WeatherNet: Nowcasting Net Radiation at the Edge

Enrique Nueve
NAISE
Northwestern University
Evanston, Illinois, USA
enrique.iv@northwestern.edu

Robert Jackson
Geospatial Computing, Innovations, and Sensing
Argonne National Laboratory
Lemont, Illinois, USA
rjackson@anl.gov

Rajesh Sankaran
Mathematics and Computer Science
Argonne National Laboratory
Lemont, Illinois, USA
rajesh@anl.gov

Nicola Ferrier
Mathematics and Computer Science
Argonne National Laboratory
Lemont, Illinois, USA
nferrier@anl.gov

Scott Collis
Geospatial Computing, Innovations, and Sensing
Argonne National Laboratory
Lemont, Illinois, USA
scollis@anl.gov

***Abstract*—In addition to natural processes such as photosynthesis and evapotranspiration, net radiation affects industrial applications such as photovoltaic energy management and solar thermal collection. We propose a deep learning approach for nowcasting net radiation within subhourly and intrahour horizons to better understand and control processes influenced by net radiation. Specifically, we developed a deep-learning-based CNN-LSTM, named WeatherNet, that combines multiple local ground-based cameras and weather sensor data to predict net radiation. Unlike previous methodologies, our approach involves images from three different cameras: a sky-facing RGB camera, a horizon-facing RGB camera, and a horizon-facing forward-looking infrared camera. Further, WeatherNet was designed to run "at the edge" using the Waggle edge computing framework to reduce the data bandwidth, improve the latency of predictions, and eliminate centralized data collection. With our proposed dataset and model, WeatherNet, we present a novel methodology using relatively inexpensive equipment for nowcasting net radiation precisely between a 15- and 90-minute horizon.**

***Index Terms*—Edge, nowcasting, net radiation**

I. Premise

The task of weather forecasting in a short horizon (such as a 24-hour window) is called nowcasting [1]. Intrahour nowcasting can provide needed information to optimize industrial applications such as photovoltaic energy management and solar thermal collection and can provide crucial alerts about unexpected dangerous weather changes. Nowcasting is traditionally performed by using radar data and Numerical Weather Prediction (NWP) models [1]. However, NWP models perform poorly at intrahour nowcasting because of the limits of using radar data, such as high computational expense, spatial resolution, and infrequent availability of observations [2]. We posit that these methods can be improved by supplementing them with local sensor data.

In this study, we focus on nowcasting net solar radiation, which is a crucial variable for renewable energy planning and agriculture. Given the challenges in using radar data in NWP models for intrahour nowcasting of net radiation, using ground-based sensors may be a viable alternative for achieving the greater temporal resolution required for renewable energy and agricultural applications [2]. In addition, assumptions made by both NWP models and satellite retrievals about cloud cover can drastically impact net solar radiation forecasts [2]. To develop a subhourly nowcasting model, we colocated and deployed weather sensors and three ground-based cameras and designed and implemented a deep learning model using the data. In addition to examining the feasibility of developing a nowcasting model that uses ground-based sensor data, we explored the feasibility of deploying such a model in the wild. For deployment, we examined the use of an edge computing framework to host our nowcasting model.

Edge computing is an approach to performing computations at the source of data collection [3]. For example, say a remote weather station is deployed at a relatively challenging location to access in person. With the weather station being infeasible to reach in person, transferring the data collected over the internet becomes necessary. Transferring such data may be infeasible, however, because of bandwidth limitations. Edge computing offers the alternative of performing analysis and forecasting using the collected data at the remote station; thus, only the analysis or forecast results—a relatively smaller amount of data—need to be transferred over the internet. We sought to design our nowcasting model to run on an edge computing framework. We believe that deploying a nowcasting model on such a framework can reduce the data bandwidth, improve the latency of predictions, and eliminate centralized data collection. To examine our model's capability to run on an edge computing framework, we tested our methodology for nowcasting with ground-based sensor data on Waggle, an open source edge computing platform [4] developed at Argonne National Laboratory.

