



UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,
ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



**“MODELOS DE APRENDIZAJE AUTOMÁTICO DE ENSAMBLE
PARA PREDECIR LA GENERACIÓN DE ENERGÍA ELÉCTRICA
FOTOVOLTAICA BASADOS EN VARIABLES
METEOROLÓGICAS EN LA CIUDAD DE PUNO Y JULIACA”**

TESIS

PRESENTADA POR:

WILSON ANTONY MAMANI MACHACA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PUNO – PERÚ

2022



DEDICATORIA

Dedico este trabajo de investigación a toda mi familia y personas que me apoyaron en los momentos malos. Gracias por enseñarme a afrontar los problemas de manera sabia y correcta. También quiero dedicar este trabajo a Dios dándome valentía y un empuje espiritual para poder seguir adelante con mis metas y objetivos.



AGRADECIMIENTOS

Agradezco a mis docentes por toda la formación brindada en mi etapa universitaria, por transmitir toda su experiencia y conocimiento a lo largo de los años en cada curso asistido y proyectos realizados dentro de las aulas de la universidad. También agradezco a mi asesor por su apoyo en todo momento, por los aportes tan importantes que realizo para guiarme y poder realizar este trabajo de investigación.



ÍNDICE GENERAL

DEDICATORIA

AGRADECIMIENTOS

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN 12

ABSTRACT..... 13

CAPÍTULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA..... 14

1.2. FORMULACIÓN DEL PROBLEMA 15

1.3. HIPÓTESIS DE LA INVESTIGACIÓN 16

1.3.1. Hipótesis General 16

1.3.2. Hipótesis Específicas 16

1.4. OBJETIVOS DE LA INVESTIGACIÓN..... 16

1.4.1. Objetivo General 16

1.4.2. Objetivos Específicos 17

1.5. METODOLOGÍA 17

CAPÍTULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN..... 18

2.2. SISTEMA DE ADQUISICIÓN DE DATOS 27

2.2.1. Adquisición y control de datos en tiempo real 28

2.3. SENSORES..... 29



| | |
|---|-----------|
| 2.3.1. Sensor de temperatura | 30 |
| 2.3.2. Sensor de luz | 31 |
| 2.3.3. Sensor de luz ultravioleta | 32 |
| 2.3.4. Sensor de corriente | 34 |
| 2.4. CONVERTIDOR ANALÓGICO DIGITAL Y DIGITAL ANALÓGICO . | 35 |
| 2.4.1. Convertidores de analógico a digital..... | 36 |
| 2.5. CONECTIVIDAD..... | 38 |
| 2.5.1. Puertos Serie (RS-232, RS-422, RS-485) | 38 |
| 2.6. ENERGÍA FOTOVOLTAICA | 40 |
| 2.6.1. Efecto fotovoltaico | 41 |
| 2.6.2. Generador fotovoltaico..... | 43 |
| 2.6.3. Fabricación de celdas solares | 44 |
| 2.6.4. Componentes de un sistema fotovoltaico..... | 45 |
| 2.7. LA RADIACIÓN SOLAR..... | 48 |
| 2.8. LENGUAJE DE PROGRAMACIÓN PYTHON..... | 50 |
| 2.8.1. Versiones de Python..... | 51 |
| 2.8.2. Recursos útiles de Python | 52 |
| 2.9. MACHINE LEARNING | 53 |
| 2.9.1. Aplicaciones | 54 |
| 2.9.2. Data Preprocessing | 55 |
| 2.9.3. Aprendizaje supervisado | 55 |
| 2.9.4. Aprendizaje no supervisado | 56 |
| 2.9.5. Aprendizaje Semisupervisado | 56 |
| 2.9.6. Aprendizaje Reforzado..... | 57 |
| 2.10. MÉTODOS DE ENSAMBLE EN MACHINE LEARNING | 57 |



| | |
|---------------------------------|----|
| 2.10.1. Bagging | 59 |
| 2.10.2. Gradient Boosting | 60 |
| 2.10.3. XGBoost..... | 62 |

CAPÍTULO III

MATERIALES Y MÉTODOS

| | |
|---|-----------|
| 3.1. MUESTRA DE LA INVESTIGACIÓN..... | 64 |
| 3.2. UBICACIÓN DE LA INVESTIGACIÓN | 64 |
| 3.3. DISEÑO DE LA INVESTIGACIÓN | 64 |
| 3.4. TIPO DE INVESTIGACIÓN | 65 |
| 3.5. SISTEMA DE ADQUISICIÓN DE DATOS | 65 |
| 3.5.1. Panel Solar | 68 |
| 3.5.2. Microcontrolador – Arduino Mega | 70 |
| 3.5.3. Modulo Ethernet..... | 71 |
| 3.5.4. Carga constante | 73 |
| 3.5.5. Sensor de amperaje | 75 |
| 3.5.6. Sensor de temperatura y humedad | 78 |
| 3.5.7. Sensor de iluminancia | 81 |
| 3.5.8. Sensor de radiación UV | 84 |
| 3.5.9. Sensor de voltaje | 86 |
| 3.5.10. Raspberry Pi | 88 |
| 3.5.11. Servidor de internet | 90 |

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

| | |
|--|-----------|
| 4.1. PREPROCESAMIENTO DE DATOS..... | 92 |
| 4.2. MODELOS DE ENSAMBLE | 95 |



| | |
|--|------------|
| 4.2.1. Bagging | 97 |
| 4.2.2. Gradient Boosting | 99 |
| 4.2.3. XGBoost..... | 101 |
| 4.3. COMPARANDO MODELOS | 103 |
| V. CONCLUSIONES | 107 |
| VI. RECOMENDACIONES | 108 |
| VII. REFERENCIAS BIBLIOGRÁFICAS..... | 109 |
| ANEXOS..... | 116 |
| ANEXO 1 - MUESTRA DE LA DATA PREPROCESADA | 116 |
| ANEXO 2 – CODIGO PREPROCESAMIENTO | 117 |
| ANEXO 3 – CODIGO MODELOS DE ENSAMBLE..... | 120 |
| ANEXO 4 – FRAGMENTO DEL ÁRBOL DE DECISIÓN DEL MODELO XGBOOST..... | 125 |

Área: Telecomunicaciones y redes de datos.

Línea: Machine Learning y telecomunicaciones

FECHA DE SUSTENTACIÓN: 08 de noviembre 2022.



ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1: <i>Diagrama de flujo de un sistema de adquisición de datos.</i> | 29 |
| Figura 2: <i>Sensor de temperatura.</i> | 31 |
| Figura 3: <i>Sensor de luz mediante fotoresistencia</i> | 32 |
| Figura 4: <i>Sensor de detección de luz UV.</i> | 33 |
| Figura 5: <i>Módulo de sensor de corriente.</i> | 35 |
| Figura 6: <i>Elementos para un sistema de adquisición de datos.</i> | 36 |
| Figura 7: <i>Convertor Analógico Digital.</i> | 37 |
| Figura 8: <i>Conversión de energía fotovoltaica.</i> | 43 |
| Figura 9: <i>Módulo fotovoltaico encapsulado.</i> | 44 |
| Figura 10: <i>Estructura de soporte convencional.</i> | 46 |
| Figura 11: <i>Batería para sistemas fotovoltaicos.</i> | 46 |
| Figura 12: <i>Convertidor CC/CC.</i> | 47 |
| Figura 13: <i>Inversor para sistema fotovoltaico.</i> | 47 |
| Figura 14: <i>Controlador de carga solar.</i> | 48 |
| Figura 15: <i>Irradiancia y longitud de onda.</i> | 49 |
| Figura 16: <i>Algoritmos de Ensamble.</i> | 58 |
| Figura 17: <i>Algoritmo Bagging.</i> | 59 |
| Figura 18: <i>Arboles de decisión de tipo ensamble.</i> | 62 |
| Figura 19: <i>Algoritmo XGBoost.</i> | 63 |
| Figura 20: <i>Esquema de adquisición de datos.</i> | 66 |
| Figura 21: <i>Diagrama de Flujo del sistema de adquisición de datos.</i> | 67 |
| Figura 22: <i>Sistema de adquisición de datos.</i> | 68 |
| Figura 23: <i>Panel solar de 100W.</i> | 70 |



| | |
|---|----|
| Figura 24: <i>Arduino Mega.</i> | 71 |
| Figura 25: <i>Ethernet Shield W5100.</i> | 72 |
| Figura 26: <i>Diagrama de Arduino Mega y módulo Ethernet Shield.</i> | 72 |
| Figura 27: <i>Carga constante del sistema fotovoltaico.</i> | 73 |
| Figura 28: <i>Diagrama de circuito pictórico de la carga y el Panel Solar.</i> | 74 |
| Figura 29: <i>Diagrama circuito esquemático de la carga y el Panel Solar.</i> | 74 |
| Figura 30: <i>Sensor de corriente ACS712.</i> | 77 |
| Figura 31: <i>Diagrama de circuito pictórico del sensor ACS712.</i> | 77 |
| Figura 32: <i>Diagrama circuito esquemático del sensor ACS712.</i> | 78 |
| Figura 33: <i>Sensor de temperatura y humedad DHT22.</i> | 79 |
| Figura 34: <i>Diagrama de circuito pictórico del sensor DTH22:</i> | 80 |
| Figura 35: <i>Diagrama de circuito esquemático del sensor DTH22.</i> | 81 |
| Figura 36: <i>Sensor de iluminancia – Modulo LDR.</i> | 82 |
| Figura 37: <i>Diagrama de circuito pictórico del sensor LDR.</i> | 83 |
| Figura 38: <i>Diagrama de circuito esquemático del sensor LDR</i> | 83 |
| Figura 39: <i>Sensor de radiación UV - ML8511.</i> | 84 |
| Figura 40: <i>Diagrama de circuito pictórico del sensor UV ML8511.</i> | 85 |
| Figura 41: <i>Diagrama de circuito esquemático del sensor UV ML8511.</i> | 86 |
| Figura 42: <i>Divisor de voltaje.</i> | 87 |
| Figura 43: <i>Diagrama de circuito pictórico del divisor de tensión.</i> | 87 |
| Figura 44: <i>Diagrama de circuito esquemático del divisor de tensión.</i> | 88 |
| Figura 45: <i>Raspberry Pi 3 Modelo B.</i> | 89 |
| Figura 46: <i>Diagrama de circuito pictórico Raspberry Pi 3 Modelo B con modulo Arduino Ethernet Shield.</i> | 90 |
| Figura 47: <i>Data original.</i> | 92 |



| | |
|---|-----|
| Figura 48: <i>Conversión de rangos.</i> | 94 |
| Figura 49: <i>Descripción de la conversión y cálculo de la potencia.</i> | 94 |
| Figura 50: <i>Eliminar variables y guardar data preprocesada.</i> | 95 |
| Figura 51: <i>Cargar y visualizar data.</i> | 96 |
| Figura 52: <i>Regresión con modelo Bagging.</i> | 97 |
| Figura 53: <i>Resultados del modelo Bagging.</i> | 98 |
| Figura 54: <i>Código para graficar valores reales y predichos.</i> | 98 |
| Figura 55: <i>Datos predichos y datos reales.</i> | 99 |
| Figura 56: <i>Regresión con modelo Gradient Boosting.</i> | 99 |
| Figura 57: <i>Resultados del modelo Gradient Boosting.</i> | 100 |
| Figura 58: <i>Datos predichos y datos reales.</i> | 101 |
| Figura 59: <i>Regresión con modelo XGBoost.</i> | 101 |
| Figura 60: <i>Resultados del modelo XGBoost.</i> | 102 |
| Figura 61: <i>Datos predichos y datos reales.</i> | 102 |
| Figura 62: <i>Fragmento del árbol de decisión del modelo XGBoost.</i> | 103 |
| Figura 63: <i>Resultados de la métrica MAE.</i> | 104 |
| Figura 64: <i>Resultados de la métrica RMSE.</i> | 105 |
| Figura 65: <i>Resultados de la métrica MSE.</i> | 105 |
| Figura 66: <i>Resultados de la métrica Accuracy con data de entrenamiento.</i> | 106 |
| Figura 67: <i>Resultados de la métrica Accuracy con data de test.</i> | 106 |



ÍNDICE DE TABLAS

Tabla 1: *Resultados de los modelos.* 104



RESUMEN

Esta investigación logra obtener un prototipo de adquisición de datos mediante sensores, dispositivos embebidos y dispositivos de telecomunicaciones; el prototipo contiene lecturas de sensores que miden constantemente Temperatura, Humedad, Radiación UV e Iluminancia; estas variables mencionadas serán la data de entrada para el modelo o también llamada variables independientes, por otro lado la variable dependiente o data que se quiere calcular para su predicción posterior será la potencia del panel. El prototipo construido en esta investigación funciona correctamente teniendo automatizada la labor de recolección de datos mediante los sensores y posteriormente almacenados en un servidor mediante una base de datos. Posterior a la recolección de datos en dos puntos distintos del departamento de Puno (Puno y Juliaca) se realiza el preprocesamiento de datos limpiando la data y preparándola para ser entrenada con los modelos de ensamble. Teniendo la data limpia y preprocesada se aplicó tres modelos de ensamble los cuales fueron: Bagging, Gradient Boosting y XGBoost; cada uno de estos modelos tuvieron ajuste de hiperparametro para optimizar mucho más el proceso de aprendizaje. Luego de entrenar a los modelos con los datos se concluye que Bagging es el mejor modelo teniendo como resultado en la métrica MAE equivalente a 1.81, por otro lado, el RMSE fue de 4.44, un MSE de 19.73, la precisión del modelo con la data de entrenamiento fue de 98.52% y finalmente la precisión con la data de test alcanzo el 89.41%, el hiperparametro ajustado fue “n_estimators” con un valor igual a 500, el cual es equivalente al número de rondas de refuerzo. La investigación se llevó de forma satisfactoria teniendo un modelo optimo en base al sistema prototipo de adquisición de datos.

Palabras claves: Aprendizaje Automático, Energía Fotovoltaica, Variables Meteorológicas.



ABSTRACT

This research manages to obtain a data acquisition prototype through sensors, embedded devices, and telecommunications devices; the prototype contains readings from sensors that constantly measure Temperature, Humidity, UV Radiation, and Illuminance; These mentioned variables will be the input data for the model or also called independent variables, on the other hand, the dependent variable or data that you want to calculate for its subsequent prediction will be the power of the panel. The prototype built in this research works correctly, having automated the task of data collection through sensors and later stored on a server through a database. After data collection at two different points in the department of Puno (Puno and Juliaca), data preprocessing is performed, cleaning the data and preparing it to be trained with the assembly models. Having the data clean and preprocessed, three assembly models were applied, which were: Bagging, Gradient Boosting, and XGBoost; each of these models had hyperparameter tuning to further optimize the learning process. After training the models with the data, it is concluded that Bagging is the best model, resulting in the MAE metric equivalent to 1.81, on the other hand, the RMSE was 4.44, an MSE of 19.73, and the precision of the model with the data of training was 98.52% and finally, the accuracy with the test data reached 89.41%, the adjusted hyperparameter was "n_estimators" with a value equal to 500, which is equivalent to the number of reinforcement rounds. The investigation was carried out satisfactorily, having an optimal model based on the prototype data acquisition system.

