M. A. Md Yunus, and S. C. Mukhopadhyay, "A wifi based smart wireless sensor network for an agricultural environment," in *2011 Fifth International Conference on Sensing Technology*, 2011, pp. 405–410.

[6] D. Taşkın, C. Taşkin, and S. Yazar, "Developing a bluetooth low energy sensor node for greenhouse in precision agriculture as internet of things application," *Advances in Science and Technology. Research Journal*, vol. 12, no. 4, pp. 88–96, 2018.

[7] N. Ahmad, A. Hussain, I. Ullah, and B. H. Zaidi, "Iot based wireless sensor network for precision agriculture," in *2019 7th International Electrical Engineering Congress (iEECON)*, 2019, pp. 1–4.

[8] A. Spiess, "296 LoRa Propagation, Range, Antennas, and Link Budget (incl. LoRaWAN)," 11 2019, accessed: 2025-08-18. [Online]. Available: https://www.youtube.com/watch?v=BOc3N3Yl38o

[9] L. Vangelista, "Frequency shift chirp modulation: The lora modulation," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1818–1821, 2017.

[10] T. N. Gia, L. Qingqing, J. P. Queralta, Z. Zou, H. Tenhunen, and T. Westerlund, "Edge ai in smart farming iot: Cnns at the edge and fog computing with lora," in *2019 IEEE AFRICON*. IEEE, 2019, pp. 1–6.

[11] R. K. Kodali, S. Yerroju, and S. Sahu, "Smart farm monitoring using lora enabled iot," in *2018 second international conference on green computing and internet of things (ICGCIoT)*. IEEE, 2018, pp. 391–394.

[12] L. P. Blunn, F. Ames, H. L. Croad, A. Gainford, I. Higgs, M. Lipson, and C. H. B. Lo, "Machine learning bias correction and downscaling of urban heatwave temperature predictions from kilometre to hectometre scale," *Meteorological Applications*, vol. 31, no. 3, p. e2200, 2024.

[13] P. Kumar, R. Chandra, C. Bansal, S. Kalyanaraman, T. Ganu, and M. Grant, "Micro-climate prediction - multi scale encoder-decoder based deep learning framework," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, ser. KDD '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 3128–3138. [Online]. Available: https://doi.org/10.1145/3447548.3467173

[14] M. Zanchi, S. Zapperi, and C. A. La Porta, "Harnessing deep learning to forecast local microclimate using global climate data," *Scientific Reports*, vol. 13, no. 1, p. 21062, 2023.

[15] M. K. Abdelmadjid, S. Noureddine, B. Amina, and B. Khelifa, "Enhancing accuracy in greenhouse microclimate forecasting through a hybrid long short-term memory light gradient boosting machine ensemble approach," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 15, no. 2, pp. 2392–2403, 2025.

[16] R. J. Wieringa, "Design science methodology for information systems and software engineering," 2014.

[17] "Circuitpython," https://circuitpython.org, accessed: 2025-12-03.

[18] OpenWeather. One call api 3.0. Accessed 31 August. [Online]. Available: https://openweathermap.org/api/one-call-3

[19] S. (Seeed), "Sensecap s2120 8-in-1 lorawan weather sensor," 2025, accessed: 2025-08-29. [Online]. Available: https://thepihut.com/products/sensecap-s2120-8-in-1-lorawan-weather-sensor?variant=54458635026817

[20] DecentLab, "Decentlab eleven-parameter weather station for lorawan," 2025, accessed: 2025-08-29. [Online]. Available: https://www.alliot.co.uk/product/decentlab-eleven-parameter-weather-station-for-lorawan-special-order-part/

[21] O. . HOBO, "Rx3000 system kit (international) - weather station bundle," 2025, accessed: 2025-08-29. [Online]. Available: https://www.weathershop.co.uk/rx3000-sys-kit-int-bundle?srsltid=AfmBOoqQWkymjf1HSUGLsR57EoHbqsLn4VCQ00kKK3RauGQ86hw9iMnYyKQ

[22] S. Electronics, "Sparkfun arduino iot weather station," 2025, accessed: 2025-08-29. [Online]. Available: https://thepihut.com/products/sparkfun-arduino-iot-weather-station?srsltid=AfmBOoq78B0tnTz9317fpZFt5icrOpb5GalxbQm7fiBYiqPLlTf5hf3H

[23] J. Brooke, "Sus: A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, 11 1995.

[24] N. Haider, C. Kirkeby, B. Kristensen, L. J. Kjær, J. H. Sørensen, and R. Bødker, "Microclimatic temperatures increase the potential for vector-borne disease transmission in the scandinavian climate," *Scientific Reports*, vol. 7, 2017.