II. Related Work

Recent advancements in deep learning computing hardware [5], software frameworks [6], and mathematical techniques [7] have created a strong motivation to explore the use of deep

learning to perform nowcasting. Shi et al. [8] were among the first to use deep learning for nowcasting. The authors proposed using a model called a convolutional long short-term memory (ConvLSTM) network, which combines a convolutional network (a model capable of learning patterns in an image) and a long short-term memory network (a model capable of learning patterns in a sequence) to perform nowcasting. By combining these two models, the ConvLSTM network is able to learn temporal trends over a sequence of images. The ConvLSTM network made by the authors learned to perform nowcasting by using a sequence of time-stamped radar images as input. In more recent work, Google Research presented a deep learning model, called MetNet, for precipitation nowcasting [9].

MetNet, a ConvLSTM variant, also used sequenced radar images but combined them with optical satellite imagery as input. Through the use of ConvLSTM layers and other novel methods, MetNet was able to outperform the High-Resolution Rapid Refresh system, the state-of-the-art operational Numerical Weather Prediction (NWP) method available from the National Oceanic and Atmospheric Administration for precipitation forecasting within a 7–8 hour window at a spatial resolution of 1 km^2 [9]. Other recent work such as [10] has been exploring the applications of deep learning for subhourly nowcasting of precipitation. Through the use of radar data and deep learning, state-of-the-art results in subhourly nowcasting have been achieved, as demonstrated by [9] and [10]. Here, however, we are focused on the feasibility of nowcasting using a remote ground-based sensor system to enhance nowcasting capabilities for a precise location.

In previous work on nowcasting at the edge, Richardson et al. [11] examined the task of nowcasting net radiation using a ground-based sensor system. They used a sky-facing camera combined with a reflective dish to obtain a wide view of the sky, which served as the input data for their model. In contrast, we sought to design a deep learning model to nowcast using data from a ground-based system. Moreover, whereas Richardson et al. used only one camera for data, we used a sensor system consisting of multiple types of cameras. In particular, we used a horizon-facing RGB camera, a sky-facing RGB camera, and a horizon-facing forward-looking infrared (FLIR) camera. By adding the FLIR and horizon-facing camera images as input to our model, we can obtain additional useful information for nowcasting. With the FLIR camera, which records temperature, we can gain information such as the height of the clouds in an image, which is an insightful indicator of current and future weather activity. By having both RGB cameras, information such as cloud and snow coverage is provided to the model. Moreover, since net outgoing longwave radiation is influenced by cloud-related properties such as height and optical thickness, having the three cameras allows our data to capture both of these cloud-related properties [12]. Through influences of previous work and using a novel ground-based sensor system and deep learning model, we developed a new approach to nowcasting net radiation.

III. Approach

In order to perform nowcasting of net radiation, two kinds of data were collected and used: weather sensor data and ground-based camera image data. Both of these data sources were collected through the Waggle edge computing framework [4]. In particular, the data used in our model was collected by using a Waggle node [4] configured with weather sensors and cameras and was located on the Argonne National Laboratory campus in Lemont, Illinois (Lat: 41.701538, Long: -87.994742). The methods used for collecting and preparing the data are described further in the following sections.

A. Weather Sensor Data

The weather sensor data consisted of samples from February to June of 2020. The weather data was sampled at 1 Hz and later averaged to produce 15-minute samples that were fed to our nowcasting model. The weather sensors collected about two dozen parameters listed in Table I. Besides the weather sensor data, images were collected through a ground-based camera system.

TABLE I: Weather Data

Average 60 m temperature	deg C
Average 60 m wind speed	m/s
Vector-averaged 60 m wind speed	m/s
Vector-averaged 60 m wind direction	deg
Standard deviation of 60 m wind direction	deg
Total precipitation for the period	deg
Estimated heat flux	W/m^2
Average 10 m temperature	deg C
Average 10 m wind speed	m/s
Vector-averaged 10 m wind speed	m/s
Vector-averaged 10 m wind direction	deg
Standard deviation of 10 m wind direction	deg
Average global irradiation	W/m^2
Average net radiation	W/m^2
Estimated surface roughness length	cm
Average 10 m vapor pressure	kPa
Average 10 m dew point temperature	deg C
Day of the month	[0-31]
Month	[0–23]
Hour and minute	[00:00–23:45]

B. Ground-Based Camera Data

The ground-based camera system consisted of a sky-facing RGB camera, a horizon-facing RGB camera, and a horizon-facing forward-looking infrared camera. Images were collected every fifteen minutes from the three cameras to correspond to the 15-minute sample rate of the weather sensor data. The images of all three cameras were 640 by 480 pixels. An example of a set of images from the three cameras can be seen in Fig. 1.

