Advances in Management & Applied Economics, Vol. 14, No. 1, 2024, 31-48
ISSN: 1792-7544 (print version), 1792-7552(online)
https://doi.org/10.47260/amae/1412
Scientific Press International Limited

Developing Information Gateway for Intelligent Decision-Making and Stability in Solar Power Generation

Cheng-Wen Lee[1]* and Wen-Chuan Wu[2]

Abstract

This study has developed a specialized information gateway for inverters, utilizing relays, data converters, and single-board computers, among other components. Upon receiving data from the gateway, the server processes it to generate an intelligent solar monitoring system. The platform utilizes deep learning RNN and LSTM algorithms to forecast power generation at solar power plants, allowing for real-time monitoring of weather and power generation. By comparing actual and expected power generation data, the system can adjust equipment maintenance and cleaning schedules. It is designed to automatically alert staff members to take appropriate action when anomalous power generation data is continuously transmitted. Additionally, the system sends an alert for on-site inspection and removal of any abnormal situations to increase the stability of solar power generation. This study employs deep learning and IoT data collection to provide the knowledge necessary for intelligent decision-making and increased stability in solar power generation.

JEL classification numbers: C43, F68, H41.
Keywords: Solar energy, IoT monitoring, RNN, LSTN, Abnormal return.

[1] Department of International Business, College of Business, Chung Yuan Christian University, Taoyuan City, Taiwan. *Corresponding author.
[2] Ph.D. Program in Business, College of Business, Chung Yuan Christian University, Taoyuan City, Taiwan.

Article Info: *Received:* October 18, 2023. *Revised*: November 6, 2023.
Published online: November 9, 2023.

1. Introduction

Many solar power plant monitoring systems lack the ability to respond quickly to unforeseen circumstances, such as equipment malfunctions or emergencies. The systems are often limited to calculating power generation and benefits. In larger industrial plants, if a faulty device cannot be promptly removed, the power supply becomes unstable, and downtime increases. This inefficient approach to equipment issues will not adequately address future scenarios.

Typically, conventional solar monitoring systems are limited to monitoring data in the back-end domain, and they require notification from the site engineers in the event of an issue. Although the new solar monitoring system has been querying pertinent data via the remote network in recent years, it is still unable to actively and intelligently report on its own; Furthermore, as solar power facilities are typically constructed in sparsely inhabited areas, poor network signal quality might lead to data loss or package damage. In terms of power-generating calculations and statistics, errors in the transmission process that lead to anomalous power-generation data or even loss will be a major worry.

In order to determine whether variations in temperature and humidity affect power generation, this paper aims to develop a monitoring system that includes data monitoring, benefit conversion, report management, and abnormal warning. Additionally, deep learning will be used in conjunction with current weather conditions to predict future power generation, and it will be possible to compare the differences between actual and predicted power generation. In the event of data loss or damage, the expected power generation can be utilized to backfill the data. Additionally, in the event of an emergency involving equipment failure, it can give real-time abnormal warning messages, thereby minimizing the need for, and expense of, human personnel.

2. Literature Review

2.1 Monitoring Systems

Solar monitoring systems commonly collect information through receivers or modular hardware, which is then analyzed and processed to perform monitoring tasks. To enhance the energy efficiency of solar systems, Bagdadee, Hoque, and Zhang (2020) suggest utilizing communication systems based on wireless sensor networks (WSN-based). They also implement dynamic controllers to monitor voltage rise and power quality. This communication system ensures that power quality meets the required criteria and facilitates the efficient operation of solar systems. Arge and Pizzo (2000) recently incorporate high-performing wireless metering modems with real-time capabilities in specific solar fields. These modems gather data on the generated power, including its volume and type.

Then, using the eXtreme Gradient Boosting (XGBoost) computation method, they employ the time series and field locations of these data as input characteristics to generate predictions (Chen and Guestrin, 2016). Additionally, they change variables like sequence length and learning rate to determine the anticipated electricity generation during various seasons. The goal of this research is to enhance solar monitoring systems' accuracy and performance. More effective data transmission and monitoring can be achieved with WSN-based communication systems, and machine learning techniques can further enhance predictive skills to assist the solar field in producing more precise power generation forecasts. More resources and technical assistance are offered by these approaches for the administration and operation of solar energy installations.

2.2 Data Forecasting

A technique for imputation of missing values utilizing a regression mechanism in a K-nearest neighbor (KNN) is presented by Bagdadee et al., (2020). They contrasted this method's efficacy with those of four additional missing value imputation techniques: mean, mode, C4.5, and CN2. The KNN is a simple but effective method for classification and performs better than the other algorithms when comparing the experimental outcomes (Guo et al., 2003). To predict solar radiation, Park et al. (2016) utilize deep learning recurrent neural networks (RNNs), and Hosseini et al. (2020) use meteorological environment and cloud cover data. RNN is a neural network model whose output is dependent on the input and state of prior memory and is intended to address time-related challenges.

The Long Short-Term Memory Model (LSTM), is developed to address the issue with RNNs processing long-term sequence training (Sherstinsky, 2020). LSTM, which is developed from conventional RNNs, controls the model training process through the use of input, output, memory cells, and forget gates. The input gate determines whether values are entered and computed when data is entered. After then, the memory cells are utilized by later units and stored in accordance with the computed values. The decision of whether to remove undesirable values from memory cells is made via the amnesia gate. Ultimately, the output gate chooses whether to output the memory cell's value or not.

Furthermore, enhanced RNN models with unique gate mechanisms that better manage long-term dependencies and regulate gradient flow, like extended LSTM and gated recurrent units (GRUs), can be applied (Dey and Salem, 2017). It is necessary to combine suitable activation functions, weight initialization, and other strategies in order to solve the vanishing gradient and gradient explosion problems. It is also necessary to take enhanced RNN models into consideration. By using these techniques, time-series data processing recurrent neural networks can have a better training effect.