Keywords: Automatic Learning, Photovoltaic Energy, Meteorological Variables.



CAPÍTULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad la emisión de dióxido de carbono entre otros gases incrementa y aceleran el cambio climático dando como lugar al efecto invernadero; las temperaturas del planeta tierra continúan subiendo, las sequías son cada vez más pronunciadas, la descongelación de los hielos va en aumento y como consecuencia también la extinción de especies y fenómenos meteorológicos de mayor grado. Un gran porcentaje de emisiones proviene del sector eléctrico y gran parte de estas dan lugar a la generación de dicha electricidad por medio de combustibles fósiles no renovables. El futuro sostenible significara una gran mejora en masa para los problemas que vivimos en la actualidad; esta cambio implica el reforzamiento de ahorro de energía y contribuir con tecnologías de generación de energía con bajas o nulas emisiones de CO₂ y otros gases tóxicos o contaminantes como también reducir o anular cualquier otro factor contaminante en la etapa de generación de energía eléctrica; se registra que en el 2011 los sistemas de energía sostenible limpia en el área de sistemas fotovoltaicos en el mundo se generaron 85 teravatios/hora. En el Perú de acuerdo a la afirmación de la directora ejecutiva de la Asociación Peruana de Energías Renovables, Paloma Sarria refirió que para el 2030 se puede generar una inversión de 1,470 millones de dólares como también la creación de 7,350 empleos directos e indirectos en esta industria energética. La presente investigación nos ayudará a modelar los factores climatológicos por medio de algoritmos de ensamble para la predicción de energía eléctrica fotovoltaica; esta investigación ayudará a las próximas aplicaciones de sistemas fotovoltaicos y así poder abrir más posibilidades en la implementación de estos sistemas energéticos sostenibles ya que esta fuente de energía



es limpia haciéndola aplicable a múltiples actividades o sistemas que requieran de energía eléctrica constantes; para lo cual la contribución de esta investigación son:

- El diseño de un sistema de adquisición de datos de generación de energía eléctrica fotovoltaica teniendo como variables medidas meteorológicas para que posteriormente sean almacenadas y procesadas.
- Determinar modelos de aprendizaje automático de tipo ensamble para posteriormente predecir la generación de energía eléctrica fotovoltaica mediante las variables meteorológicas ya obtenidas en una data unificada.

1.2. FORMULACIÓN DEL PROBLEMA

¿Cómo realizar los modelos de aprendizaje automático de ensamble para predecir la generación de energía eléctrica fotovoltaica basados en variables meteorológicas en las ciudades de Puno y Juliaca?

En base a esta pregunta identificamos que fuentes renovables de energía eléctrica es una gran demanda a nivel mundial por sus altas soluciones eficientes, factibles, seguras y limpias; en la actualidad no se tiene una fuente sostenible, económica y segura para el desarrollo humano en la vida cotidiana. Las causas para este problema generado provienen de la premisa que este tipo de tecnologías no son usualmente utilizadas por sus altos costos iniciales que esta demanda. En el Perú se impulsa esta iniciativa para poder impulsar el uso de las energías alternativas renovables con un crecimiento de la energía solar en el Perú proyectado al año 2030 generara una inversión de 1.470 millones de dólares con la creación de 7,350 empleos directos e indirecto en la industria informo la ejecutiva Paloma Sarria de la Asociación Peruana de Energías Renovables (Forbes, 2021) lo que se traduce en un incremento considerable de sistemas de energías renovables el cual reducirá en gran parte la utilización de energías por medio de combustibles fósiles



que dañan al medio ambiente. Las consecuencias de esta problemática afecta directamente en las personas todos los días ya que se tiene baja disponibilidad y confianza en una solución con fuentes de energías renovables limpias mientras que actualmente la generación de energías cotidianas daña al medio ambiente, a las personas y continúan generando el calentamiento global; estas fuentes de energía eléctrica provienen de centrales hidroeléctricas, termoeléctricas o uso de combustibles fósiles (Sampieri et al., 2014).

1.3. HIPÓTESIS DE LA INVESTIGACIÓN

1.3.1. Hipótesis General

Los modelos de aprendizaje automático de ensamble podrán predecir la generación de energía eléctrica fotovoltaica basándose en las variables meteorológicas de las ciudades de Puno y Juliaca.

1.3.2. Hipótesis Específicas

- Se puede diseñar un sistema de adquisición de datos de generación de energía eléctrica fotovoltaica y variables meteorológicas.
- Es posible determinar los modelos de aprendizaje automático de ensamble para predecir la generación de energía eléctrica fotovoltaica.

1.4. OBJETIVOS DE LA INVESTIGACIÓN

1.4.1. Objetivo General

Determinar los modelos de aprendizaje automático de ensamble para predecir la generación de energía eléctrica fotovoltaica basados en variables meteorológicas en las ciudades de Puno y Juliaca.



1.4.2. Objetivos Específicos

- Diseñar un sistema de adquisición de datos de generación de energía eléctrica fotovoltaica y variables meteorológicas.
- Determinar los modelos de aprendizaje automático de ensamble para predecir la generación de energía eléctrica fotovoltaica.

1.5. METODOLOGÍA

La investigación presente será de forma experimental ya que se realizarán las pruebas y sistemas prototipo implementado en la ciudad de Puno y Juliaca para así evaluar posteriormente el rendimiento energético con modelos de ensamble. La investigación también será cuantitativa ya que se tendrá etapas de observación y medición de los factores climatológicos como también de los resultados a través de métricas por parte del modelo de ensamble.



CAPÍTULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

Se nota que un estudio hace que la estimación de la irradiación solar sea una obligación para cualquier persona involucrada en la planificación de un jardín solar, ya sea fuera de la red o fuera de la red. Esto incluye aprovechar las energías renovables, reducir los efectos con respecto a los cambios climático y aumentar la cobertura de los servicios eléctricos. Además, se nota que la cantidad de estaciones que se utiliza para la medición no es suficiente para abordar en su totalidad la geografía del área, y como también varias de estas no tienen la capacidad de obtener datos de la radiación solar. Es esa la razón por la cual es importante utilizar modelos basados en estadística, matemática y modelos de inteligencia artificial que se encuentren en la capacidad de pronosticar y predecir la irradiancia solar en base a los datos meteorológicos que se encuentran disponibles. En esta investigación se obtuvo un dataset de la ciudad de Cali y Villavicencio midiendo estaciones ubicadas estratégicamente y para así tener un dataset que fue generado a través de tecnologías API utilizando la API World Weather Online esto en la ciudad de Mocoa. Es por eso que el motivo es estimar o calcular la irradiancia solar utilizando diversas técnicas de aprendizaje automático (Machine Learning) para aplicarla a modelos de regresión como también de clasificación; con el objetivo de evaluar su desempeño. Aunque los investigadores han utilizado el aprendizaje profundo en la mayor parte de su trabajo relacionado para predecir la radiación solar, este estudio muestra que, aunque las redes neuronales son una tecnología ampliamente utilizada, existen otros algoritmos de aprendizaje automático como Random Forest, Vector Support



y también AdaBoost con suficiente precisión para ser utilizado en esta área de investigación (Ordoñez Palacios et al., 2020).

Este proyecto se centra en un problema al que se enfrentan los sistemas fotovoltaicos. Generalmente, en estos sistemas, los sistemas fotovoltaicos son implementados con un grado inclinación y así tener una máxima radiación solar y como consecuencia un mejor rendimiento, sin embargo, no será efectivo durante todo el día y solo una cantidad pequeña de horas, ya que la rotación de la tierra hace de que el haz de luz que impacta en el panel solar es de diferentes valores y por esa razón, tenemos que hay un desperdicio de una gran cantidad de luz como consecuencia de energía. Entonces sobre esa premisa se realiza una técnica para poder ser utilizada en el control de este tipo de aplicaciones implementado mediante redes neuronales artificiales, también de cómo utilizarlas para crear un sistema fotovoltaico más eficiente. En rigor, se realiza un sistema seguidor solar para ser aplicado al panel fotovoltaico haciendo de que este siga al sol en todo momento y conseguir en todo momento la radiación máxima y como consecuencia incrementa y mejora el almacenamiento de potencia en algún tipo de comercio. Dichos sistemas seguidor solar fue implementado con un controlador basado en los algoritmos de redes neuronales artificiales (Andrade Caicedo y Sanyer Espinoza, 2021).

Diseño de un algoritmo de predicción de irradiancia solar de aprendizaje automático desarrollado mediante la aplicación de bibliotecas keras y Tensor Flow para generar un modelo secuencial de una red neuronal LSTM previamente seleccionada una semana después, registrada por una estación meteorológica. medir minuto a minuto las variables climáticas para generar un archivo de medición en formato .CSV, procesar los datos codificados en Google Colab para que el servidor reconozca la información del recurso, para la investigación se utiliza Python como lenguaje de programación utilizado en este caso, los resultados obtenidos son valores de radiación solar para



aproximadamente 6 a. M. a 6:00 PM, serie de tiempo con tiempo relevante, si el valor es cero o se cambian los datos de la orden, el resultado será diferente del resultado real porque la serie de tiempo se aplica a los datos pronóstico por hora y resultados de pronóstico para los próximos siete días consecutivos. El diseño del algoritmo de predicción de la irradiancia solar se aplica en series de tiempo con tensor de flujo para conectar los datos de entrada y los keras al entrenar el algoritmo, se creó una red secuencial con LSTM para facilitar su uso, fácil de probar el comportamiento de los datos, se ha estructurado con 6 neuronas de entrada, una capa oculta y una de salida, para el aprendizaje es necesario escalar los datos para no modificar los diferentes valores. El algoritmo durante el entrenamiento muestra datos de error, pero el algoritmo verá los datos de entrada y para verificar la eficiencia del algoritmo, se decidió ingresar los valores registrados estacionalmente de un nuevo mes que no ve el algoritmo, septiembre se ingresa para pronosticar días consecutivos de octubre, el resultado óptimo se obtiene ingresando los datos filtrados, llenando los campos faltantes, en blanco o vacíos, se hace una comparación entre fechas pronosticadas con mediciones con media comprobando que el valor del error experimental es 22.8 % y la media de los datos de pronóstico de siete días es 20.9%, verifique si los valores están cerca de la unidad pura (Achachi y Fernando, 2021).

El objetivo de esta investigación fue el modelamiento de un algoritmo para hacer la predicción de las fallas aplicadas en los inversores que están instaladas en una planta de energía fotovoltaica y utilizan varias avanzadas técnicas de aprendizaje automático. Esto dependerá de los diversos datos o variables monitoreadas en SCADA para el inversor y de los datos ambientales considerados. Se tiene que los objetivos que persigue esta investigación son: Predecir el estado un inversor el cual este operativo o no en los próximos números de días operativos para poder realizar su respectiva instalación como



también el diagnóstico para posteriormente ejecutar un mantenimiento si así es necesario. Evaluaron el aprendizaje y rendimiento con diferentes modelos de aprendizaje automático y poder determinar cuál es el más adecuado para predecir la falla de estos dispositivos. Una de las tareas futuras podría centrarse en desencadenar este tipo de errores. Para que nuestro modelo ML mejore su rendimiento, es útil dividir los diferentes picos según el modo de error del módulo inversor. Este estudio no se realizó por no contar con la información necesaria de cada incidente para formar un grupo por tipo de incidente. Si se trata de una pieza de ajedrez, el mantenimiento y la sustitución pueden ser más específicos, el diseño será más óptimo. Se desarrollará un modelo para cada tipo de error, de esta manera se obtendrá un mejor resultado porque las variables típicas de cada tipo de error serán diferentes. Por ejemplo, en caso de avería por sobrecalentamiento, una de las magnitudes características sería la diferencia de temperatura entre el interior del inversor y la temperatura ambiente. Es cierto que, al seleccionar las variables para este método de reducción, una de las variables retenidas será el delta de temperatura y el rendimiento obtenido será mucho mayor durante esta descomposición (Daimiel Borrallo, 2020).

Este artículo analiza la viabilidad de utilizar el aprendizaje automático en la predicción meteorológica. Se recopila datos de distintos lugares, repositorios o fuentes como también de la información que genera la API, la cual es generada por un prototipo basado en una estación meteorológica de tamaño pequeño que fue implementado mediante el microcontrolador Arduino como también contrastado con los datos del instituto Meteorológico de Uruguay. Luego se inyecta o implementa un algoritmo diseñado para predecir aplicando modelos de aprendizaje automático (Machine Learning), en base a mediciones de presión, humedad y temperatura, se elaboran variables predictivas de lluvia. También se muestra por medio de una exposición los objetivos de



la investigación, indicadores de consecución verificables, investigación sobre el estado del arte, panorama tecnológico, conclusiones y fases futuras. La aplicación está desarrollada en Azure, se describen detalladamente los modelos predictivos y el desarrollo de la mini estación con Arduino. En conclusión, se muestra que este enfoque de la predicción meteorológica mediante estas técnicas es factible y que es necesario seguir avanzando en la investigación, para poder concluir si la predicción climática es mejorable o no. En esta sección, presentaremos las conclusiones sobre el desarrollo del proyecto, así como los posibles pasos futuros a cubrir. Como proyecto de investigación sobre la viabilidad del uso del aprendizaje automático en la predicción climática, los resultados fueron positivos, logró alcance y se confirmó como posible, aunque con limitaciones, oportunidad de desarrollo. sección de análisis. Tanto la precipitación (con lluvia o sin lluvia) como la precipitación continua (valor) se pueden hacer predicciones en base a los datos meteorológicos de humedad, temperatura y presión barométrica. Es posible construir una buena base de datos con una data de tamaño grande con registros de forma histórica y actuales de un número sustancial de datos o variables meteorológicas, independientemente de que para la tesis se hayan utilizado únicamente las cuatro variables mencionadas. Los resultados resultantes se muestran en Power BI para poder unir la lectura de PWS y también la llamada al modelo y sus resultados de predicción. En cualquier caso, pretendemos dejar claro que todavía no hemos considerado estos resultados como algo que sea realmente aplicable a corto plazo, sino que deberían ser mejorados y utilizados con los métodos establecidos allí para predecir el tiempo. En cuanto a la experiencia en procesamiento de datos (depuración), herramientas y lenguajes de programación utilizados y técnicas de aprendizaje automático, el equipo aún es inexperto, lo que nos obliga a esforzarnos más, con el tiempo porque es un gran desafío (Gnoza Tansini y Barberena Allietti, 2018).