C. Data Preparation

The data was preprocessed before being used to nowcast net radiation with our deep learning model. The first stage of preprocessing consisted of making multiple versions of the dataset, where each was created with a different horizon

periods. In all, six versions of the dataset were formed in order to nowcast net radiation for 15-, 30-, 45-, 60-, 75-, and 90-minute horizon periods. In the second stage of preprocessing, the data was scaled between zero and one. The data then was cut into windows for training and testing, as discussed in detail in the following section.

IV. Process

Through experimentation, we sought to gain insight into the nontraditional dataset's potential, consisting of images from a ground-based camera system and weather sensors, for nowcasting net radiation. In order to gauge the data's ability to be used for the task of nowcasting net radiation, a series of deep learning models were developed, trained, and tested using different combinations of the data. In particular, four experiments were conducted to compare the performance of nowcasting net radiation using deep learning with different variations of the collected data.

The first three experiments were as follows: nowcasting net radiation using the weather sensor data with a gated recurrent unit (GRU) [13], nowcasting net radiation using camera images with a ConvLSTM from the ground-based camera system, and nowcasting net radiation using images from the ground-based camera system and weather sensor data through a combination of a GRU and a ConvLSTM. These first three experiments with their corresponding datasets and deep learning models used data collected from February 2020 to the second week of June 2020 and then tested on the last two weeks of data from June 2020. Each of the three experiments' models were trained and tested to predict 15-, 30-, 45-, 60-, 75-, and 90-minute forecast horizons for net radiation. A fourth experiment was used to test the model in an online learning scenario.

A. Experiment I: Nowcasting Net Radiation with Weather Sensor Data

For the first experiment, a GRU was trained and tested to predict net radiation using the weather sensor data mentioned in Section III.A. The input sequence for the GRU consisted of four sequential samples from the previous hour, since the data was sampled every fifteen minutes. The target value was a future period's net radiation value. The model was trained and tested for 15-, 30-, 45-, 60-, 75-, and 90-minute horizon periods.

B. Experiment II: Nowcasting Net Radiation with Ground-Based Camera Data

For the second experiment, the Convolutional LSTM variant named WeatherNet was trained and tested to predict net radiation using the images collected from the ground-based camera system, as mentioned in Section III.B. As in the first experiment, four sequential samples were used from the previous hour to form an input sequence. Since the system had three cameras (sky-facing RGB, horizon-facing RGB, and horizon-facing forward-looking infrared), three input sequences were made, one for each camera.

Since we had three cameras, the Convolutional LSTM variant had three inputs, one for each image sequence. Because this model was intended to run over an edge computing device that has limited computational resources, the model ideally should have used the least amount of parameters possible in order to ensure that it would fit in CPUs/GPU memory. However, Convolutional LSTMs are computationally expensive because of the large number of parameters used within the layer.

In order to counter this large number of parameters, the model was designed to first feed the input sequences through time-distributed separable convolutional layers. Separable convolutions are an efficient alternative to traditional convolutional layers [14] because they use only about a tenth of the number of parameters that a convolutional layer uses. Thus, the model was capable of encoding the input sequences into a smaller dimension efficiently through separable convolutional layers while preserving the temporal relationships by feeding the sequences first through time-distributed layers. In addition to the three mentioned inputs for the three cameras and their respective images, the model has a fourth input for the weather sensor data listed in Table I.

For the second experiment, the fourth input took in a limited number of variables from Table I: average net radiation, day of the month, month, hour, and minute were fed into a GRU. The GRU's output was concatenated with the Convolutional LSTM layer's output, which was then fed into a dense layer with a linear activation function to perform nowcasting of future net radiation. This side information was fed into the model in order to give context to the input image sequences. The details of WeatherNet's architecture can be viewed in Fig. 2.