Three recurrent neural network models (RNN, LSTM, and GRU) and hybrid models (RNN+LSTM, RNN+GRU, and LSTM+GRU) are tested for predictive power (Muhammad et al., 2019). Their study employs the square root error (RMSE) between the actual and projected values as an indication to evaluate the model's prediction ability. The model is created by figuring out the RMSE value for every model while accounting for various layers and neurons.

2.3 Data Preparation

Since many sensors are set up simultaneously to monitor the site environment and a large number of packages are transmitted to the server at the same time, the frequency of using sensors to transmit values to the server is increasing as the complexity of the Internet of Things (IoT) increases. However, when sensor data is applied to complex network architecture, it is frequently interfered with by many factors, resulting in data loss or damage, so if data occurs under the presumption of ensuring the integrity of the data ways to respond to the situation of loss.

When the percentage of data loss in a study is too high, traditional data loss research uses statistical analysis to compare the data with the data that has been completely removed. It is discovered that there is a clear difference between the two comparison results, such as the loss of the original data. In addition to raising the standard error and decreasing statistical power, an abundance of data can also distort or misrepresent the information it represents, diminishing its validity. At the moment, imputation is the primary method used to handle data loss.

There are two types of imputation: multiple imputation, which employs a specific imputation method to impute the original data several times, and single imputation, which uses only one missing data (Shrive et al., 2006). A single imputation will only result in a full set of post-imputation datasets; on the other hand, the multiple imputation method, which employs a variety of imputation techniques, will yield a variety of data sets with varying outcomes. Lastly, the imputed data set can be utilized independently for deep learning or data analysis; typical imputation techniques include mean imputation and regression imputation.

The mean imputation method replaces a missing value in a data set with its value by averaging a segment's value. This method has the advantage of being straightforward and simple to apply, but it destroys the original data set's distribution or variance. If the dataset has a high proportion of missing values, this will have a significant impact on the imputed data set's overall distribution. Similar to the mean imputation, the regression imputation method works by establishing a regression model or a set of mathematical formulas for predicting missing values (Jadhav et al., 2019). Typically, variables containing missing values are chosen as predictors, and there is a certain correlation between variables. The missing data is then replaced with the predicted values obtained from the regression results. Regression bias may therefore arise, causing the value to differ from the true value. For this reason, the research procedure takes the case site's actual circumstances into account while choosing the best data imputation technique, which is applied in this study.

3. Methodology

3.1 System Process

The first stage of developing a comprehensive platform for monitoring systems uses a hardware-fixed architecture. In order to create a special information gateway, the architecture takes into account components like sensors, relays, data converters, and single-board computers. During the design phase, the placement and quantity of electronics, embedded systems, and computer systems are taken into consideration while selecting a hardware-fixed architecture. Its benefit is that it is simple to create and manage, and once implemented, it is fixed and unchangeable, offering great performance and high stability to satisfy the demands of steady control and efficient production.

Data is transmitted to the database through the information gateway, and to access it, a connection must be established using a programming language such as PHP or Python. Once connected, the data undergoes processing and transformation, and the resulting output is displayed on the monitoring system platform. This aids in assessing whether current power generation is impacted by weather fluctuations and if it deviates from projected values when combined with real-time weather data from the meteorological department. The collected data is pre-processed to ensure compatibility with deep learning models such as RNN, LSTM, etc., enabling data prediction.

The primary data gathered in the power plant includes the optical amplitude, the total power generation of the day, the cumulative power generation of the day, and the total MPPT power of the inverter. The daily cumulative power generation and the daily cumulative light amplitude over a continuous period are the two primary experimental data used in this investigation. Data from the power plant's inverter No. 31 between January 1, 2022, and June 30, 2022, are used as experimental data in the study. The typical power generation data chart should be in the form of a mountain. Data loss occurs during data transfer to the database if the single-day power generation data line chart exhibits oscillation. Python will be used to ascertain the missing number of columns in the daily power generation data archive, ensuring the integrity of the data utilized in numerical statistics and deep learning model training.

3.2 Monitoring System Platform

In order to quickly establish a responsive web design and construct it in accordance with various functional needs, this part makes use of the Bootstrap framework. An excellent user experience can be achieved by employing Bootstrap, which enables web pages to adjust to multiple device sizes. PHP will be used by the website to connect to the database and retrieve the required information. The user will be given information relevant to their question, including the converted power-generating value, the inverter's current state, and other pertinent data. Through the web platform, users may access statistics on the generation of power in real-time, comprehend the state of inverters as they operate, and acquire other pertinent information.

3.3 Meteorological Data Collection

The aim of this study is to show real-time temperature and humidity data on the monitoring system platform and to confirm the relationship between power generation and temperature and humidity at the location of the power plant. In order to achieve this, weather status information for the region where the power plant is located is obtained via an API made available by the central meteorological agency's open weather data platform.

To accomplish two distinct goals, the collecting of weather data is split into two phases. First, download weather information from the Central Meteorological Administration's open platform for meteorological data for the period of January 1, 2022 to June 30, 2022. During the data pre-processing stage, the obtained data, which is in JSON (JavaScript Object Notation) format, is changed into the format required for the study. These ongoing meteorological data periods are fed into models like recurrent neural networks. Second, real-time weather data is gathered from the closest sensing station based on the latitude and longitude of the power plant. Users will be able to examine the power plant's temperature, humidity, and weather conditions thanks to these real-time temperature, humidity, and weather data that are synced with the monitoring system platform.