En los tiempos actuales, la electricidad y el consumo de la misma es necesaria en la vida diaria de todos. Por esa razón el consumo de electricidad es ahora una necesidad de magnitud básica en la sociedad. En esta presente investigación se realiza la predicción del consumo energético para edificios en base a Machine Learning. Por lo tanto, en base a una gran cantidad de datos de consumo eléctrico en los edificios, es posible realizar predicciones de consumo utilizando aprendizaje automático. Para esto lo principal es estudiar los diferentes modelos o algoritmos de aprendizaje supervisado como también del no supervisado que puedan ser aplicativos a este tipo de datos consumidores para obtener la mejor predicción ya que muchos de estos datos tienen valores nulos, por lo que el modelo utilizado puede estar sesgado. aplicar a tales predicciones. Además, es posible mejorar la eficiencia energética y aprovechar al máximo las energías renovables reduciendo los costes asociados al consumo eléctrico tradicional (Meneses Díez, 2021). Las sombras con presencia parcial en los paneles solares y su dicha superficie contienen implicaciones para la creación, generación y disipación de electricidad. Para describir el comportamiento de las placas es necesario disponer de curvas de corriente y tensión (IV) compatibles con la presencia de cielos parcialmente con sombras no uniformes. La investigación se basa en poder hallar un algoritmo de tipo regresión en base del aprendizaje automático para predecir la curva IV y así poder aproximar cada punto de inflexión actuales, generados al activar los diodos de protección del panel en caso de sombra. Primero, se desarrolló una función de sombreado que toma un vector numérico con un radio de sombra para cada celda del panel. Luego, se estudiaron en detalle 3 modelos de regresión (Nearest Neighbor, Random Forest y SubTree). Toman como entrada las características del color, temperatura e iluminancia desarrollados; y salida 8 pares interpolados de puntos IV. Concluyendo se pudo desarrollar un simulador el cual podrá validar el modelo y así obtener una curva IV basada en un patrón de sombreado



especificado por el usuario. Se puede realizar una cuantificación precisa de la sombra en cada parcela y se ha observado el algoritmo original de cuadrados medios, que es el mejor Random Forest. Las características con respecto a la sombra en el algoritmo para realizar las predicciones, así como la aproximación de la curva IV, son aportes valiosos frente a otros estudios en este campo. El desarrollo del trabajo explicado en este artículo lleva a la conclusión de que una buena característica de sombreado es fundamental para determinar que las curvas de corriente y tensión de los paneles solares se verán afectadas. Aquí, el número de celdas sombreadas y la fuerza de la pelota juegan un papel fundamental. Las técnicas estándar propuestas en este programa en papel son suficientes para determinar el área sombreada y la intensidad de las sombras, ya que logran exhibir un comportamiento cercano a la sustracción de las observaciones de la imagen. El algoritmo de regresión elegido, Random Forest Regression, constituye un algoritmo completo para el problema cubierto en este proyecto al generar una decisión de ejecutar múltiples factorías de ejecución en paralelo, en caso de existir cuando la cantidad de datos es deficitaria. El algoritmo tiene un tiempo de ejecución bajo (casi 0,05 segundos) y la probabilidad de error más baja entre los algoritmos estudiados para el componente de voltaje (promedio 0,77), así como el error incorrecto de la componente de línea (promedio 0,72). Aunque el error obtenido sigue siendo alto, esto se explica por la falta de datos para entrenar el modelo y, por lo tanto, se espera que su rendimiento pueda mejorar significativamente entrenando con bases de datos más grandes que nunca. El simulador desarrollado para obtener curvas sombreadas de tensión y corriente definidas por el usuario es una herramienta muy útil, ya que puede predecir curvas IV de forma fácil, rápida y económica en comparación con los trazadores de corriente y tensión. Valdrá la pena utilizar este simulador una vez que se disponga de un modelo adecuado y se haya comparado su rendimiento con los plotters actuales y sus simuladores. El trabajo



realizado, así como la investigación en curso en esta área, es de gran utilidad en el campo de la producción de energía, ya que es un factor importante en la optimización de los sistemas fotovoltaicos porque 'facilita la predicción del consumo de energía de producción de electricidad, que actualmente no tiene en cuenta la influencia de la oscuridad. Se necesitan más investigaciones y estudios para aumentar el uso de la energía solar en el mundo (Castellanos Taita, 2020).

Las estimaciones precisas de las producciones como también el consumo energético son clave para impulsar la forma de integrar fuentes energéticas en base a sistemas renovables directamente conectadas a la red, guiando el equilibrio del suministro de energía a través de ciclos y fluctuaciones inherentes a estas fuentes. En este escenario, muchos experimentos han desarrollado métodos de predicción de la radiación solar, pero pocos de ellos aplican redes neuronales cíclicas y su potencial de modelado de series temporales. Esta tesis revisa varios estudios y describe un experimento que compara redes neuronales repetidas, XGBoost y la existencia de índices Clear Sky y Clearness en las predicciones de irradiancia solar. La precisión del modelo se verificó con observaciones cuidadosas de Denver y Las Vegas, se volvieron a muestrear a 5 min, 30 min, 1 hora y media de resolución todos los días. XGBoost proporciona los mejores resultados en todos los horizontes de pronóstico, con nRMSE de 12. La resolución es de 9 minutos y la media diaria es de 21,2 minutos. Los modelos de fuerza para los índices Clear Sky y Clarity muestran una precisión comparable, con nRMSE que van del 1,2 % al 22,5 % para pronósticos de hasta 1 hora de anticipación. Los RNN superan a los que existen en las proyecciones de irradiancia media diaria, alcanzando nRMSE 21,3% (Dapper Heinen, 2018).

El diagnóstico médico es un tema complejo de suma importancia y poder así cuidar de la salud. Los sistemas multiagente presentan un medio para modelar, a través



de un modelo informático, el proceso. El aspecto que actualmente es foco de investigación es cómo se integran o involucran las decisiones de los diferentes actores para llegar a la decisión final. Un modelo propuesto consiste en una federación de agentes racionales que cooperan para dividir el problema de diagnóstico en varias etapas y sintetizar las decisiones de varias redes neuronales diferentes, más o menos especializadas, diferenciadas y máquinas de vectores de soporte, según cuatro modelos: votación simple, estratificado votación simple, votos ponderados e impulsos. El modelo se ha aplicado a los síndromes coronarios agudos, incluido el infarto de miocardio, la principal causa de muerte en todo el mundo (Díaz y Guerrero, 2018).

Las fallas de equipos de tipo industrial son acciones importantes en el contexto de cualquier empresa u organización. Se nota que la clasificación y características representan un valor importante para tomar decisiones en las operaciones que contienen el mantenimiento de estas. Por esa razón la minería de datos juega uno de los papeles importantes para poder evaluación como también de clasificación los errores informados. Se utilizan modelos basados en árboles de decisión y también en redes bayesianas por separado para construir modelos clasificadores híbridos con el objetivo de evaluar las características de los defectos. Este proyecto propone desarrollar modelos híbridos utilizando métodos de votación de clasificador y ensamblaje, combinando modelos de redes bayesianas (Bayesnet y Naive BaySp Compatible) como también árboles de decisión (RandomTree). La precisión de los métodos de coincidencia está determinada por diferentes algoritmos, por pruebas con el mismo conjunto de datos particionados (Corso et al., 2017).



2.2. SISTEMA DE ADQUISICIÓN DE DATOS

Aunque conceptos como adquisición de datos y medición pueden ser sorprendentemente difíciles de definir por completo, la mayoría de los usuarios de computadoras, ingenieros y científicos están de acuerdo en que hay varios elementos comunes como una computadora personal (PC) se utiliza para programar equipos de prueba y manipular o almacenar datos. El término "PC" se usa en un sentido general para incluir cualquier computadora que ejecute cualquier sistema operativo y software que admita el resultado deseado. La PC también se puede usar para funciones de apoyo, como gráficos en tiempo real o generación de informes. Es posible que la PC no tenga necesariamente un control constante del equipo de adquisición de datos o incluso que permanezca conectada al equipo de adquisición de datos en todo momento. El equipo de prueba puede consistir en tarjetas enchufables de adquisición de datos para PC, chasis de tarjetas externas o instrumentos discretos. El chasis externo y los instrumentos discretos generalmente se pueden conectar a una PC utilizando puertos de comunicación estándar o una placa de interfaz patentada en la PC. El equipo de prueba puede realizar uno o más procesos de medición y control utilizando varias combinaciones de entrada analógica, salida analógica, E/S digital u otras funciones especializadas. La dificultad que implica diferenciar términos como adquisición de datos, prueba y medición, y medición y control surge de los límites borrosos que separan los diferentes tipos de instrumentación en términos de funcionamiento, características y rendimiento. Por ejemplo, algunos instrumentos independientes ahora contienen ranuras para tarjetas y microprocesadores, usan software de sistema operativo y funcionan más como computadoras que como instrumentos tradicionales. Algunos instrumentos externos ahora hacen posible construir sistemas de prueba con un alto número de canales que recopilan datos y los registran en una computadora de control. Los tableros enchufables pueden transformar las

computadoras en multímetros digitales de rango múltiple, osciloscopios u otros instrumentos, completos con paneles frontales virtuales en pantalla fáciles de usar. El hardware de adquisición y control de datos está disponible en varias formas, que ofrecen diferentes niveles de funcionalidad, cantidad de canales, velocidad, resolución, precisión y costo. Esta sección resume las características y los beneficios generalmente asociados con las diversas categorías, con base en una amplia muestra representativa de productos (Ramírez Ríos, 2016).

2.2.1. Adquisición y control de datos en tiempo real

El control crítico en tiempo real es un tema importante en los sistemas de control y adquisición de datos. Las aplicaciones que exigen control en tiempo real normalmente se adaptan mejor a los sistemas externos que a los sistemas basados en tarjetas de conexión de PC. Aunque Microsoft Windows se ha convertido en el sistema operativo estándar para aplicaciones de PC, es un sistema operativo no determinista que no puede proporcionar tiempos de respuesta predecibles en aplicaciones críticas de medición y control. Por lo tanto, la solución es vincular la PC a un sistema que pueda operar de manera autónoma y brindar respuestas rápidas y predecibles a estímulos externos. Los sistemas de adquisición de datos (DAQ) son los principales instrumentos utilizados en la investigación de laboratorio por parte de científicos e ingenieros; en particular, para prueba y medición, automatización, etc. Por lo general, los sistemas DAQ son instrumentos DAQ de uso general que son adecuados para medir señales de voltaje o corriente. Sin embargo, muchas señales de salida de sensores y transductores deben ser acondicionadas antes de que una placa pueda adquirir y transformar en digital la señal (Lad et al., 2017). Los elementos básicos de DAQ se muestran en la Fig. 1.

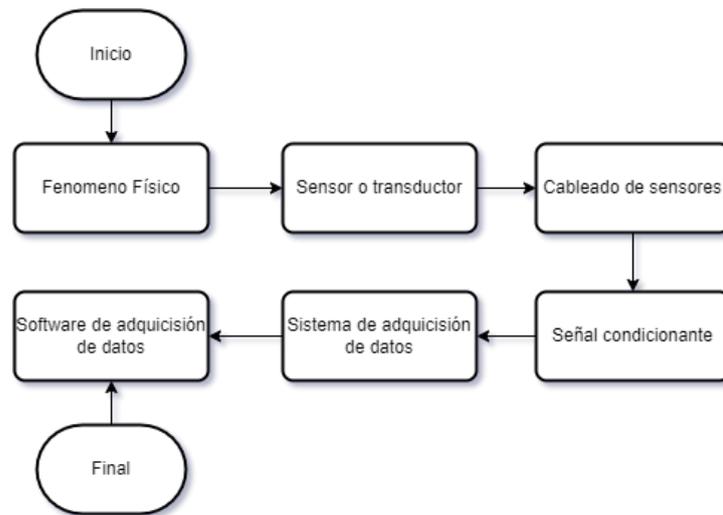


Figura 1: Diagrama de flujo de un sistema de adquisición de datos.

Elaboración propia.

2.3. SENSORES

Todos los sistemas de procesamiento industrial, fábricas, maquinaria, instalaciones de prueba y vehículos consisten en componentes de hardware y software de computadora cuyo comportamiento sigue las leyes de la física tal como las entendemos. Estos sistemas contienen miles de fenómenos mecánicos y eléctricos que cambian continuamente; no son de estado estacionario. Las cantidades medibles que representan las características de todos los sistemas se denominan variables. El buen funcionamiento de un sistema en particular depende de ciertos eventos en el tiempo y de los parámetros de estas variables. A menudo, estamos interesados en la ubicación, magnitud y velocidad de las variables, y usamos instrumentos para medirlas. Asignamos a las variables unidades de medida como voltios, libras y millas por hora, por nombrar algunas (Emilio, 2013). La mayoría de las variables deben medirse con un dispositivo que convierta los fenómenos en una forma que un ser humano pueda percibir, como una pantalla visual, un transductor de sonido o vibraciones para estimular sensaciones físicas. Los dispositivos



de conversión se denominan transductores o sensores y traducen los fenómenos físicos en señales eléctricas (o viceversa) para ser medidas con instrumentos electrónicos. Estos instrumentos han sido tradicionalmente amperímetros, voltímetros y varios otros medidores, y las variables se pueden observar en tiempo real. Pero una necesidad creciente de registrar y preservar estos fenómenos y analizarlos en un momento posterior obligó a los ingenieros a desarrollar registradores de datos y sistemas de adquisición de datos. Las variables se pueden clasificar de muchas maneras, pero, en general, la mayoría de los expertos prefieren dos clasificaciones: por característica y por tipo de señal de medición. Las variables clasificadas por característica incluyen térmica, radiación, fuerza, velocidad, cantidad, tiempo, geometría, propiedades físicas, composición química y eléctrica. Los clasificados por señal de medición incluyen movimiento, fuerza, eléctrica y modulada en el tiempo. Las señales de medición para variables a menudo son difíciles de diferenciar del sistema de medición. Hay cuatro factores que requieren una cuidadosa consideración para las señales y los sistemas de medición: los tipos de transductores disponibles para convertir variables en señales de medición, las características de transmisión, la coincidencia de entrada del sistema de adquisición de datos y los transductores disponibles para convertir de un tipo de señal de medición a otra señal de medición (Lad et al., 2017).

2.3.1. Sensor de temperatura

Actualmente se utilizan varias técnicas para la detección de la temperatura. Los más comunes son los detectores de temperatura de resistencia (RTD), los termopares, los termistores y los circuitos integrados de sensores. La elección de uno para su aplicación puede depender de algunos factores como: rango de temperatura requerido, linealidad, precisión, costo y características. Los RTD son más conocidos; están contruidos con varios materiales diferentes para el elemento sensor, por ejemplo, el platino. El platino se

usa por diferentes razones: clasificación de alta temperatura, muy estable y muy repetible. Otros materiales utilizados para los sensores RTD son el níquel y el cobre. El termopar está compuesto por dos metales diferentes que tienen un punto de contacto común donde se produce un voltaje (algunos mV) proporcional a la variación de la temperatura. Los termistores generalmente están compuestos de materiales semiconductores. Hay termistores con coeficiente de temperatura positivo y negativo. Los termistores con coeficiente de temperatura negativo se utilizan para monitorear bajas temperaturas del orden de 10 K (Emilio, 2013), un ejemplo de este tipo de sensores de temperatura se muestra en la Fig. 2.