C. Experiment III: Nowcasting Net Radiation with Weather Sensors and Ground-Based Camera Data

The third experiment also used WeatherNet to perform nowcasting of net radiation. However, it took in the image sequences from the ground-based camera system and all of the variables mentioned in Table I.

D. Experiment IV: Online Learning with WeatherNet

Since we intend our methodology to be viable in an applied setting for nowcasting net radiation, we conducted a fourth experiment to emulate an online learning scenario with our proposed datasets and models as mentioned in Experiments I–III. To do so, we initially trained a model (from Experiments I–III) on a fixed cycle length (5, 10, or 15 days) and then used the model for inference for the next cycle length amount of days. The procedure was repeated on all our collected data from February 2020 to the end of June 2020. Unlike Experiments I–III, where the forecast windows were 15, 30, 45, 60, and 90 minutes, the forecast windows were always 60 minutes for the fourth experiment. By conducting this fourth experiment, we were able to get insight into how many days are initially needed for the model to perform nowcasting accurately and how the models would perform depending on the number of days in between retraining the model on new data.

Fig. 1: Sample of Images

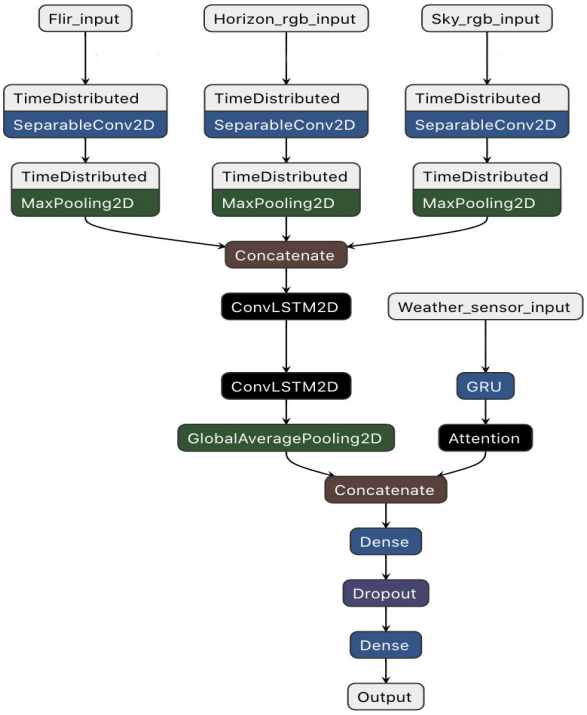

Fig. 2: WeatherNet Diagram

TABLE II: Results of Experiments I–III

E1	**MSE**	**RMSE**	**MAE**	**R2**
15	0.0043	0.0656	0.0470	0.8782
30	0.0059	0.0766	0.0503	0.8337
45	0.0091	0.0951	0.0679	0.7437
60	0.0128	0.1130	0.0762	0.6382
75	0.0097	0.0985	0.0659	0.7252
90	0.0120	0.1097	0.0716	0.6595
E2	**MSE**	**RMSE**	**MAE**	**R2**
15	0.0111	0.1056	0.0654	0.7722
30	0.0122	0.1114	0.0686	0.7411
45	0.0114	0.1066	0.0651	0.7470
60	0.0117	0.1079	0.0652	0.7266
75	0.0118	0.1086	0.0656	0.7075
90	0.0129	0.1134	0.0641	0.6637
E3	**MSE**	**RMSE**	**MAE**	**R2**
15	0.0093	0.0964	0.0627	0.8113
30	0.0131	0.1063	0.0719	0.7599
45	0.0110	0.1048	0.0642	0.7569
60	0.0137	0.1173	0.0718	0.6777
75	0.0121	0.1100	0.0656	0.6999
90	0.0140	0.1184	0.0796	0.6332

V. Outcome

In this section the results from the experiments in Section IV are analyzed.

A. Results of Experiments I, II, and III

For Experiments I–III, the maximum nowcast R^2 value was from Experiment I with a 15-minute horizon, producing an R^2 of 0.8782. The minimum nowcast R^2 was from Experiment III with a 90-minute horizon, producing an R^2 of 0.6332. On average, the model's R^2 values stayed above 0.7 for all of the experiments up to a one-hour horizon. Past the one-hour horizon, however, R^2 began to drop. The overall results from Experiments I–III can be viewed in Table II.