3.4 Filling of Missing Data

Under typical conditions, the study's data ought to contain all the details shown in Table 1. The whole generating data from the previous day is backed up from the database to the server every day at 00:00 using a scheduled way, and the inverter data is communicated back to the database in accordance with the numbered sequence. Full-text searches of the backup complete power generation data can be performed using Python and other extensions to look for missing data.

When comparing the two, it is important to ensure that the inverter number and the index value of the file's columns align properly. If a missing data column is found, the entire column value that corresponds to the index value is used to fill in the missing column number, which pushes back the index value of the inverter number. The pushed-back index value then replaces the number of missing columns in each column to maintain data integrity, since the cumulative power generation of a single inverter should increase linearly. To prevent any disruptions to this linear curve, the cumulative generation of the previous two inverters is calculated and used to fill in any missing data, if necessary.

In conclusion, note the number of missing inverters. This will help establish whether an inverter needs to be repaired or needs to be harmed if it disappears too frequently. This stage has the flexibility to identify additional problems that call for exception exclusion in addition to ordinary maintenance.

Table 1: Complete Data Sorting Table

Inverter number	Daily cumulative power generation	Time
31	1	2023-01-01 04:30
32	1	2023-01-01 04:30
33	1	2023-01-01 04:30
31	3	2023-01-01 04:35
32	3	2023-01-01 04:35
33	3	2023-01-01 04:35
… ..	… .	… ..
31	500	2023-01-01 19:30
32	500	2023-01-01 19:30
33	500	2023-01-01 19:30

3.5 Deep Learning Model Training

In this study, KNN machine learning and RNN and LSTM deep learning model training were conducted using continuous period power generation data. Prior to training the model, normalize all of the data. To reduce the values to a range near zero, this entails removing the mean from the data and dividing the result by the standard deviation. This stage's objectives are to improve the learning effect of the next neural network and guarantee that the value is within a given range. The processed data is then divided into three sections, each of which is used to train a different model of a KNN, RNN, or LSTM.

The first part is to use the regression mechanism of KNN to determine whether the power generation value of each new record deviates from the regression curve. According to the trend of power generation data in this study, under normal circumstances, it should be a mountain shape, and the power generation value is correlated with temperature and humidity. Therefore, the regression curve is calculated using the regression algorithm of the KNN model, and whether the newly recorded data deviates from the regression curve based on linear regression is evaluated. The power generation value is affected by the daily weather conditions and the degree of sunshine, so the model obtained by selecting different K values needs to use the R^2 score and mean squared error MSE (Mean Squared Error) as evaluation indicators.

The Equation (1) for calculating the R^2 score is 1 - (SSR/SST), where SSR (Sum of Squared Residuals) is the sum of squares representing the difference between the actual and predicted values for each observation, and SST (Total Sum of Squares) is the sum of squares representing the difference between the actual and mean values for each observation. A closer value to 1 denotes a higher degree of data interpretation ability for the model. The R^2 score is a number between 0 and 1. Equation (2) demonstrates how to compute MSE, another assessment indicator that quantifies the square of the average difference between the values that were predicted and those that were observed. The model's prediction and the true value are more closely aligned when the MSE value is less.

$$\Sigma(yi - \bar{y})^2 \quad (1)$$

yi: The actual value of the observation
$\bar{y}$: The mean value of the observation
Σ: the sum of squared residuals of all observations

$$(1/n) * \Sigma(yi - \hat{y}i)^2 \quad (2)$$

yi: Actual value of the observation
J: Predicted value of the observation
Σ: Sum of squared prediction errors of the observation
n: Total number of observations

Using their capacity to model temporal data, RNNs were employed in this study for data prediction. The TimeseriesGenerator function provided by Keras can be used to create a high-quality RNN deep learning model (Kera, 2020), as well as to process time series data and produce samples and labels. It has the ability to create sample and label data through the use of a batch process, Sequence object setup, and sample continuous time series data. To meet varied needs, you can change a number of characteristics.

To separate the training, validation, and test data for later model training, first utilize the start_index and end_index arguments to establish the dataset's sampling starting point. Next, choose the first sample's data range by using the sampling window length (Length). Every sample's data is added to the array of sample data, and the sampling rate determines how long to wait between samples. The sample data of this range and its accompanying label data will be the next data point if there are no data available for sampling in the sampling range. Using the set stride, multiple data points must be split between the two sampling windows. After that, adjust the window range and carry on sampling the subsequent data point until the entire data range has been sampled.

The generated data is utilized as the RNN deep learning model's training set after sampling. After adding the necessary number of layers to the model using the add method, compile the model using the compile method. To customize the model's training process, you can select several optimizers during model compilation. Once the model has been trained, its capabilities can be evaluated using various assessment measures. The average absolute error between the expected and actual values is measured by the evaluation statistic known as MAE (Mean Absolute Error). The more closely the model's predictions match the real observations, the more accurate the model is, as indicated by a smaller MAE number (see Equation (3)).

$$(1/n) * \Sigma|Yi - \hat{Y}i| \quad (3)$$

Yi: Actual observation of the *i*th sample
T÷i: Predicted value of the *i*th sample
Σ: Sum of all samples *i*
n: Sample size

4. Result and Discussion

4.1 System Environment

Various computer and hardware combinations were used for various purposes during the course of this investigation. First, an Ubuntu operating system is chosen, a machine is designated as the server, and Nginx is set up as the web server. Nginx is a good choice for managing websites with a lot of traffic because it is lighter and requires fewer resources than Apache. The database to retrieve the information returned by the inverter and other relevant data is also chosen to be MariaDB. An additional machine is devoted to deep learning operations like LSTM and RNN. Table 2 describes the detailed system environment.