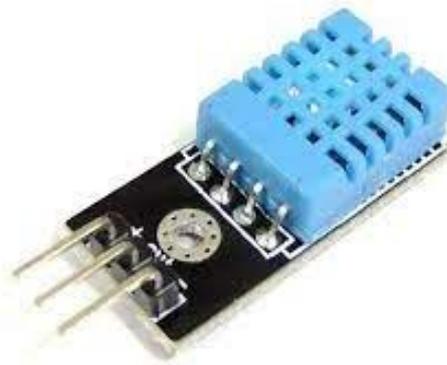


Figura 2: *Sensor de temperatura.*

Fuente: Cruz, 2022.

2.3.2. Sensor de luz

Los sensores de luz detectan la luz emitida o desprendida de un objeto, como un LED, reflejada en superficies, transmitida desde un dispositivo electrónico, etc. El LED o diodo emisor de luz es un semiconductor de estado sólido que emite luz cuando la corriente lo atraviesa en dirección directa. Un sensor fotoeléctrico es un dispositivo eléctrico que responde al cambio en la intensidad de la luz que cae sobre él. Hay muchas situaciones de detección en las que el espacio es demasiado restringido o el entorno es demasiado hostil, incluso para los sensores remotos. La fibra óptica es una tecnología

alternativa en el "empaquetado" de sensores para aplicaciones tales como la tecnología de detección fotoeléctrica. Además, la fibra óptica es una fibra flexible y transparente hecha de vidrio (sílice), estos sensores tienen un aspecto circular como se ve en la Fig. 3.

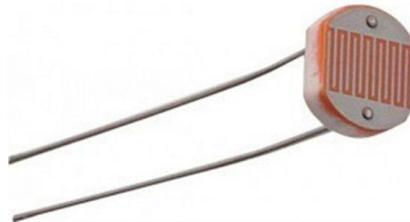


Figura 3: *Sensor de luz mediante fotoresistencia*

Fuente: PROMETEC, 2022.

2.3.3. Sensor de luz ultravioleta

La tecnología de producción en masa para sensores de luz ultravioleta (UV) basados en silicio, adecuada para teléfonos inteligentes y dispositivos portátiles en la era de Internet de las cosas (IoT), ha sido desarrollada conjuntamente por un equipo de investigación de la Universidad de Tohoku y SII Semiconductor Corporation, un fabricante de semiconductores de Seiko. Grupo de Instrumentos. En los últimos años, ha habido un interés creciente dentro de la comunidad de atención médica en la prevención de las quemaduras solares y las imperfecciones de la piel. Como tal, la fácil medición de la luz ultravioleta mediante el uso de un teléfono inteligente o un dispositivo portátil podría ser de gran beneficio para la salud y la medicina estética. De hecho, la necesidad de medir la luz ultravioleta invisible también está aumentando en los campos industriales, donde los equipos como las máquinas de curado ultravioleta y las impresoras que utilizan tintas curables ultravioleta se utilizan ahora con más frecuencia que nunca. La nueva

tecnología de sensor de luz ultravioleta, desarrollada por el equipo de investigación dirigido por el profesor Shigetoshi Sugawa y el profesor asociado Rihito Kuroda en la Escuela de Graduados de Ingeniería de la Universidad de Tohoku, utiliza solo semiconductores de silicio para detectar y medir selectivamente la intensidad de la luz UV-A (315 ~ 400 nm) y bandas de onda de luz UV-B (280~315nm). Estas son las bandas a las que se atribuyen las quemaduras solares y las manchas en la piel. Los sensores de semiconductores de silicio versátiles se adaptan mejor a las integraciones con circuitos y agregan más funciones que los sensores UV de semiconductores compuestos. Convencionalmente, los sensores de luz UV de fotodiodo de silicio emplean filtros ópticos que cortan las bandas de luz visible no deseadas. Al utilizar la respuesta espectral diferencial de los fotodiodos de silicio con sensibilidades a la luz ultravioleta altas y bajas, los investigadores pudieron desarrollar un sensor con capacidades de detección selectiva de rango ultravioleta sin emplear un filtro óptico. La estructura óptica sin filtro obtiene una mayor sensibilidad al evitar una disminución de la intensidad de la luz ultravioleta incidente en el sensor, en la Fig. 4 se muestra un modelo de este tipo de sensores (Outsourcing, 2017).



Figura 4: *Sensor de detección de luz UV.*

Fuente: Robotop, 2022.



2.3.4. Sensor de corriente

Un sensor de corriente es un dispositivo que detecta y convierte la corriente en un voltaje de salida fácilmente medible, que es proporcional a la corriente a través de la ruta medida. Existe una amplia variedad de sensores, y cada sensor es adecuado para un rango de corriente específico y una condición ambiental. Entre estos sensores, una resistencia de detección de corriente es la más utilizada. Se puede considerar un convertidor de corriente a voltaje, donde al insertar una resistencia en la ruta de la corriente, la corriente se convierte en voltaje de forma lineal (Fig. 5). La tecnología utilizada por el sensor de corriente es importante porque diferentes sensores pueden tener diferentes características para una variedad de aplicaciones. Los sensores de corriente se basan en tecnología de efecto Hall de bucle abierto o cerrado. Un sensor de bucle cerrado tiene una bobina que se impulsa activamente para producir un campo magnético que se opone al campo producido por la corriente que se detecta. El sensor Hall se utiliza como un dispositivo de detección nulo y la señal de salida es proporcional a la corriente que se conduce a la bobina, que es proporcional a la corriente que se mide. En un sensor de corriente de bucle abierto, el flujo magnético creado por la corriente primaria se concentra en un circuito magnético y se mide con un dispositivo Hall. La salida del dispositivo Hall es la señal acondicionada para proporcionar una representación exacta (instantánea) de la corriente primaria (Emilio, 2013).



Figura 5: *Módulo de sensor de corriente.*

Fuente: Cablematic, 2022.

2.4. CONVERTIDOR ANALÓGICO DIGITAL Y DIGITAL ANALÓGICO

En el corazón del sistema DAQ se encuentra la conversión de datos, el proceso de convertir señales analógicas en formatos digitales y viceversa. La conversión se realiza en dos pasos: primero se cuantifica la señal y luego se codifica. Medios de cuantificación que representan los valores continuos de la señal analógica utilizando un conjunto de valores discretos, y medios de codificación que representan estos valores discretos por secuencias de bits. El número de bits de estas secuencias determina el número de valores posibles de la conversión: 2^n para n bits. Como ejemplo, la figura 19.3 ilustra la digitalización de una forma de onda de 10 V con 3 bits de resolución. A/D y D/A proporcionan la interfaz entre el mundo analógico (el mundo real) y el mundo de la computación digital y el procesamiento de datos. Muchas de las aplicaciones de las interfaces analógico-digitales son bastante evidentes cuando las usamos en nuestra vida diaria. Los controladores de temperatura, las computadoras y los periféricos relacionados, la inyección electrónica de combustible y los sistemas de música con discos compactos son solo una muestra de las aplicaciones típicas que requieren interfaces analógico-digitales. El circuito electrónico de A/D realiza los procesos de cuantificación y

codificación. El A/D opera con señales de amplitud determinada, por ejemplo, entre $-SV$ y $+SV$. Otro circuito electrónico de D/A realiza la conversión de señales digitales a señales analógicas. Este dispositivo convierte códigos digitales de n bits en una señal de $2n$ niveles discretos de voltaje o corriente. Primero veremos D/A, ya que suelen ser más simples que los A/D y, a menudo, se utilizan como un componente dentro de un A/D (Lipták, 2012).

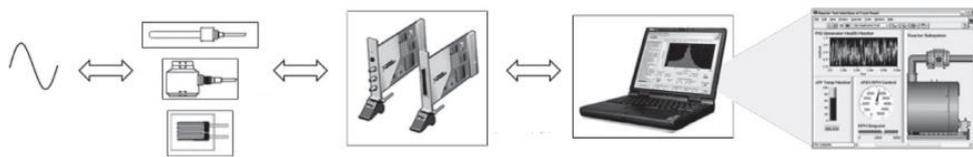


Figura 6: Elementos para un sistema de adquisición de datos.

Fuente: Lipták, 2012.

2.4.1. Convertidores de analógico a digital

Los A/D, por supuesto, realizan la operación inversa de los D/A, convirtiendo las señales de entrada analógicas en una representación binaria digital. El número de bits en esta representación digital determina la resolución. Un A/D con resolución de n bits puede generar una salida con uno de 2 elevado a n códigos distintos. Cuando se combina con el rango de entrada, o intervalo, del A/D, la resolución indica el cambio más pequeño que se puede detectar en la señal de entrada o el valor de un bit menos significativo (LSB). La Tabla 19.1 compara las resoluciones comúnmente disponibles en la actualidad para algunos rangos de entrada típicos. Hay muchos tipos de A/D en uso práctico y la razón de esto es que hay un número diverso de aplicaciones. En última instancia, la aplicación determina los requisitos y es posible que la resolución del convertidor A/D no sea necesaria o suficiente para especificar el rendimiento requerido. En muchos casos, la calidad del convertidor A/D debe probarse para la aplicación específica. La amplia

variedad de aplicaciones de ADC conduce a una gran cantidad de cifras de mérito para especificar el rendimiento. Estas figuras de mérito incluyen precisión, resolución, rango dinámico, compensación, ganancia, línea diferencial, no linealidad integral, relación señal a ruido, relación señal a ruido y distorsión, número efectivo de bits, rango dinámico libre de espurias, distorsión de intermodulación, distorsión armónica total, ancho de banda de resolución efectivo, ancho de banda de potencia completa, ancho de banda lineal completo, retardo de apertura. El costo de un ADC es proporcional a los siguientes parámetros: precisión, número de bits y estabilidad. Pero incluso el ADC más costoso puede comprometer la precisión cuando el ruido excesivo interfiere con la señal de entrada, ya sea que la señal esté en milivoltios o mucho más grande. Una técnica para reducir el ruido y garantizar la precisión de la medición es eliminar los bucles de tierra que pueden ocurrir cuando dos o más dispositivos están conectados a terminales de tierra en diferentes ubicaciones (Lipták, 2012). La señal de entrada es continua en el tiempo y es necesario definir una tasa de acuerdo a la extrapolación de nuevos valores digitales a partir de la señal analógica. La tasa de valor nuevo se denomina tasa de muestreo o frecuencia de muestreo del convertidor como se muestra en la Fig. 7.

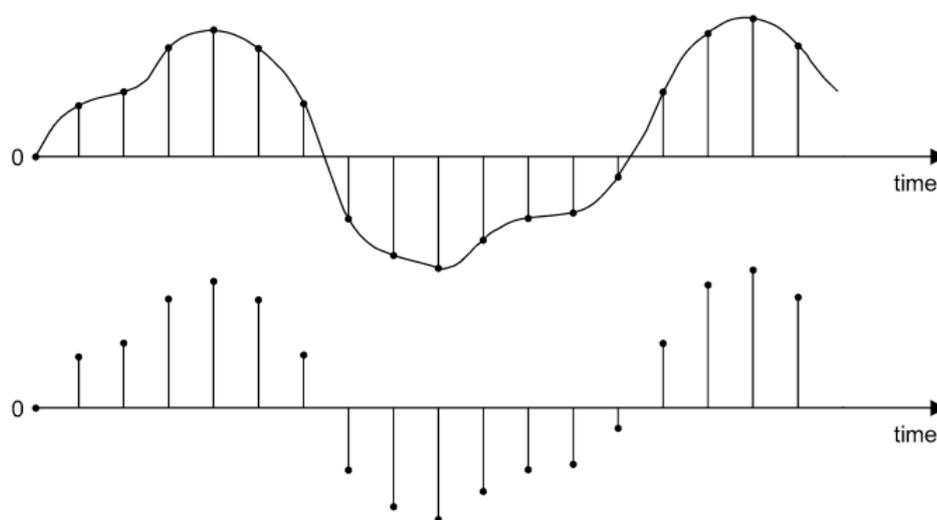


Figura 7: *Convertor Analógico Digital.*

Fuente: Emilio, 2013.



2.5. CONECTIVIDAD

El tema de la conectividad incluye una variedad de puertos y buses externos que están disponibles en las PC, ya sea como equipo estándar o como opciones de interfaz enchufable. Estos incluyen puertos de impresora paralelos, el puerto serial tradicional RS-232, los puertos seriales RS-422 y RS-485, IEEE-488 y buses de alta velocidad más nuevos como el Universal Serial Bus (USB), bus IEEE 1394 FireWire, y Ethernet (Lipták, 2012).