It was observed that E2's model, which had image data, had a higher R^2 value than that of E1's and E3's model for 60- and 90-minute horizons. From these results, one can infer that adding the image data could make the model more robust for greater forecast horizons.

Although the results show the potential of using ground-based sensor data for nowcasting of net radiation, an apparent shortcoming of the models must be addressed. Specifically, all three models suffered at being able to predict sudden jumps. This shortcoming can be seen in Fig. 3, which shows a series of plots of predictions from Experiments I–III over a three-day forecast between the time window of June 20 to June 22. From Fig. 3 one can also see that on the third peak, representing the net radiation on June 22, the models fail to predict the sudden jumps.

We hypothesize that the improper predictions for the jumps in the data were due to the models not being trained on summer data. For this example prediction shown in Fig. 3, our model was trained on data from February to the second week of June 2020 and then performed predictions for the last two weeks of June. To further this research, it would be beneficial

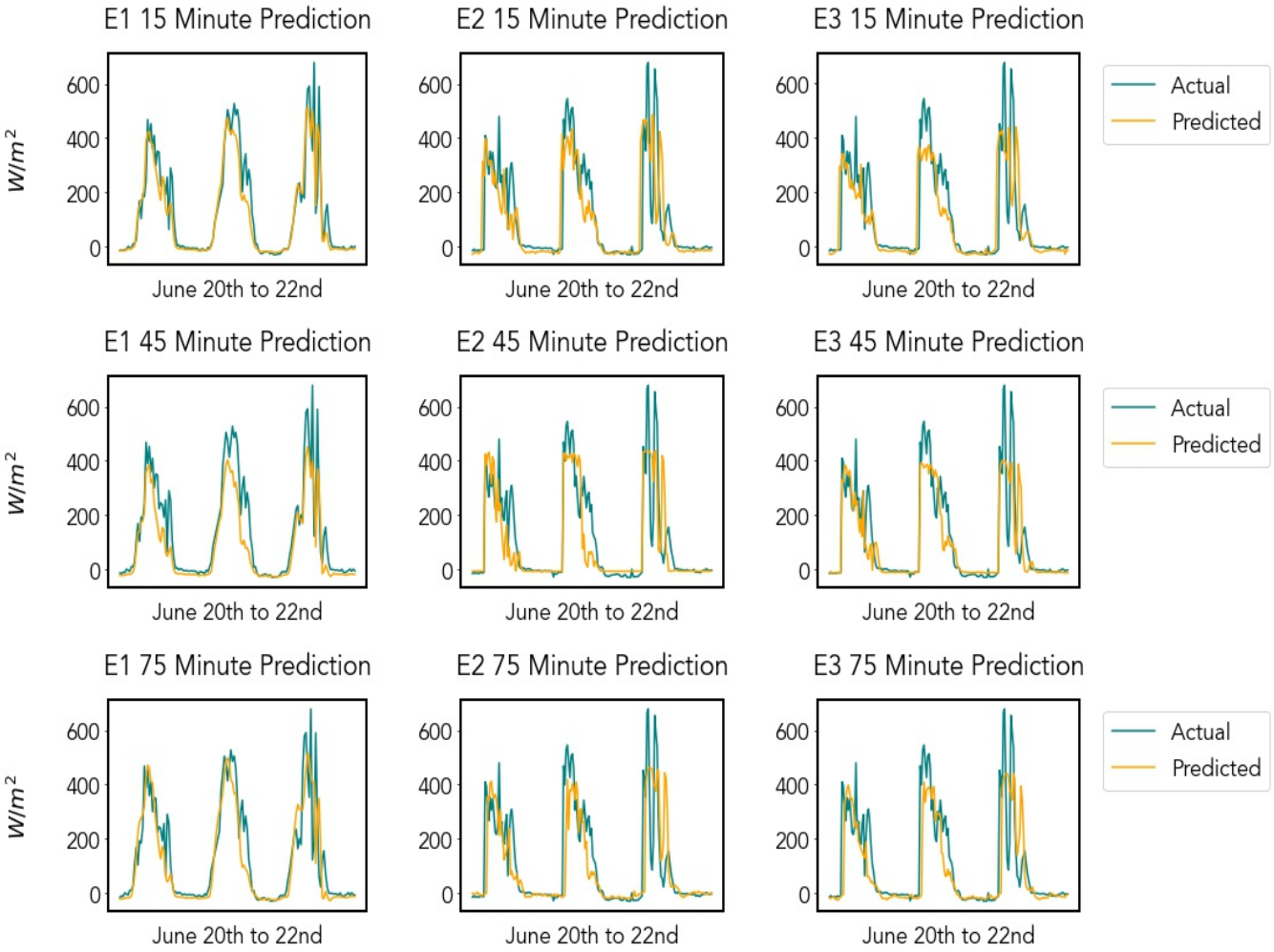

Fig. 3: Example Prediction

to collect at least a year's worth of data and use said data to train the models and then evaluate their predictions. Doing so would allow insight into whether the model's inability to predict jumps is due to a shortcoming of the limited data, the data itself, the model's capabilities, or possibly something different altogether.

B. Results of Experiment IV