Table 2: System Environment

Project	Specification	
CPU	Intel(R) Core(TM) i5-9400F	AMD EPYC-Rome Processor
Memory	16G	8GB
GPU	NVIDIA GeForce RTX 2060	
Hard drive	256GB SSD 1TB HDD	256GB SSD
Use	Deep learning operations	InebServer、DataBase
Other	TensorFlow 2.6.0 Python 3.9.13	

4.2 Datasets

At the plant, 56 inverters in all were installed. The aggregated information from the case site's 31 inverter numbers between January 1, 2022, and June 30, 2022, used as the dataset for this study. In the study, temperature and humidity data from the publicly available meteorological data source were also utilized. Every five minutes, the values are sent back to the database, as Table 3 illustrates. Package loss is a common occurrence because of the massive volume of data transit. To guarantee the integrity of the data, CSV files that contain the data are backed up to the server every day at 0:00. After the data-filling process was complete, a total of 52128 data points were gathered and prepared for the models. In Figure 1, the dataset data line chart is depicted.

Table 3: Example of a Dataset

c_dpy	Stime	Hum	Temp
0	2022/2/1 07:00	94	14.3
0	2022/2/1 07:05	94	14.3
1	2022/2/1 07:10	94	14.3
2	2022/2/1 07:15	94	14.3
3	2022/2/1 07:20	94	14.3
4	2022/2/1 07:25	94	14.3
5	2022/2/1 07:30	94	14.3
6	2022/2/1 07:35	94	14.3
8	2022/2/1 07:40	94	14.3
9	2022/2/1 07:45	94	14.3
10	2022/2/1 07:50	94	14.3
12	2022/2/1 07:55	94	14.3
15	2022/2/1 08:00	95	14.9

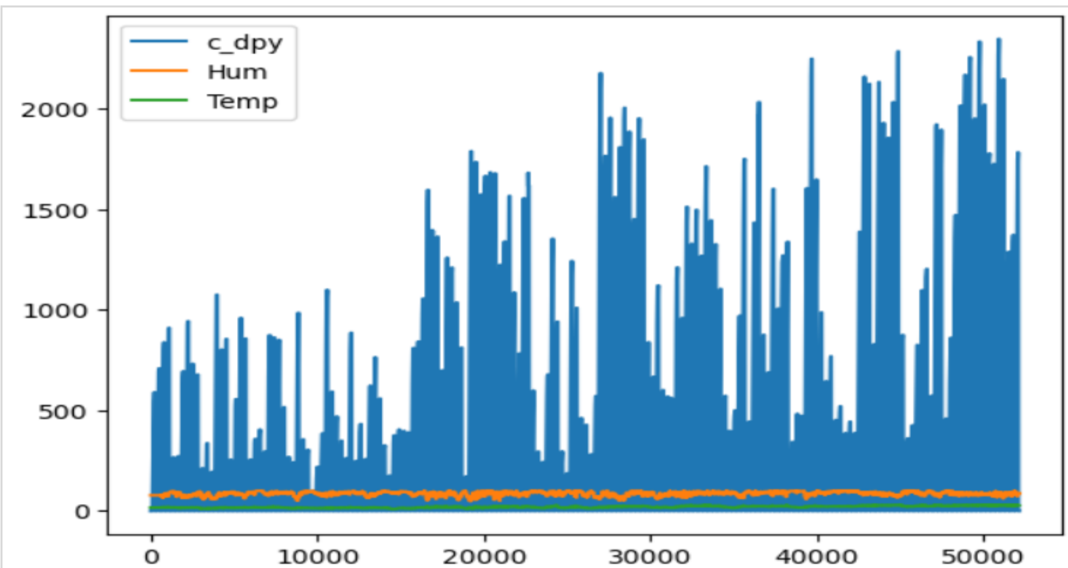

Figure 1: Dataset Data Line Chart

4.3 Data Filling Processing

Device anomalies or data packages could be lost in the course of transferring the data back to the database. Some inverters may have inaccurate cumulative generation values if the data are returned correctly the next time they are returned. The cumulative power generation numerical chart, as shown in Figure 2, will experience high and low oscillations as a result of this circumstance.