2.5.1. Puertos Serie (RS-232, RS-422, RS-485)

El estándar serial RS-232 de la Electronic Industries Association (EIA) es uno de los protocolos de comunicación más comunes. Introducido en 1962, el RS-232 se convirtió en una de las primeras características comunes en computadoras centrales y minicomputadoras, terminales de datos, impresoras, módems y otros equipos de datos mucho antes de que se desarrollaran las PC. RS-232 se especificó originalmente para comunicaciones de datos entre un solo dispositivo controlador y un solo receptor en distancias de hasta 15 metros (50 pies), a velocidades de hasta 19,200 bits por segundo. Estos límites de distancia y velocidad han demostrado ser conservadores; RS-232 ahora se usa rutinariamente a velocidades de 115 a 230 kbits/segundo. RS-232 también ha sido operado a distancias superiores a 25 metros. La capacidad de lograr estos parámetros operativos depende del entorno y la calidad del cableado utilizado. En términos generales, un solo sistema no puede proporcionar tanto la velocidad máxima como la distancia máxima. El conector RS-232 tradicional era un diseño D-shell de 25 pines. Sin embargo, muchos productos seriales ahora incluyen puertos seriales de 9 pines con (o en lugar de) la versión de 25 pines. Ambos tipos de puertos contienen las mismas nueve líneas de señal, aunque los números de pin para señales específicas son diferentes. Un enlace RS-



232 completo incluye Tx (transmisión), Rx (recepción), tierra de señal y varias líneas de protocolo de enlace adicionales. El protocolo de enlace es importante con algunos periféricos (como los módems) para controlar la transmisión y evitar el desbordamiento de datos. Es posible establecer comunicación en serie para algunas aplicaciones usando solo Tx, Rx y señal de tierra, en cuyo caso, los pines de protocolo de enlace en cada pieza del equipo en serie deben terminarse con varias conexiones loop-back y pull-up. Consulte la documentación de cualquier dispositivo RS-232 que se vaya a utilizar para obtener información sobre el pin-out y la conexión. Un segundo estándar serial popular, RS-422, es similar a RS-232, excepto que usa transmisión de datos diferencial. Este método emplea dos líneas activas para transmitir una señal, en lugar de la línea de señal y tierra de RS-232. La transmisión de datos diferencial proporciona un rechazo de ruido superior, lo que admite velocidades de datos más altas y mayores distancias operativas. Son posibles velocidades de transmisión de datos de hasta 115 kbits/segundo y distancias de hasta 4000 pies. Al igual que con RS-232, la velocidad máxima y la distancia máxima son mutuamente excluyentes. Las designaciones de los pines del conector para RS-422 difieren de las de RS-232, por lo que los puertos y equipos que utilizan estos estándares son incompatibles. La capacidad del RS-422 para manejar aplicaciones multipunto, donde un dispositivo controlador puede transmitir hasta diez receptores, es una ventaja significativa sobre el RS-232. RS-485 es otro estándar serial popular, que combina elementos de RS-422 con la capacidad de manejar múltiples receptores y múltiples transmisores en un bus. Al igual que RS-422, RS-485 utiliza transmisión de datos diferencial y puede funcionar a velocidades de hasta 115 kbits/segundo en distancias de hasta 4000 pies. Además, RS-485 puede admitir hasta 32 dispositivos de controlador y 32 receptores, lo que hace posible construir una red multipunto de transmisores y receptores utilizando un solo tramo de cable RS-485 (Keithley, 2001). Con la posible



excepción de Ethernet, RS-422 y RS-485 brindan la mejor combinación de distancia y facilidad de uso para aplicaciones de adquisición de datos. Al igual que con RS-232, lograr una comunicación de larga distancia sin errores depende de la calidad del cable y las condiciones ambientales. A distancias inferiores a 4000 pies, las velocidades de transmisión permitidas aumentan, con velocidades de hasta 10 megabits/segundo posibles a 50 pies. El uso de cableado de par trenzado blindado de alta calidad es fundamental para lograr un rendimiento óptimo. Las interfaces enchufables RS-422 y RS-485 para buses ISA y PCI están fácilmente disponibles. Muchos instrumentos de sobremesa y algunos productos de adquisición de datos externos incluyen un puerto RS-232 o IEEE-488, los cuales se pueden convertir a RS-422 o RS-485 utilizando el adaptador adecuado. Los puertos RS-422 y RS-485 son más comunes en los sistemas industriales, donde el soporte para la adquisición de datos a larga distancia es una preocupación principal (Keithley, 2001).

2.6. ENERGÍA FOTOVOLTAICA

La energía fotovoltaica (PV) es la conversión directa de la radiación en electricidad. Si bien la electricidad se utilizará cada vez más como fuente de energía, la fotovoltaica desempeñará un papel importante en el campo de las energías renovables. La tecnología fotovoltaica es modular (es decir, los sistemas existentes son ampliables), tiene una larga vida útil (los fabricantes ofrecen garantías de hasta 25 años), es silenciosa y libre de emisiones durante su uso. Existe un potencial considerable para la reducción de costos debido a la tecnología de semiconductores conocida; además, los procesos de producción actuales se pueden hacer más eficientes y económicos desarrollando técnicas de producción en masa. Aunque en la actualidad, los precios en las fábricas p están estabilizados en el entorno de los 3 €/W, por falta de silicio como materia prima. Se espera que después de que entren en funcionamiento las nuevas instalaciones de producción de



silicio en 2006, los precios volverán a caer. La información más reciente indica que los costes de producción p de 1–1,6 €/W son factibles (Stefan, 2006). Los sistemas fotovoltaicos pueden ser pequeños y muy simples, que consisten solo en un módulo fotovoltaico y una carga, como en la alimentación directa de un motor de bomba de agua que solo necesita funcionar cuando brilla el sol. Por otro lado, los sistemas fotovoltaicos también se pueden construir como grandes centrales eléctricas con una potencia máxima de varios MW; estos están conectados a la red eléctrica. Muchos sistemas se colocan en casas residenciales. Cuando toda una casa necesita energía y no está conectada a la red eléctrica, el sistema fotovoltaico debe estar operativo día y noche. También puede tener que alimentar cargas de CA y CC, tener energía de reserva e incluso puede incluir un generador de respaldo. Según la configuración del sistema, podemos distinguir tres tipos principales de sistemas fotovoltaicos: autónomos, conectados a la red e híbridos. Los principios y elementos básicos del sistema fotovoltaico siguen siendo los mismos. Los sistemas se adaptan para satisfacer requisitos particulares variando el tipo y la cantidad de los elementos básicos. Un diseño de sistema modular permite una fácil expansión cuando cambia la demanda de energía (Arno et al., 2016).

2.6.1. Efecto fotovoltaico

Una celda solar es un diodo semiconductor de gran área. Consiste en una unión p-n creada por una adición de impurezas (dopaje) en las células solares de silicio de cristal semiconductor comúnmente utilizadas). Si las impurezas son átomos de fósforo, que tienen cinco electrones externos, solo se requieren cuatro electrones para encajar el átomo en la estructura cristalina de silicio, el quinto electrón es móvil y libre. Entonces, en esta región del cristal hay muchas (la mayoría) cargas negativas libres, por lo que se llama región n. Viceversa para la región p: al dopar el cristal con átomos de boro, que tienen solo tres electrones externos, siempre falta un electrón para una unión completa en la



estructura cristalina. Este electrón podría ser "prestado" de átomos vecinos, por lo que el lugar del electrón faltante se desplaza. Este electrón perdido también podría verse como un "agujero" con una carga positiva que es móvil y errante. Hay muchos más agujeros libres que electrones libres en las regiones p, por lo que los electrones se denominan portadores de carga minoritarios (Stefan, 2006), este concepto esta ilustrada en la Fig. 8.

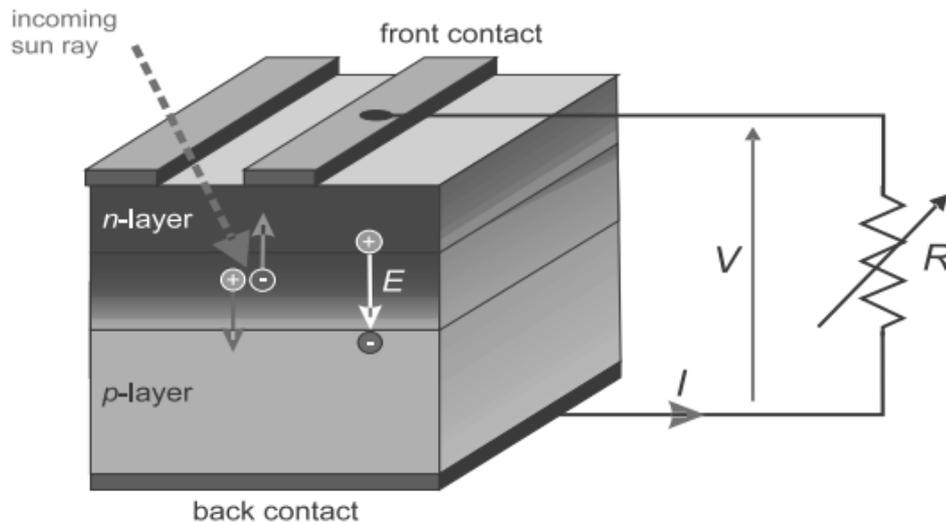


Figura 8: *Conversión de energía fotovoltaica.*

Fuente: Stefan, 2006.

2.6.2. Generador fotovoltaico

El campo eléctrico interno en una celda solar es relativamente débil y solo se pueden lograr pequeñas diferencias en el potencial eléctrico (0,3 V para germanio y 0,7 V para silicio). El voltaje de circuito abierto real que podría lograrse es

ligeramente inferior a estos valores. Para obtener voltajes más altos, las células solares se conectan en serie, las llamadas "cadenas" de células solares. Debido a que estas cadenas son muy frágiles, por lo general se incrustan en un sándwich de plástico blando y vidrio, lo que crea el llamado "módulo solar" o "módulo fotovoltaico". El plástico blando transparente por encima y por debajo de las cadenas de celdas es normalmente el copolímero EVA (etileno-acetato de vinilo), pero también son posibles opciones el PVB, las siliconas o el TPU (poliuretano termoplástico, de desarrollo muy reciente). Para endurecer el compuesto y hacerlo más duradero, se agrega un vidrio frontal templado. Si la parte trasera consiste en una lámina compuesta (que consiste en una combinación de capas como PVF-aluminio-PVF o PVF-poliéster-PVF), lo llamamos "módulo laminado";

si la parte trasera es de vidrio, lo llamamos “módulo encapsulado” como se puede ver en la Fig. 9.

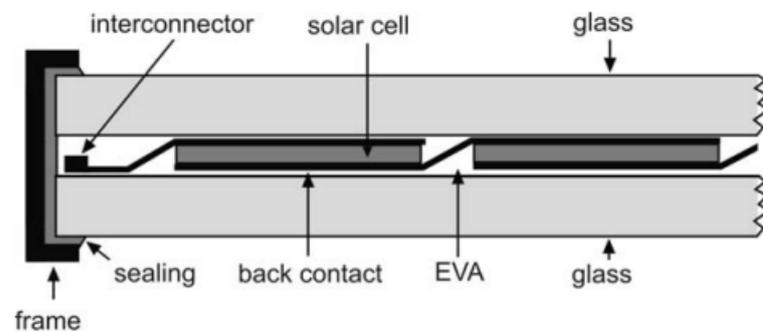


Figura 9: Módulo fotovoltaico encapsulado.

Fuente: Stefan, 2006.

2.6.3. Fabricación de celdas solares

Nos limitaremos a la fabricación de células solares de Si monocristalinas y multicristalinas basadas en obleas. Los principales pasos de fabricación son los siguientes (Arno et al., 2016):

- Las obleas de tipo p con una resistividad de $1e10 \text{ U cm}$ y un grosor de aproximadamente 0,2 mm se difunden primero con fósforo para producir una capa n+ con un grosor de 0,1e0,2mm. Para llevar a cabo este proceso se utiliza un horno de difusión. El gas de protección es nitrógeno. Las fuentes de difusión son sólidas en forma de discos o fuentes líquidas giratorias. El proceso puede funcionar bajo una atmósfera de oxígeno.
- Luego, las obleas difundidas se graban con ácido fluorhídrico para eliminar las capas de la fuente de difusión.
- A continuación, las obleas se metalizan mediante: Evaporación de níquel-paladio-plata utilizando evaporadores térmicos o de haz de electrones. La evaporación se

realiza a través de máscaras metálicas para ahorrar el paso litográfico o Serigrafando una película gruesa de plata ($w25 \mu\text{m}$)

- Metalización del reverso por evaporación.
- El lado frontal tiene un recubrimiento antirreflectante ya sea por técnicas de centrifugado o por evaporación de TaO o TiO y SiOx. Se utiliza un evaporador de haz de electrones.
- En el caso de ensamblaje de celdas solares usando reflujo de soldadura, las áreas de contacto se recubren con soldadura (estaño + plomo + plata).
- Ahora las células solares están listas y normalmente se almacenan en casetes.

2.6.4. Componentes de un sistema fotovoltaico

Debido al tamaño limitado de la celda solar, solo entrega una cantidad limitada de energía bajo condiciones de voltaje de corriente fija que no son prácticas para la mayoría de las aplicaciones. Para utilizar la electricidad solar en dispositivos prácticos que requieren un voltaje y corriente particular para su funcionamiento, se deben conectar varias células solares para formar un módulo solar, también llamado módulo fotovoltaico. Para la generación a gran escala de electricidad solar, los paneles solares se conectan entre sí en una matriz fotovoltaica. Aunque los paneles solares son el corazón de un sistema fotovoltaico, se requieren muchos otros componentes para que un sistema funcione, como discutimos muy brevemente anteriormente. Juntos, estos componentes se denominan el equilibrio del sistema (BOS). Los componentes necesarios dependen de si el sistema está conectado a la red eléctrica o si está diseñado como un sistema independiente (Arno et al., 2016). Los componentes más importantes pertenecientes al BOS son:

- Una estructura de montaje para fijar los módulos y orientarlos hacia el sol.

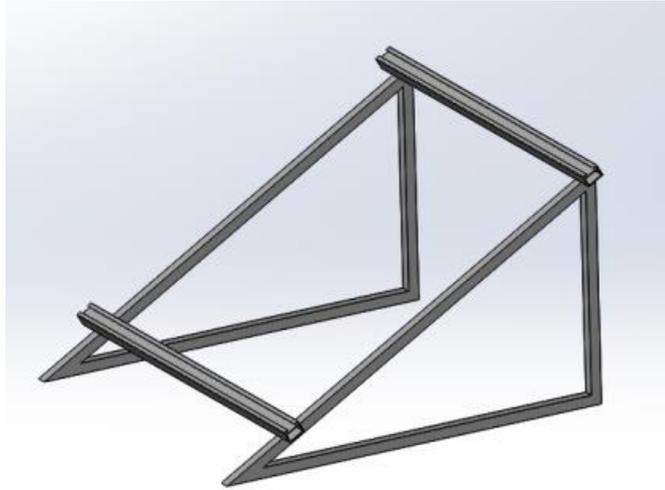


Figura 10: *Estructura de soporte convencional.*
Fuente: Pelayo, 2016.

- El almacenamiento de energía como parte vital de los sistemas autónomos, porque asegura que el sistema pueda entregar electricidad durante la noche y en períodos de mal tiempo. Por lo general, las baterías se utilizan como unidades de almacenamiento de energía.



Figura 11: *Batería para sistemas fotovoltaicos.*
Fuente: Pelayo, 2016.

- Convertidores CC-CC para convertir la salida del módulo, que tendrá un voltaje variable según la hora del día y las condiciones climáticas, en un voltaje de salida compatible que pueda usarse como entrada para un inversor en un sistema conectado a la red.



Figura 12: *Convertidor CC/CC.*

Fuente: Energy, 2022.

- Inversores que se utilizan en sistemas conectados a la red para convertir la electricidad de CC que se origina en los módulos fotovoltaicos en electricidad de CA que se puede alimentar a la red eléctrica. Muchos inversores tienen un convertidor DC-DC incluido para convertir el voltaje variable de la matriz fotovoltaica en un voltaje constante que es la entrada para el convertidor DC-AC real. También los sistemas autónomos pueden tener un inversor que está conectado a las baterías. El diseño de un inversor de este tipo difiere considerablemente del de un sistema conectado a la red.



Figura 13: *Inversor para sistema fotovoltaico.*

Fuente: MPPTSolar, 2021.

- Controladores de carga que se utilizan en sistemas autónomos para controlar la carga y, a menudo, también la descarga de la batería. Evitan que las baterías se sobrecarguen y también que se descarguen a través de la matriz fotovoltaica durante la noche. Los controladores de carga de gama alta también contienen convertidores CC-CC junto con un rastreador de punto de máxima potencia para que el voltaje y la corriente fotovoltaica sean independientes del voltaje y la corriente de la batería.



Figura 14: Controlador de carga solar.

Fuente: AutoSolar, 2021.