Experiment IV was conducted to gain insight into how the different models used for nowcasting would perform in a deployed online learning scenario, where the model were retrained on newly collected data every 5, 10, or 15 days, depending on the test.

As expected, the models' performance, regardless of the cycle length for retraining, increased as the models were exposed and retrained on more data. This experiment revealed issues with having a small cycle retraining length. As observed from Fig. 4, the most performance volatility (measured by R^2) occurred when the cycle retraining length was five days. Otherwise, the rate of improvement of the R^2 values was steadier. However, WeatherNet, which was used for Experiments II and III that used images as data, had much higher volatility in performance than that of Experiment I, which did not use images as data.

During the retraining process in Experiment IV, only data collected from the previous period was used. We hypothesize that doing so caused WeatherNet to overfit to the images and thus have volatile performance in the following period. We remark, then, that if WeatherNet was deployed in an online setting, it would be crucial to balance retraining the new model on newly collected data and old data in order to prevent bias in the model.

As shown by Fig. 4, the models take about eighty days' worth of data to be able to begin nowcasting solar radiation with a modest R^2 and MAE value. This suggests that one should collect at least eighty days' worth of data and use the data to train one of the models before deploying the model in the field. However, as shown by the results of Experiments I–III, we suggest that the model be trained initially on a year's