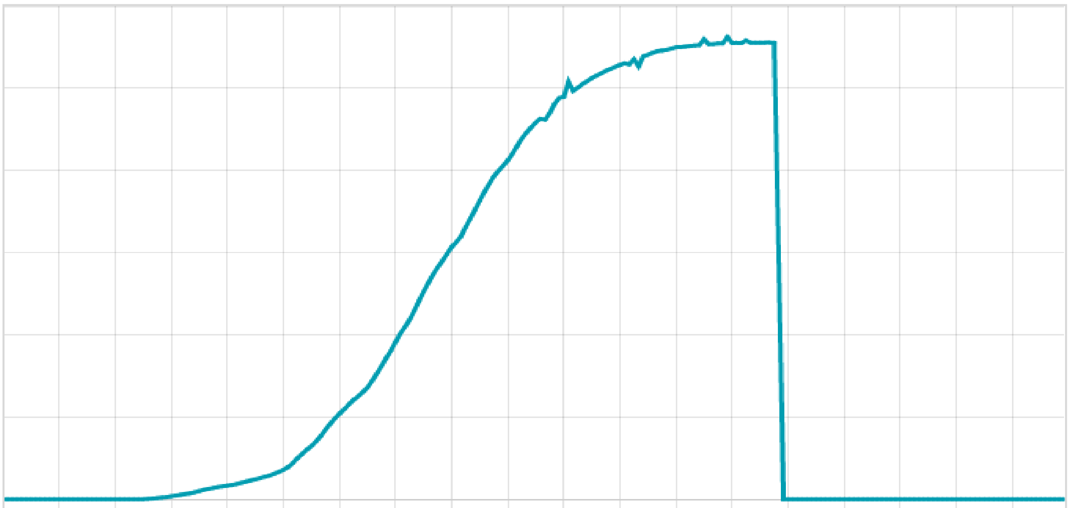

Figure 2: Cumulative Power Generation of Power Plants Prior to Data Filling

The data (CSV file) that was backed up to the database is imported into Python on a daily basis at 00:05. To use related functions for overall data processing, first transform the data from the CSV file into a DataFrame. Next, set the initial inverter number. Then, use the panda's library's loc function inside a loop to check if the inverter number indexed in the DataFrame matches the one that is currently set. In the event that the two are not equal, there is a missing value for the current inverter number. This study employs the pushback approach to determine the value of the same inverter number in the previous item because the order of inverter numbers is unchanging. The data of the same inverter number from the previous item is filled in the missing column to preserve the data's undulating structure. Next, reorganize the index values using the sort index function to prevent mistakes in later comparisons brought on by recently added columns. A graph of cumulative power generation following data filling is displayed in Figure 3. The curve following data filling is smoother and devoid of ups and downs than the curve prior to data filling.

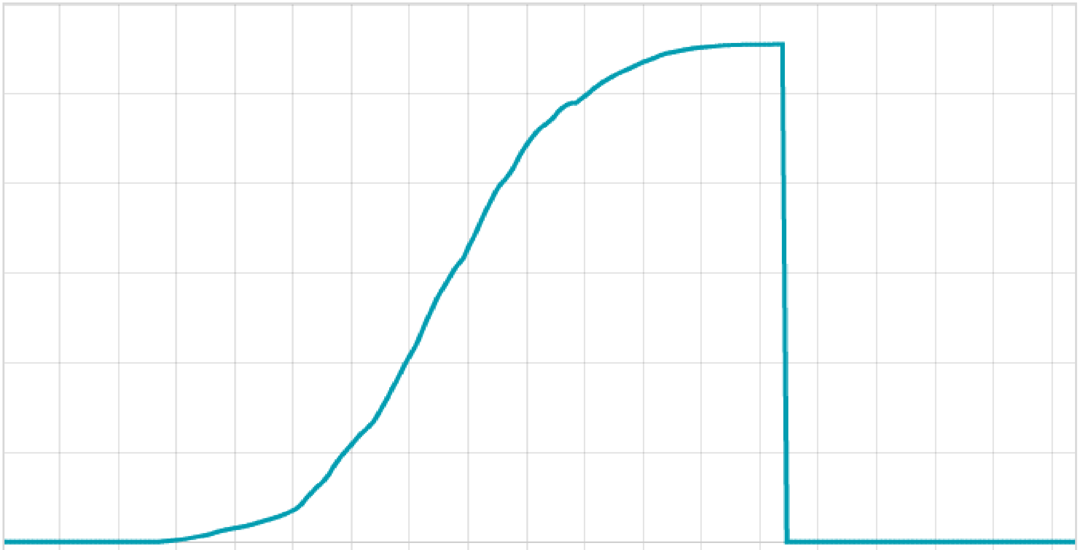

Figure 3: Cumulative Power Generation of Power Plants after Data Filling

When missing conditions are identified, a dictionary is formed and the missing inverter number and cumulative number of missing times are recorded in order to comprehend the inverter with recent missing data issues and its frequency. Managers will be able to see from this record whether the frequency of inverter failure in a given day is excessive, whether an overhaul is necessary, etc.

4.4 Deep Learning Model Training Results and Analysis

Temperature has a strong correlation with the numerical variations of cumulative power generation, which typically does not fluctuate much over short time intervals. As a result, the R^2 score and MSE performance are monitored while various K values are chosen for the KNN model. The one with the closest R^2 score to 1 is 0.821. The model assessment scores for various K values are displayed in Table 4.

Table 4: KNN Model Performance

K value	R^2 score (range from 0 ~ 1)	MSE
1	0.815	41370.031
3	0.821	41368.378
5	0.813	41369.784
7	0.808	41371.508

Table 4 demonstrates that the model's performance is minimally impacted by the selection of various K values. The R^2 value indicates that the model possesses a commendable level of predictive capacity. However, as depicted in Figure 4, the Y-axis represents temperature, while the X-axis represents the power generation value. Despite differences in power generation levels, the temperature remains relatively consistent, resulting in multiple points with diverse values that align with the same temperature coordinates.

The KNN regression model does a good job of predicting outcomes, but it ignores the properties of time series and does not clearly show a correlation between temperature and power generation data. Consequently, using recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), which are experts at processing time-series data, will be the next step.

The RNN and LSTM models in this study were trained using a total of 52128 power-generating data. The parameters, length 288, delay 288, sampling rate 2, stride 288, and epoch number 100, are chosen in accordance with the 7:3 ratio of training to validation data. The model's convergence curve and numerical prediction plot are displayed in Figures 5 and Figure 6 following training. The MAE model score assigns a performance rating of 0.392 to the model.