- Cables que se utilizan para conectar los diferentes componentes del sistema fotovoltaico entre sí y con la carga eléctrica. Es importante elegir cables de suficiente grosor para minimizar las pérdidas resistivas.

2.7. LA RADIACIÓN SOLAR

Uno debe ser capaz de describir la radiación solar, ya que es la fuente de alimentación de entrada a los generadores fotovoltaicos. El sol emite radiaciones electromagnéticas como un cuerpo negro que tiene una temperatura superficial de unos 6000 K. Esto se debe a la reacción nuclear que tiene lugar en él, donde el sol convierte el hidrógeno en helio. El radio del sol es de $1,39 \times 10^9$ m. La potencia de radiación total

recibida del sol en una unidad de área perpendicular a los rayos solares a la distancia media tierra-sol, denominada unidad astronómica, se denomina constante solar (SC), donde 1 unidad astronómica = 1 AU = $1,496 \times 10^{11}$ m. La intensidad de la radiación solar a otras distancias se expresa en términos de SC con $SC \approx 1.353 \text{ kW/m}^2$ (González et al., 2010). Al igual que la radiación de un cuerpo negro, la radiación del sol cubre un amplio espectro de longitudes de onda, desde el ultravioleta profundo hasta el infrarrojo lejano. La distribución espectral de potencia de la luz solar se muestra en la Fig. 10. El eje vertical representa la irradiancia espectral $I(\lambda)$ mientras que el eje horizontal representa la longitud de onda en μm . La irradiancia espectral disminuye debido a la presencia de aire en la atmósfera. Las moléculas de aire dispersan y absorben la radiación solar. Existen múltiples bandas de absorción para O_2 , H_2O y CO_2 . Es importante notar que la radiación solar se asemeja a la radiación del cuerpo negro a 6000 K representada por la línea discontinua.

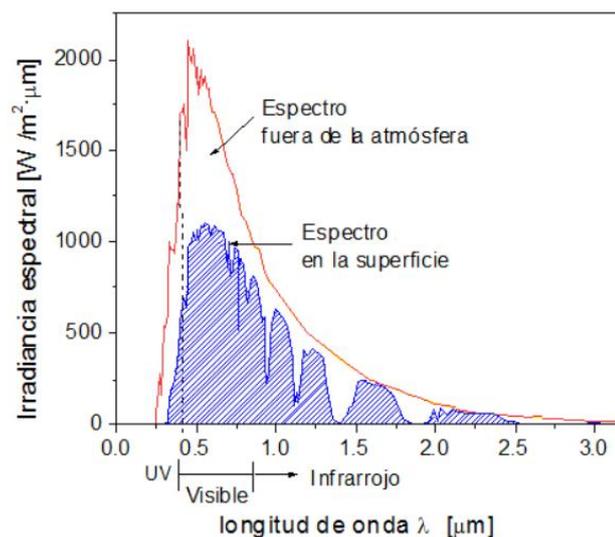


Figura 15: Irradiancia y longitud de onda.

Fuente: González et al., 2010.



2.8. LENGUAJE DE PROGRAMACIÓN PYTHON

Desde 1991, el lenguaje de programación Python se consideró un relleno de lagunas, una forma de escribir scripts que " automatizan las cosas aburridas " (como lo expresó un libro popular sobre el aprendizaje de Python) o crear rápidamente prototipos de aplicaciones que se implementarán en otros lenguajes. Sin embargo, en los últimos años, Python se ha convertido en un ciudadano de primera clase en el desarrollo de software moderno, la gestión de infraestructura y el análisis de datos. Ya no es un lenguaje de utilidad de trastienda, sino una fuerza importante en la creación de aplicaciones web y la gestión de sistemas, y un impulsor clave de la explosión en el análisis de big data y la inteligencia artificial. La cantidad de funciones en el lenguaje en sí es modesta y requiere una inversión de tiempo o esfuerzo relativamente pequeña para producir sus primeros programas. La sintaxis de Python está diseñada para ser legible y sencilla. Esta simplicidad hace de Python un lenguaje de enseñanza ideal y permite que los recién llegados lo aprendan rápidamente. Como resultado, los desarrolladores dedican más tiempo a pensar en el problema que intentan resolver y menos tiempo a pensar en las complejidades del lenguaje o en descifrar el código dejado por otros. A lo largo de los años, he escuchado a muchas personas decir que Python es un lenguaje fácil de aprender y que Python también es un lenguaje simple. Hasta cierto punto, ambas afirmaciones son verdaderas; pero sólo hasta cierto punto (Hunt, 2019). Si bien el núcleo del lenguaje Python es fácil de aprender y relativamente simple (en parte gracias a su consistencia); la gran riqueza de las construcciones del lenguaje y la flexibilidad disponible pueden ser abrumadoras. Además, el entorno de Python, su ecosistema, la gama de bibliotecas disponibles, las opciones disponibles que suelen competir, etc. Python es versátil para distintas áreas de aplicaciones en la tecnología y son los siguientes:



- Gráficos de computadora: El libro cubre gráficos por computadora y arte generado por computadora en Python, así como interfaces gráficas de usuario y gráficos a través de Matplotlib.
- Programación de juegos: Este tema se cubre usando la biblioteca pygame.
- Entrada/salida de archivos: El libro cubre la lectura y escritura de archivos de texto, así como la lectura y escritura de archivos CSV y Excel. Aunque no están estrictamente relacionadas con la entrada de archivos, las expresiones reguladoras se incluyen en esta sección, ya que pueden usarse para procesar datos textuales contenidos en archivos.
- Acceso a la base de datos: El libro presenta bases de datos y bases de datos relacionales en particular. Luego presenta el estándar de acceso a la base de datos Python DB-API y una implementación de este estándar, el módulo PyMySQL utilizado para acceder a una base de datos MySQL.
- Inteligencia Artificial: Esta área es muy usada para este lenguaje de programación, tenemos librerías como Pytorch, Tensorflow, FastAI, entre otras que hacen de este lenguaje un recurso óptimo y líder en el ámbito de inteligencia artificial.

2.8.1. Versiones de Python

Actualmente hay dos versiones principales de Python llamadas Python 2 y Python 3 y son las siguientes:

- Python 2 se lanzó en octubre de 2000 y ha sido, y sigue siendo, muy utilizado.
- Python 3 se lanzó en diciembre de 2008 y es una revisión importante del lenguaje que no es compatible con versiones anteriores.

Los problemas entre las dos versiones se pueden resaltar mediante la impresión simple.



Instalaciones:

- En Python 2 esto se escribe como `print 'Hello World'`.
- En Python 3 esto se escribe como impresión ('Hello World').

Puede que no parezca una gran diferencia, pero la inclusión de '()' marca un cambio importante y significa que cualquier código escrito para una versión de Python probablemente no se ejecutará en la otra versión. Hay herramientas disponibles, como la utilidad 2 a 3, que (parcialmente) automatizarán la traducción de Python 2 a Python 3 pero, en general, aún le queda mucho trabajo por hacer. Esto entonces plantea la pregunta de qué versión usar. Aunque el interés en Python 3 aumenta constantemente, hay muchas organizaciones que todavía usan Python 2. Elegir qué versión usar es una preocupación constante para muchas empresas. Sin embargo, el plan de finalización de la vida útil de Python 2 se anunció inicialmente en 2015 y, aunque se pospuso hasta 2020 debido a la preocupación de que una gran cantidad de código existente no pudiera transferirse fácilmente a Python 3, todavía vive en préstamo. tiempo. Python 3 es el futuro del lenguaje Python y es esta versión la que ha introducido muchas de las características nuevas y mejoradas del lenguaje y de la biblioteca (que ciertamente se han vuelto a trasladar a Python 2 en muchos casos). Esta investigación se centra únicamente en Python versión 3 (python, 2022).

2.8.2. Recursos útiles de Python

Hay una amplia gama de recursos en la web para Python; Destacaremos algunos aquí que debe marcar (White, 2016). No seguiremos haciendo referencia a estos para evitar repeticiones, pero puede volver a consultar esta sección siempre que lo necesite:

- https://en.wikipedia.org/wiki/Python_Software_Foundation Software Python.



- <https://docs.python.org/3/> El sitio principal de documentación de Python 3. Contiene tutoriales, referencias de bibliotecas, guías de configuración e instalación, así como procedimientos de Python.
- <https://docs.python.org/3/library/index.html> Una lista de todas las funciones integradas para el lenguaje Python: aquí es donde puede encontrar documentación en línea para las diversas clases y funciones que usaremos a lo largo este libro.
- <https://pymotw.com/3/> el módulo Python 3 del sitio de la semana. Este sitio contiene muchos, muchos módulos de Python con breves ejemplos y explicaciones de lo que hacen los módulos. Un módulo de Python es una biblioteca de funciones que se basan en el lenguaje central de Python y lo amplían. Por ejemplo, si está interesado en crear juegos con Python, pygame es un módulo diseñado específicamente para hacerlo más fácil.
- <https://www.fullstackpython.com/email.html> es un boletín mensual que se enfoca en un solo tema de Python cada mes, como una nueva biblioteca o módulo.
- <http://www.pythonweekly.com/> es un resumen semanal gratuito de los últimos artículos, proyectos, videos y próximos eventos de Python.

2.9. MACHINE LEARNING

El aprendizaje automático puede aparecer de muchas formas. Ahora analizamos una serie de aplicaciones, los tipos de datos que manejan y, por último, formalizamos los problemas de una manera algo más estilizada. Esto último es clave si queremos evitar reinventar la rueda para cada nueva aplicación. En cambio, gran parte del arte del aprendizaje automático consiste en reducir una gama de problemas bastante dispares a un conjunto de prototipos bastante limitados. Gran parte de la ciencia del aprendizaje automático consiste en resolver esos problemas y proporcionar buenas garantías para las soluciones (Smola & Vishwanathan, 2008). Comencemos por considerar un par de



ejemplos del aprendizaje natural de los animales. Algunos de los problemas más fundamentales en ML ya surgen en ese contexto, con el que todos estamos familiarizados. Timidez ante el cebo: las ratas aprenden a evitar los cebos venenosos: cuando las ratas encuentran alimentos con un aspecto u olor novedosos, primero comen cantidades muy pequeñas y la alimentación posterior dependerá del sabor del alimento y su efecto fisiológico. Si el alimento produce un efecto nocivo, el nuevo alimento a menudo se asociará con la enfermedad y, en consecuencia, las ratas no lo comerán (Nilsson, 2005). Claramente, hay un mecanismo de aprendizaje en juego aquí: el animal usó la experiencia pasada con algún alimento para adquirir experiencia en la detección de la seguridad de este alimento. Si la experiencia pasada con la comida fue etiquetada negativamente, el animal predice que también tendrá un efecto negativo cuando la encuentre en el futuro (Shalev & Ben, 2014).

2.9.1. Aplicaciones

La mayoría de los lectores estarán familiarizados con el concepto de clasificación de páginas web. Es decir, el proceso de enviar una consulta a un motor de búsqueda, que luego encuentra páginas web relevantes para la consulta y las devuelve en su orden de relevancia. Es decir, el motor de búsqueda devuelve una lista ordenada de páginas web dada una consulta. Para lograr este objetivo, un motor de búsqueda necesita "saber" qué páginas son relevantes y qué páginas coinciden con la consulta. Dicho conocimiento se puede obtener de varias fuentes: la estructura de enlaces de las páginas web, su contenido, la frecuencia con la que los usuarios seguirán los enlaces sugeridos en una consulta o de ejemplos de consultas en combinación con páginas web clasificadas manualmente. Cada vez más se utiliza el aprendizaje automático en lugar de las conjeturas y la ingeniería inteligente para automatizar el proceso de diseño de un buen motor de búsqueda. Una aplicación bastante relacionada es el filtrado colaborativo. Las librerías de Internet, como

Amazon, o los sitios de alquiler de videos, como Netflix, utilizan esta información ampliamente para atraer a los usuarios a comprar bienes adicionales (o alquilar más películas). El problema es bastante similar al de la clasificación de páginas web. Como antes, queremos obtener una lista ordenada (en este caso de artículos). La diferencia clave es que falta una consulta explícita y, en cambio, solo podemos usar las decisiones de compra y visualización anteriores del usuario para predecir los hábitos de visualización y compra futuros (Deisenroth et al., 2022). La información lateral clave aquí son las decisiones tomadas por usuarios similares, de ahí la naturaleza colaborativa del proceso. Es claramente deseable tener un sistema automático para resolver este problema, evitando así conjeturas y tiempo.

2.9.2. Data Preprocessing

Para aplicar el aprendizaje profundo en la naturaleza, debemos extraer datos desordenados almacenados en formatos arbitrarios y procesarlos previamente para satisfacer nuestras necesidades. Afortunadamente, la biblioteca “pandas” puede hacer gran parte del trabajo pesado (Aston et al., 2022).

2.9.3. Aprendizaje supervisado

El aprendizaje supervisado describe tareas en las que se nos proporciona un conjunto de datos que contiene características y etiquetas y se nos asigna la tarea de producir un modelo para predecir las etiquetas dadas las características de entrada. Cada par de etiquetas de características se denomina ejemplo. A veces, cuando el contexto es claro, podemos usar el término ejemplos para referirnos a una colección de entradas, incluso cuando se desconocen las etiquetas correspondientes. La supervisión entra en juego porque para elegir los parámetros, nosotros (los supervisores) proporcionamos al modelo un conjunto de datos que consta de ejemplos etiquetados. En términos

probabilísticos, normalmente nos interesa estimar la probabilidad condicional de una etiqueta dadas las características de entrada. Si bien es solo uno entre varios paradigmas dentro del aprendizaje automático, el aprendizaje supervisado representa la mayoría de las aplicaciones exitosas del aprendizaje automático en la industria (Morde, 2019).

2.9.4. Aprendizaje no supervisado

En el otro escenario, denominado aprendizaje no supervisado, simplemente tenemos un conjunto de vectores de entrenamiento sin valores de función para ellos. El problema en este caso, por lo general, es dividir el conjunto de entrenamiento en subconjuntos, de alguna manera apropiada. (Todavía podemos considerar el problema como uno de aprendizaje de una función; el valor de la función es el nombre del subconjunto al que pertenece un vector de entrada). Los métodos de aprendizaje no supervisados tienen aplicación en problemas taxonómicos en los que se desea inventar formas de clasificar los datos en categorías significativas. También describiremos métodos que son intermedios entre el aprendizaje supervisado y no supervisado (Burkov, 2019).