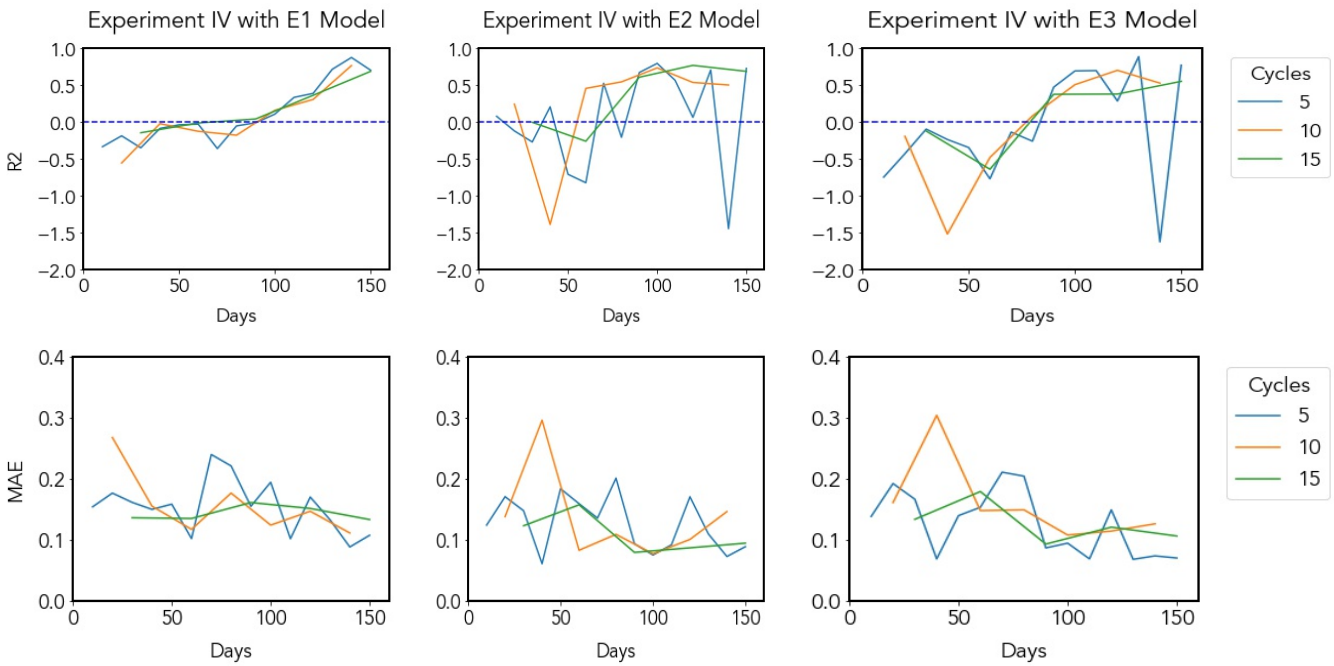

Fig. 4: Experiment IV Results

worth of data. Our hypothesis that more data would improve the models is supported by Fig. 4, showing that MAE drops as the number of days pass and more data is provided to the model. In order to deploy any of the models presented, further research would be necessary to develop an optimal online learning methodology since this can significantly affect the model's performance.

To summarize the results from our experiments, it is plausible that this additional data in combination with a nowcasting system that uses radar data could be used to create a new enhanced model that combines low-altitude and high-altitude data for nowcasting. As stated in the introduction, we do not intend our presented methodology to be a replacement for traditional NWP models or more recent deep learning models that use radar data. However, because of the moderately positive results shown through our experiments, it is plausible that adding ground-based sensor data may help overcome the limitations of radar data for nowcasting such as high-computational expense, spatial resolution, and infrequent availability of observations [2].

VI. Performance on Waggle

The methodologies for nowcasting that we have presented were designed with the intent of being able to perform nowcasting on an edge computing framework. In this section, therefore, we evaluate the feasibility of running the models on a node connected to the Waggle edge computing framework [4]. The node that was used for benchmarking had an Nvidia-AGX-Xavier, a microcomputer designed for machine learning on edge devices. The Nvidia-AGX-Xavier has a peak compute for the GPU of 32 teraops, making this an ideal machine for deep learning over an edge computing framework.

Each of the models was able to perform inference successfully over the remote node. Table III shows performance metrics of the Nvidia-AGX-Xavier for the three models from Experiments I–III. The benchmark values are the peak values from when the models were performing inference. As demonstrated in the table, all the models were able to run in real time on the Waggle node, thus showing that running these models over an edge computing framework such as Waggle [4] is computationally feasible.

VII. Conclusion

Through the use of a ground-based sensor system consisting of cameras and weather sensors, a nontraditional dataset was created to perform nowcasting of net radiation. By experimentation, we were able to gain insight into WeatherNet's ability to nowcast net radiation using the proposed dataset. WeatherNet was shown to have promising nowcasting abilities for net radiation in subhourly and intrahourly horizons (15–90 minutes); however, it was less successful in predicting sudden jumps. Through future research, we hope to increase the prediction horizon of WeatherNet using the ground-based sensor data and also to correct the underlying issue causing WeatherNet to struggle to predict sudden jumps.

In this work, we also showed that WeatherNet has the ability to run on the Waggle edge computing framework efficiently, suggesting that WeatherNet is feasible for deployment in the wild. We look forward to further developing WeatherNet's ability to nowcast net radiation using ground-based sensor data