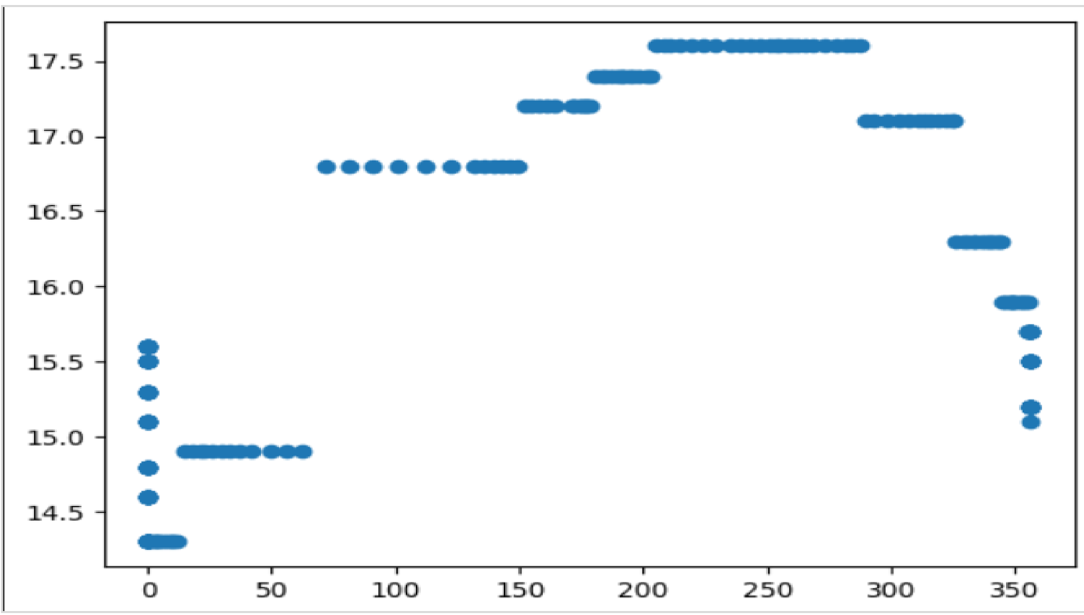

Figure 4: Power Generation Data and Temperature Distribution Diagram

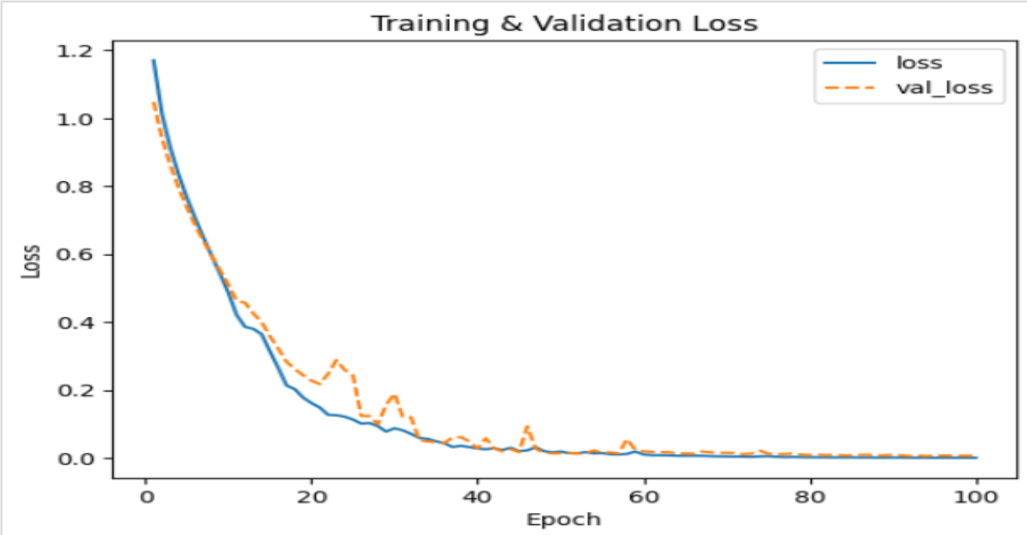

Figure 5: Model Convergence Curve

Because the loss has achieved a stage of convergence, the number of training (Epoch) in this study is fixed at 100. Overfitting may arise as a result of increasing the number of training sessions (Epoch). As a result, adding more pieces of training is not essential. However, the model's performance score still has some space for improvement. In order to boost the model's predictive capabilities, one may consider adjusting the data structure and scale. This may entail modifying the data's structure or feature selection, as well as the size of the training and validation data. By making these enhancements, you may be able to improve the model's precision and ability to generalize. Give it a try and see if you can generate more accurate predictions.

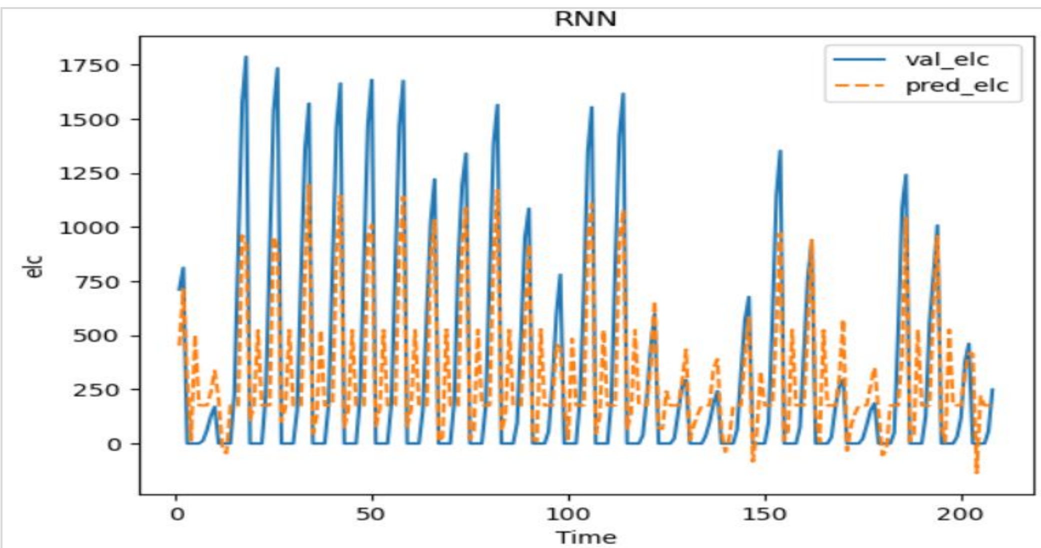

Figure 6: Model Prediction Numerical Trend

The deep learning model is trained using this modified training and validation ratio of 8:2, but the other RNNs' setting parameters stay the same, in order to confirm the effect of the dataset's proportionate distribution on the training outcomes. As can be observed from the data in Figures 7 and Figure 8, the new model's prediction values are closer to the low point value than those of the prior model, although the convergence curve is still very similar. This training's MAE model performance score is 0.253, which is higher than that of the prior training.