2.9.5. Aprendizaje Semisupervisado

El aprendizaje semisupervisado es un paradigma de aprendizaje relacionado con el estudio de cómo las computadoras y los sistemas naturales, como los humanos, aprenden en presencia de datos etiquetados y no etiquetados. Tradicionalmente, el aprendizaje se ha estudiado en el paradigma no supervisado (por ejemplo, el agrupamiento, detección de valores atípicos) donde todos los datos no están etiquetados, o en el paradigma supervisado (por ejemplo, la clasificación, regresión) donde todos los datos están etiquetados. El objetivo del aprendizaje semisupervisado es comprender cómo la combinación de datos etiquetados y no etiquetados puede cambiar el comportamiento



de aprendizaje y diseñar algoritmos que aprovechen dicha combinación. El aprendizaje semisupervisado es de gran interés en el aprendizaje automático y la minería de datos porque puede utilizar datos no etiquetados fácilmente disponibles para mejorar las tareas de aprendizaje supervisado cuando los datos etiquetados son escasos o costosos (Zhu y Goldberg, 2009).

2.9.6. Aprendizaje Reforzado

En el aprendizaje automático, el aprendizaje supervisado a veces se contrasta con el aprendizaje no supervisado. Esta es una distinción útil, pero hay algunos dominios problemáticos que tienen características compartidas con cada uno sin encajar exactamente en ninguna categoría. En los casos en que el algoritmo no tiene etiquetas explícitas, pero recibe una forma de retroalimentación, estamos tratando con un tercer y distinto paradigma de aprendizaje automático: el aprendizaje por refuerzo (Burkov, 2019).

2.10. MÉTODOS DE ENSAMBLE EN MACHINE LEARNING

Los métodos de ensamble son una técnica de aprendizaje automático que combina varios modelos básicos para producir un modelo predictivo óptimo. Para comprender mejor esta definición, retrocedamos un paso hacia el objetivo final del aprendizaje automático y la construcción de modelos. Esto tendrá más sentido a medida que profundice en ejemplos específicos y por qué se usan los métodos Ensemble. Utiliza en gran medida los árboles de decisión para delinear la definición y la practicidad de los métodos de conjunto (sin embargo, es importante tener en cuenta que los métodos de conjunto no solo pertenecen a los árboles de decisión), en la Fig. 16 se muestra el funcionamiento de los algoritmos de tipo ensamble (Sen, 2021).

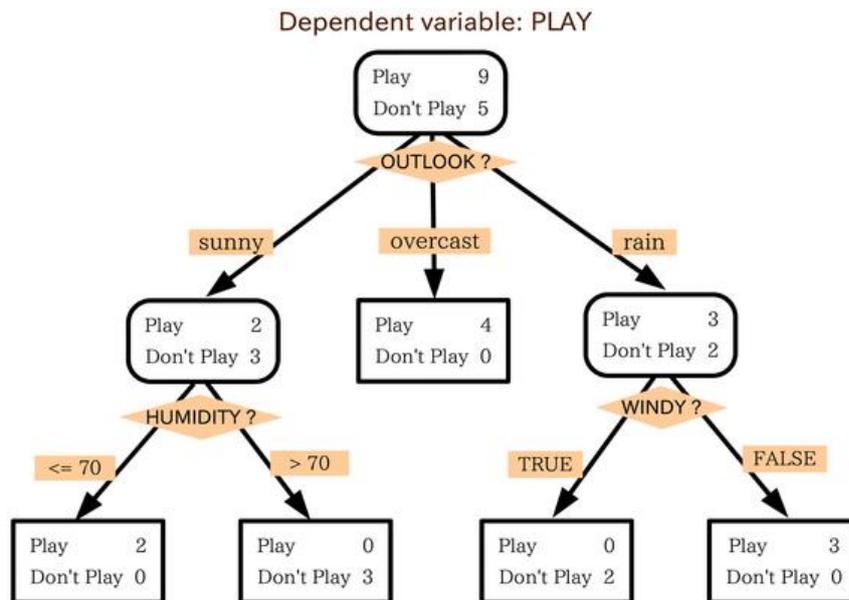


Figura 16: Algoritmos de Ensamble.

Fuente: Lutins, 2017.

Un árbol de decisión determina el valor predictivo basado en una serie de preguntas y condiciones. Por ejemplo, este simple árbol de decisiones que determina si un individuo debe jugar afuera o no. El árbol tiene en cuenta varios factores climáticos y, dado cada factor, toma una decisión o hace otra pregunta. En este ejemplo, cada vez que esté nublado, jugaremos afuera. Sin embargo, si está lloviendo, debemos preguntarnos si hace viento o no. Si hay viento, no jugaremos. Pero dado que no hay viento, áptate bien los cordones de los zapatos porque íbamos a salir a jugar (Morde, 2019). Aquí es donde los métodos de conjunto son útiles. En lugar de depender simplemente de un árbol de decisión y esperar que hayamos tomado la decisión correcta en cada división, los métodos de conjunto nos permiten tener en cuenta una muestra de los árboles de decisión, calcular qué características usar o qué preguntas hacer en cada división y hacer una evaluación final. predictor basado en los resultados agregados de los árboles de decisión muestreados.

2.10.1. Bagging

El aprendizaje de ensamble del método de embolsado representa el arranque y la agregación de la combinación de dos modelos en un modelo de conjunto. El embolsado utiliza el muestreo aleatorio para reducir la varianza del modelo mediante la creación de más datos en la etapa de entrenamiento del modelo y cada elemento del embolsado tiene una forma de aparecer en el nuevo conjunto de datos. Este método disminuye la varianza y cambia por poco la predicción a un resultado esperado (Sen, 2021). Bagging recibe su nombre porque combina Bootstrapping y Aggregation para formar un modelo de conjunto. Dada una muestra de datos, se extraen varias submuestras de arranque. Se forma un árbol de decisión en cada una de las submuestras arrancadas. Después de que se haya formado cada árbol de decisión de submuestra, se usa un algoritmo para agregar los árboles de decisión para formar el predictor más eficiente, en la Fig. 17 se muestra el funcionamiento interno del algoritmo Bagging.

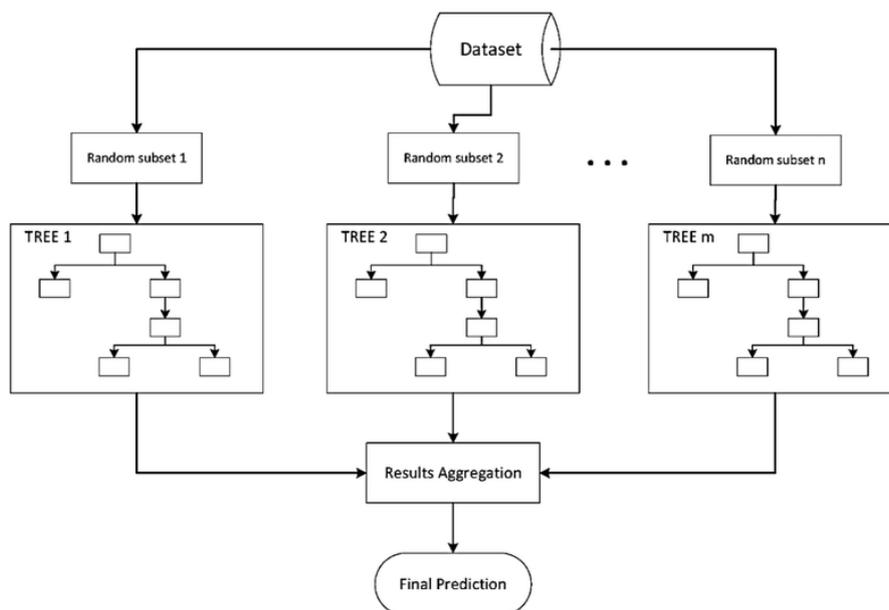


Figura 17: Algoritmo Bagging.

Fuente: Lutins, 2017.



2.10.2. Gradient Boosting

Al calcular los residuos, encontramos qué tan bien (o mal) predice el objetivo de cada ejemplo de entrenamiento el modelo actual f . Luego entrenamos otro árbol para corregir los errores del modelo actual (es por eso que usamos residuos en lugar de etiquetas reales) y agregamos este nuevo árbol al modelo existente con algo de peso. Por lo tanto, cada árbol adicional agregado al modelo corrige parcialmente los errores cometidos por los árboles anteriores hasta que se combina el número máximo de árboles. Para ver la similitud entre el aumento de gradiente y el descenso de gradiente, recuerde por qué calculamos el gradiente en regresión lineal: lo hicimos para tener una idea de dónde debemos mover los valores de nuestros parámetros para que la función de costo de MSE alcance su mínimo (Ferreira y Figueiredo , 2012). El gradiente mostraba la dirección, pero no sabíamos qué tan lejos debíamos ir en esta dirección, así que usamos un pequeño paso en cada iteración y luego volvimos a evaluar nuestra dirección. Lo mismo sucede en el aumento de gradiente. Sin embargo, en lugar de obtener el gradiente directamente, usamos su proxy en forma de residuos: nos muestran cómo se debe ajustar el modelo para que el error (el residuo) se reduzca. Los tres hiperparámetros principales para ajustar el aumento de gradiente son la cantidad de árboles, la tasa de aprendizaje y la profundidad de los árboles: los tres afectan la precisión del modelo (Schapire y Freund, 2012). La profundidad de los árboles también afecta la velocidad de entrenamiento y predicción: cuanto más cortos, más rápido. Se puede demostrar que el entrenamiento en residuos optimiza el modelo general f para el criterio de error cuadrático medio. Puede ver la diferencia con el embolsado aquí: el aumento reduce el sesgo (o el ajuste insuficiente) en lugar de la varianza. Como tal, el impulso puede sobreajustarse. Sin embargo, al ajustar la profundidad y la cantidad de árboles, se puede evitar en gran medida el sobreajuste. El aumento de gradiente es uno de los algoritmos de aprendizaje



automático más potentes. No solo porque crea modelos muy precisos, sino también porque es capaz de manejar grandes conjuntos de datos con millones de ejemplos y funciones. Por lo general, supera al bosque aleatorio en precisión, pero debido a su naturaleza secuencial, puede ser significativamente más lento en el entrenamiento (Burkov, 2019). Los árboles de decisión producen modelos interpretables que son útiles para una variedad de problemas, pero su precisión puede mejorar considerablemente cuando se combinan muchos árboles en un modelo conjunto. Por ejemplo, dada una instancia de entrada para ser clasificada, podemos probarla con muchos árboles creados en diferentes subconjuntos del conjunto de entrenamiento y devolver la moda de todas las predicciones. Esto tiene el efecto de reducir el error del clasificador porque reduce la varianza en la estimación del clasificador. En la Fig. 18 se muestra un conjunto de dos árboles de decisión. Podemos predecir la etiqueta de salida usando todos los árboles tomando la predicción de clase más común o algún promedio ponderado de todas las predicciones. Los métodos de aprendizaje por conjuntos también se pueden utilizar para reducir el componente de sesgo en el error de clasificación del alumno base. Impulsar es un método de conjunto que crea miembros de conjunto secuencialmente. El miembro más nuevo se crea para compensar las instancias etiquetadas incorrectamente por los alumnos anteriores (Mitchell y Eibe, 2017). El aumento de gradiente es una variación del aumento que representa el problema de aprendizaje como un descenso de gradiente en alguna función de pérdida diferenciable arbitraria que mide el rendimiento del modelo en el conjunto de entrenamiento.

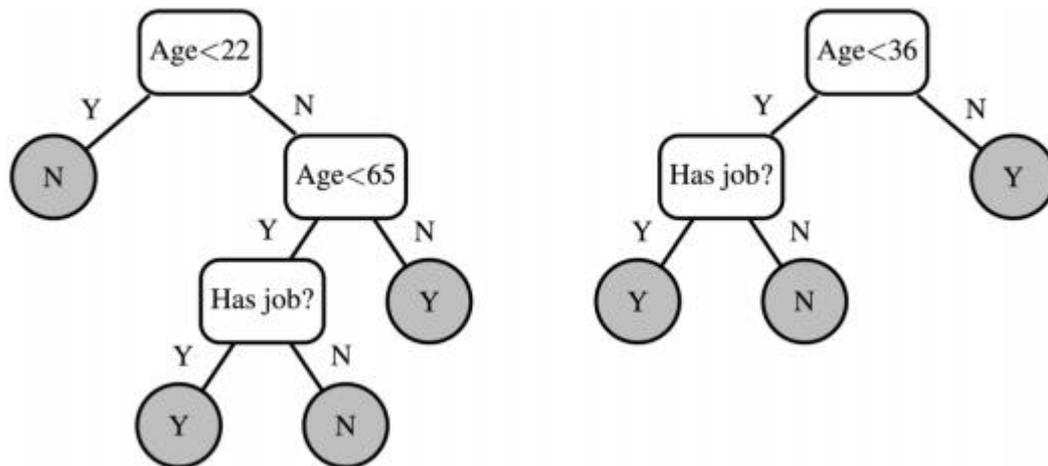


Figura 18: Árboles de decisión de tipo ensemble.

Fuente: Mitchell y Eibe, 2017.

2.10.3. XGBoost

XGBoost es un conjunto de árboles de decisión basado en la potenciación de gradientes diseñado para ser altamente escalable. De manera similar al aumento de gradiente, XGBoost genera una expansión aditiva de la función objetivo al minimizar una función de pérdida. Teniendo en cuenta que XGBoost se centra únicamente en los árboles de decisión como clasificadores base, se utiliza una variación de la función de pérdida para controlar la complejidad de los árboles. Esta estructura de almacenamiento basada en columnas permite encontrar la mejor división para cada atributo considerado en paralelo. Además, en lugar de escanear todas las divisiones candidatas posibles, XGBoost implementa un método basado en percentiles de los datos donde solo se prueba un subconjunto de las divisiones candidatas y su ganancia se calcula utilizando estadísticas agregadas. Esta idea se parece al submuestreo de datos a nivel de nodo que ya está presente en los árboles CART. Además, en XGBoost se usa un algoritmo consciente de la escasez para eliminar de manera efectiva los valores faltantes del cálculo de la ganancia de pérdida de los candidatos divididos (Bentejaca et al., 2021). Los siguientes parámetros

con más importancia o comúnmente ajustados en distintas investigaciones para XGBoost son los siguientes:

- La tasa de aprendizaje (`learning_rate`) o reducción v .
- La reducción de pérdida mínima (`gamma`): cuanto más alto es este valor, menos profundos son los árboles.
- La profundidad máxima del árbol (`max_depth`)
- La fracción de características que se evaluarán en cada división (`colsample_bylevel`).
- La tasa de submuestreo (`subsample`): el muestreo se realiza sin reposición.

XGBoost es un método de árbol de conjuntos que aplica el principio de potenciar a los alumnos débiles (en general, CART) mediante la arquitectura de descenso de gradiente. Sin embargo, XGBoost mejora el marco GBM básico a través de la optimización de sistemas y mejoras algorítmicas.



Figura 19: Algoritmo XGBoost.

Fuente: Morde, 2019.



CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. MUESTRA DE LA INVESTIGACIÓN

Debido a que la población es pequeña no es necesario obtener una muestra, por lo que se trabaja con la población total (Martínez, 2020). Por lo que se trabajó con una cantidad de datos de 12770 muestras por cada variable meteorológica obtenida por los sensores, el sistema de adquisición de datos tuvo una lectura de aproximadamente una muestra por minuto durante la duración de radiación solar. La adquisición de los datos se realizó en total 20 días calendario; los primeros 10 días se instaló el sistema de adquisición de datos en la ciudad de Puno y los 10 siguientes días en la ciudad de Juliaca.