TABLE III: Values measured at time of highest GPU usage

Benchmark of Experiment I on Node		
RAM	1979/7772MB	
SWAP	3/3886MB	
CPU Usage	Core 1: 81%	Core 2: 10%
	Core 3: 0%	Core 4: 9%
	Core 5: 0%	
GPU Usage	36%	
CPU Temp	38.5 C	
GPU Temp	38 C	
Power Consumption	4847 Milliwatts	
Benchmark of Experiment 2 on Node		
RAM	7161/7772MB	
SWAP	84/3886MB	
CPU Usage	Core 1: 0%	Core 2: 81%
	Core 3: 0%	Core 4: 0%
	Core 5: 0%	
GPU Usage	99%	
CPU Temp	47 C	
GPU Temp	46 C	
Power Consumption	5021 Milliwatts	
Benchmark of Experiment 3 on Node		
RAM	7155/7772MB	
SWAP	456/3886MB	
CPU Usage	Core 1: 0%	Core 2: 0%
	Core 3: 9%	Core 4: 0%
	Core 5: 9%	
GPU Usage	99%	
CPU Temp	39.5 C	
GPU Temp	38.5 C	
Power Consumption	4944 Milliwatts	

and developing methodology for deploying WeatherNet on an edge computing framework such as Waggle [4].

ACKNOWLEDGMENTS

We thank Pete Beckman (MCS), Sean Shahkarami (MCS), Yongho Kim (MCS), and Seongha Park (MCS) at Argonne National Laboratory for fruitful discussions, feedback, insights, and ideas during the course of this work. The Waggle platform design was supported through Argonne National Laboratory's Laboratory-Directed Research and Development program, LDRD: 2014-160-N0. The SAGE project is funded through the U.S. National Science Foundation's Mid-Scale Research Infrastructure program, NSF-OAC-1935984 [15]. Part of the Weather Data Analysis effort was supported through Argonne National Laboratory's Laboratory-Directed Research and Development program, LDRD: 2020-0471. This material was based upon work supported by the U.S. Department of Energy, Office of Science, under contract DE-AC02-06CH11357.

REFERENCES

[1] Y. Wang, E. Coning, A. Harou, W. Jacobs, P. Joe, L. Nikitina, R. Roberts, J. Wang, J. Wilson, A. Atencia, B. Bica, B. Brown, S. Goodmann, A. Kann, P.-w. Li, I. Monterio, F. Schmid, A. Seed, and J. Sun, *Guidelines for Nowcasting Techniques*, 11 2017.

[2] A. Kumler, Y. Xie, and Y. Zhang, "A new approach for short-term solar radiation forecasting using the estimation of cloud fraction and cloud albedo," 10 2018.

[3] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[4] P. Beckman, R. Sankaran, C. Catlett, N. Ferrier, R. Jacob, and M. Papka, "Waggle: An open sensor platform for edge computing," in *2016 IEEE SENSORS*, 2016, pp. 1–3.

[5] Y. LeCun, "1.1 deep learning hardware: Past, present, and future," in *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2019, pp. 12–19.

[6] Z. Wang, K. Liu, J. Li, Y. Zhu, and Y. Zhang, "Various frameworks and libraries of machine learning and deep learning: A survey," *Archives of computational methods in engineering*, pp. 1–24, 2019.

[7] D. Ramachandram and G. W. Taylor, "Deep multimodal learning: A survey on recent advances and trends," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 96–108, 2017.

[8] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. kin Wong, and W. chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," 2015.

[9] C. K. Sønderby, L. Espeholt, J. Heek, M. Dehghani, A. Oliver, T. Salimans, S. Agrawal, J. Hickey, and N. Kalchbrenner, "Metnet: A neural weather model for precipitation forecasting," 2020.

[10] S. Agrawal, L. Barrington, C. Bromberg, J. Burge, C. Gazen, and J. Hickey, "Machine learning for precipitation nowcasting from radar images," *arXiv preprint arXiv:1912.12132*, 2019.

[11] W. Richardson, H. Krishnaswami, R. Vega, and M. Cervantes, "A low cost, edge computing, all-sky imager for cloud tracking and intra-hour irradiance forecasting," *Sustainability*, vol. 9, no. 4, p. 482, 2017.

[12] M. D. Zelinka, C. Zhou, and S. A. Klein, "Insights from a refined decomposition of cloud feedbacks," *Geophysical Research Letters*, vol. 43, no. 17, pp. 9259–9269, 2016.

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

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017.

[15] P. Beckman, C. Catlett, I. Altintas, E. Kelly, and S. Collis, "Mid-scale RI-1: Sage: A software-defined sensor network (NSF OAC 1935984)," 2019, https://sagecontinuum.org/.