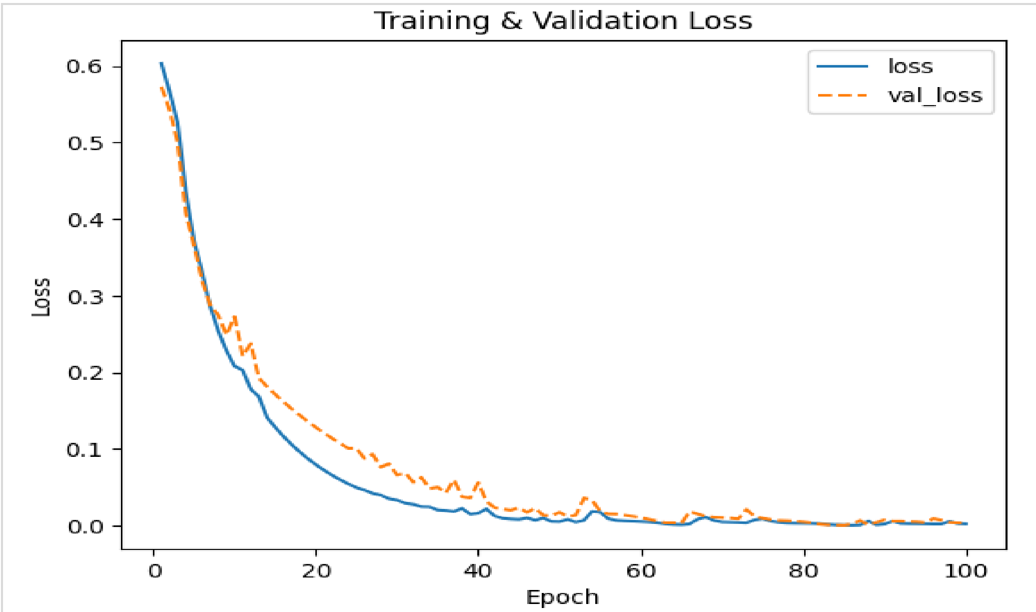

Figure 7: Model Convergence Curve after Modifying Scale

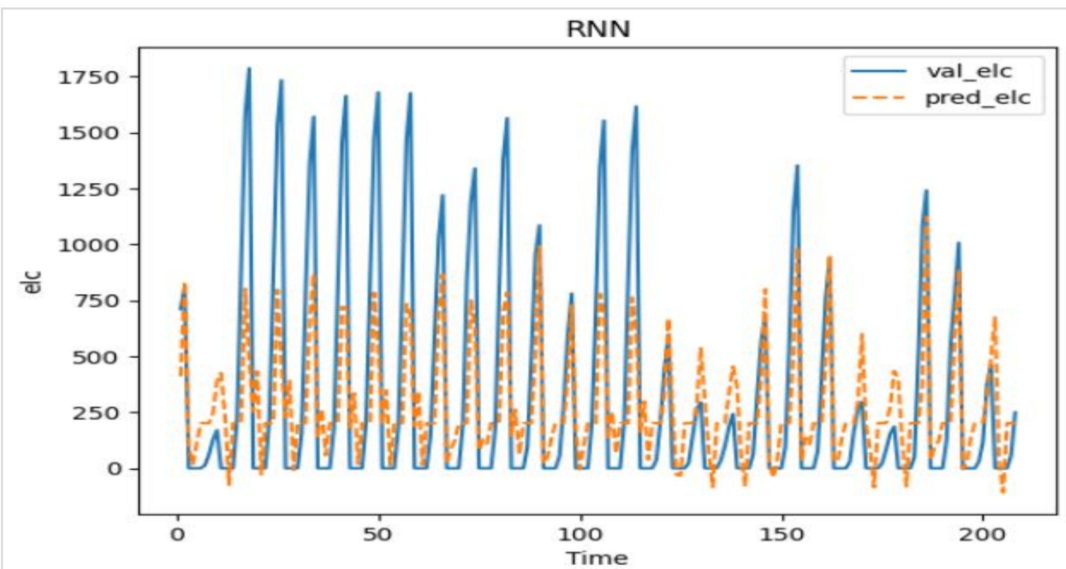

Figure 8: Model Prediction Value after Modifying Scale

The protocol for this study is to back up all of the inverters' data for that day to the server at a certain time every day and to enter all of the inverters' values back into the database with zero before sunrise. Keeping an 8:2 data split ratio for model training, all records with a value of 0 are eliminated because they make up a sizable fraction of the total dataset. Following numerical filtering, the model convergence curve and numerical prediction plot are displayed in Figures 9 and Figure 10.
The slow convergence speed and challenging smooth convergence are evident from looking at the convergence curve during training. The low point value cannot be reliably predicted by the model due to modifications in the data set. It is evident that the MAE model performs the worst, with a performance score of 0.714.

Figure 9: Numerical Filtering Model Convergence Curve

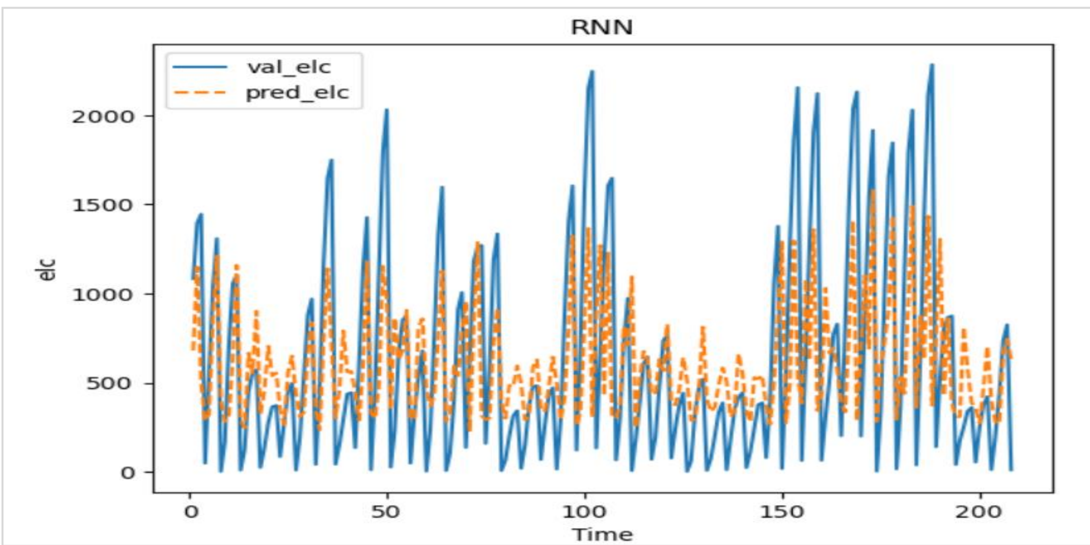

Figure 10: Numerical Filtering Model Convergence Curve

The scoring outcomes of the model trained with three distinct data-splitting techniques are displayed in Table 5. Based on observations, an 8:2 data split ratio outperforms a 7:3 ratio. The data with a high retention score is then chosen for model training after the entire data has been cleansed to eliminate excessive duplicate data. Nevertheless, the outcomes fell short of expectations, and MAE's performance declined rather than remained at the same level. This could suggest that rather than successfully eliminating extraneous data, the process of data purging actually eliminates important information useful for model training, which lowers model performance.

Table 5: Experimental Results

Models and tuning methods	MAE
RNN – Dataset split 7:3	0.392
RNN – Dataset split 8:2	0.253
RNN – Data split 8:2, clear value 0	0.714

5. Conclusion

The system created in this study makes flexible use of an information gateway and a data monitoring platform to comprehend the plant's history records and current power generation status. Simultaneously, the reporting system makes it possible to quickly determine the quantity of equipment and the cause of any abnormalities in the factory area, thereby increasing equipment management efficiency and decreasing the need for needless people use.

In order to preserve the overall data structure, data filling involves finding data with the same inverter number by pushback and filling the missing column with that data. Nevertheless, applying the data-filling approach described in this study will result in a large rise in the value if the same inverter is absent too frequently in a row and there is a normal data backhaul subsequently. Our goal is to present an improved computation technique in the future, enabling more seamless filling in of the missing values and bringing the data set closer to reality.

The duration of the data set employed in this research is barely half a year, despite the fact that the values predicted by the deep learning model do not deviate much from the real scenario. The dataset can then be kept longer, and further deep learning models (like GRU) or statistical analysis can be used to forecast future values. This could increase the model's precision and capacity for prediction, strengthening the system overall.

References

[1] Arge, C.N and Pizzo, V.J. (2000). Improvement in the prediction of solar wind conditions using near-real time solar magnetic field updates. Journal of Geophysical Research: Space Physics, 105(A5), 10465-10479.

[2] Bagdadee, A.H., Hoque, M.Z., and Zhang, L. (2020). IoT based wireless sensor network for power quality control in smart grid. Procedia computer science, 167, 1148-1160.

[3] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).

[4] Dey, R. and Salem, F.M. (2017). Gate-variants of gated recurrent unit (GRU) neural networks. In 2017 IEEE 60th international Midwest Symposium on Circuits and Systems (MWSCAS) (pp. 1597-1600). IEEE.

[5] Guo, G., Wang, H., Bell, D., Bi, Y., and Greer, K. (2003). KNN model-based approach in classification. In On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings (pp. 986-996). Springer Berlin Heidelberg.

[6] Hosseini, M., Katragadda, S., Wojtkiewicz, J., Gottumukkala, R., Maida, A., and Chambers, T.L. (2020). Direct normal irradiance forecasting using multivariate gated recurrent units. Energies, 13(15), 3914.

[7] Jadhav, A., Pramod, D., and Ramanathan, K. (2019). Comparison of performance of data imputation methods for numeric dataset. Applied Artificial Intelligence, 33(10), 913-933.

[8] Keras Development Team. Keras: Deep Learning for Humans.
Accessed: Jun. 24, 2020. Available:
https://keras.io/search.html?query=Development%20team

[9] Muhammad, A.U., Li, X., and Feng, J. (2019). Using LSTM GRU and hybrid models for streamflow forecasting. In Machine Learning and Intelligent Communications: 4th International Conference, MLICOM 2019, Nanjing, China, August 24-25, 2019, Proceedings 4 (pp.510-524).
Springer International Publishing.

[10] Park, S.U., Park, J.H., Al-Masni, M.A., Al-Antari, M.A., Uddin, M.Z., and Kim, T.S. (2016). A depth camera-based human activity recognition via deep learning recurrent neural network for health and social care services. Procedia Computer Science, 100, 78-84.

[11] Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network.
Physica D:Nonlinear Phenomena, 404, 132306.

[12] Shrive, F.M., Stuart, H., Quan, H., and Ghali, W.A. (2006). Dealing with missing data in a multi-question depression scale: a comparison of imputation methods. BMC Medical Research Methodology, 6, 1-10.