3.2. UBICACIÓN DE LA INVESTIGACIÓN

Los conjuntos de datos de entrenamiento grandes y de alta calidad son necesarios para que los clasificadores de aprendizaje automático logren un alto rendimiento. Debido al alto costo de recopilar datos de calidad, los tamaños de conjuntos de datos para aplicaciones de sistemas fotovoltaicos tienen que contener una variabilidad correcta, es por esa razón que se tuvo dos puntos de adquisición de datos para generalizar mejor los modelos de Machine Learning y así poder garantizar todos los casos posibles con las características de ambos lugares y así el modelo se ajusta mejor a los datos recolectados. Para la investigación se recolectará datos en dos puntos de la ciudad de Juliaca y Puno con el mismo sistema de adquisición, la frecuencia de muestreo también será idéntica.

3.3. DISEÑO DE LA INVESTIGACIÓN

En el presente estudio se realiza una investigación de tipo experimental utilizando la lógica y principios de la ciencia que está plasmada en el prototipo del sistema de



adquisición de datos como la predicción del rendimiento mediante herramientas de Machine Learning con modelos de ensamble; este modelo será alimentado mediante una Data de gran tamaño las cuales la variables de entrada serán los factores meteorológicos que son monitoreados por el sistema físico instalados en la ciudad de Puno y Juliaca (Sampieri et al., 2014).

3.4. TIPO DE INVESTIGACIÓN

Para la investigación presentada en este documento se aplicó una investigación de tipo experimental ya que se busca responder a las necesidades de la problemática haciendo una hipótesis predictiva bajo un diseño experimental el cual contiene la implementación en entornos controlados o condiciones de laboratorios controlados para probar experimentalmente la hipótesis planteamiento del sistema propuesto (Sampieri et al., 2014).

3.5. SISTEMA DE ADQUISICIÓN DE DATOS

También llamados DAQ, un sistema de adquisición de datos comprende un sistema de medición y una computadora que puede medir propiedades eléctricas o físicas y registrarlas para su posterior análisis. El concepto de DAQ se introdujo en 1963, donde giraba principalmente en torno a monitorear o controlar una entidad física con software. IBM fue el primero en anunciar una computadora que se hizo únicamente para tareas de adquisición de datos llamada sistema de adquisición de datos IBM 7700. El IBM 7700 fue reemplazado por un sistema de control y adquisición de datos IBM 1800 más potente en 1964. Con el progreso de la tecnología, las capacidades informáticas han cambiado drásticamente en la actualidad, lo que nos permite procesar y almacenar datos rápidamente de muchas maneras. Los sistemas DAQ son un excelente ejemplo de esto. Estos dispositivos son capaces de capturar datos de un sistema real y almacenar esos datos

en un formato fácilmente recuperable para una revisión científica o de ingeniería adicional. Teniendo en cuenta la intuición de los sistemas DAQ, estos sistemas de medición basados en computadora son de vital importancia en una variedad de verticales que necesitan precisión, como la construcción, la electrónica, la producción y la fabricación, por nombrar algunos. Los sistemas DAQ son portátiles o se pueden operar de forma remota. Los sistemas DAQ portátiles son útiles cuando debe tomar lecturas de una muestra con la que puede interactuar físicamente. Las medidas DAQ remotas se toman donde la presencia humana es incompatible. Componentes básicos de un sistema DAQ Las máquinas IBM de 1960 eran computadoras gigantes de 6 pies de alto. Afortunadamente, los sistemas DAQ modernos son compactos, pero muchas veces más potentes que sus predecesores. Años de avances tecnológicos en electrónica han hecho posible tener máquinas tan elegantes que no comprometen la precisión de la medición.

Teniendo estos conceptos establecidos se define que el sistema de adquisición de datos para esta investigación será regido por el esquema mostrado en la Fig. 20.

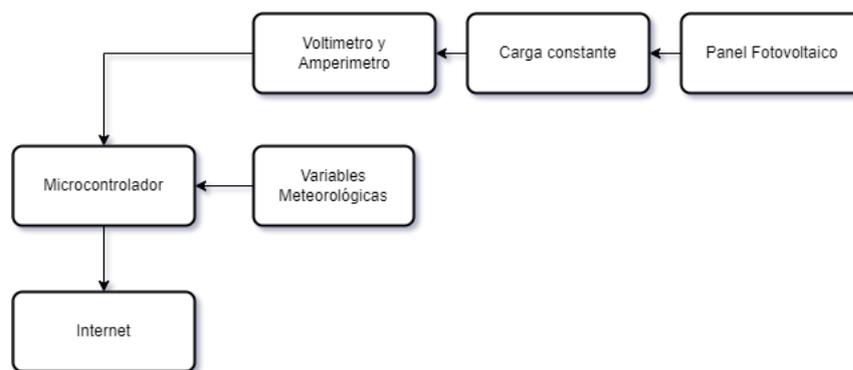


Figura 20: *Esquema de adquisición de datos.*

Elaboración propia.

El módulo de adquisición de datos contiene una caja galvanizada con una tapa de vidrio para no interferir las mediciones de la luminosidad y radiación UV, tenemos que

este sistema fue exitosamente eficiente protegiendo todos los componentes internos de la humedad como de posible lluvia y así estropear los dispositivos.

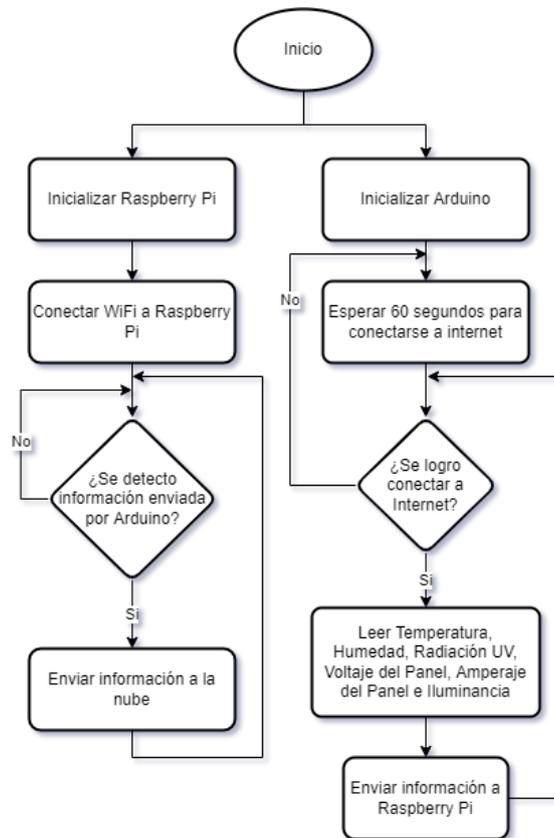


Figura 21: Diagrama de Flujo del sistema de adquisición de datos.

Elaboración propia.

En la Fig. 21 se muestra el diagrama de flujo para ambos dispositivos embebidos que tiene el sistema de adquisición de datos, de manera simultánea ambos inician al conectar el sistema a la red eléctrica de manera que la forma de desconectar dicho sistema es quitando la corriente eléctrica al sistema completo ya que dentro de sus rutinas tenemos que llegan a un punto donde tendrán un bucle para poder recolectar la información y posteriormente enviarlo, ambos sistemas no están comprometidos el uno con el otro ya que la Raspberry Pi contiene una rutina independiente a la rutina del Arduino así poder evitar errores de sincronización o que alguna falla afecte a ambos sistemas.

También la carga se estableció fuera de la caja donde se encuentra los dispositivos para salvaguardar a los mismos ya que la carga por su comportamiento físico tiende a calentar bastante y necesita de una buena ventilación así que se tiene la carga al aire libre pero también protegido de la lluvia y la humedad. Todo el sistema en conjunto se puede ver en la Fig. 22 a continuación.



Figura 22: *Sistema de adquisición de datos.*

Elaboración propia.

Se muestra en la imagen que el panel está ubicado junto al sistema de adquisición de datos esto para facilitar el cableado y monitorización de las variables ya que la estimación de la predicción se realizara con respecto al panel y es necesario tener las mediciones de las variables meteorológicas en ambiente equitativo al del panel para tener un modelo mejor entrenado.

3.5.1. Panel Solar

Los paneles solares monocristalinos están hechos de células solares monocristalinas. Cada celda es una rebanada de un solo cristal de silicio que se cultiva expresamente con el fin de crear paneles solares. En el laboratorio, el cristal crece en



forma de tronco cilíndrico llamado lingote y luego se corta en discos delgados. Cada disco se corta a lo largo de los bordes para formar octágonos. Cuando las celdas solares se colocan en el panel solar, las formas octogonales ayudan a los paneles solares a colocar una cantidad máxima de celdas solares en la matriz. Es muy parecido a las galletas en una bandeja para hornear. Incluso las galletas redondas muy espaciadas siempre tienen espacios entre ellas, pero las galletas octogonales se pueden colocar muy juntas con poco espacio desperdiciado. Distintivos por su color negro, los paneles solares monocristalinos suelen tener un rango de eficiencia de entre el 15 % y el 20 %, y algunos modelos experimentales más nuevos incluso alcanzan cerca del 50 % (Wallender y Allen, 2022). El panel utilizado en esta investigación es del tipo monocristalino con las siguientes características:

- Polaridad Negativa.
- Polaridad Positiva.
- Potencia nominal máxima de 100W.
- Tolerancia de potencia de 3%.
- Voltaje de circuito abierto con 22.32V.
- Voltaje de potencia máxima de 18.6V
- Corriente en cortocircuito de 5.81A.
- Corriente de potencia máxima de 5.38A.
- Voltaje máximo del sistema con 100W.
- Pego de 7.5Kg.
- Dimensión del panel 1000*670*30mm.

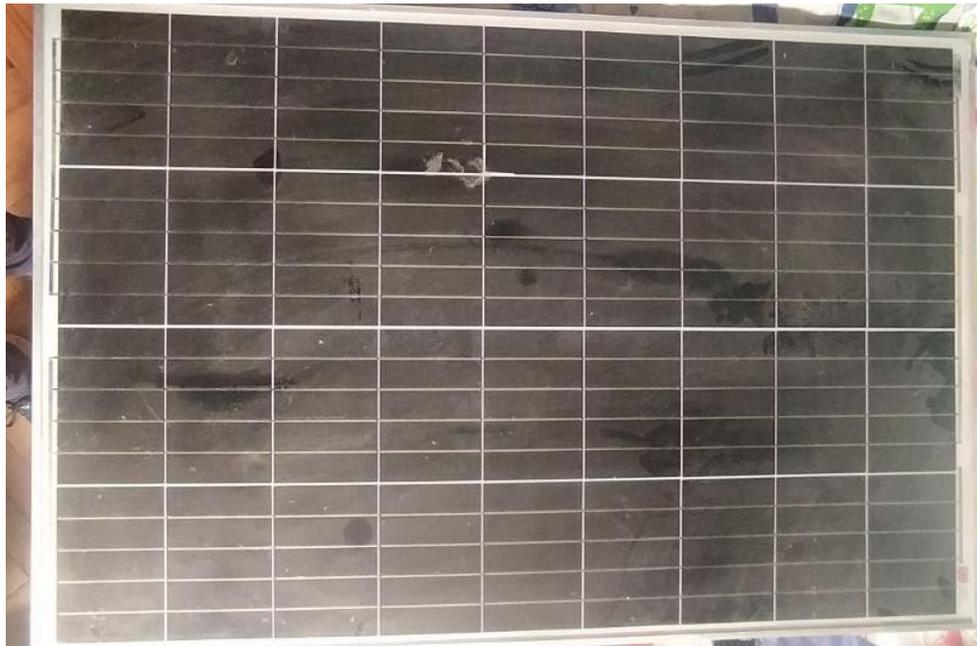


Figura 23: *Panel solar de 100W.*

Elaboración propia.

3.5.2. Microcontrolador – Arduino Mega

Arduino Mega 2560 es una placa de desarrollo de código abierto basada en el microcontrolador Atmega2560 AVR. Este microcontrolador es un microcontrolador de 8 bits. Utiliza tecnología de microchip ATmega16U2. Esta placa se puede programar utilizando el lenguaje de cableado/procesamiento. Este Arduino tiene 54 pines de entrada y de salida digital, de los cuales 14 pines se pueden usar como salidas PWM y 16 pines analógicos, las características de este microcontrolador son las siguientes:

- 4 UART (puertos serie de hardware)
- Un oscilador de cristal de 16 MHz
- Una conexión USB
- Un gato de poder
- Un encabezado ICSP
- Un botón de reinicio

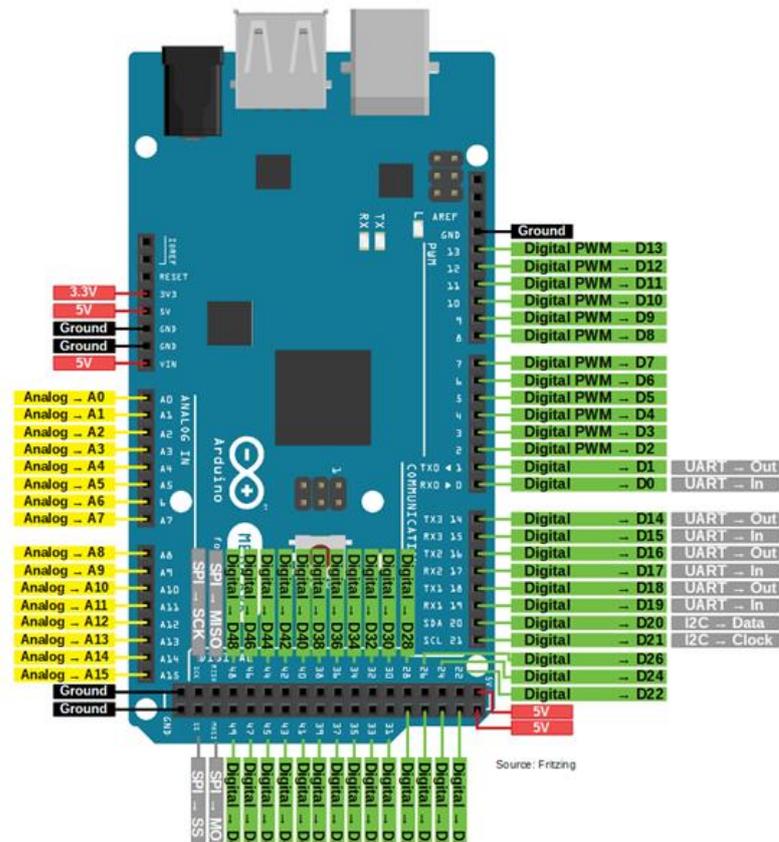


Figura 24: Arduino Mega.

Fuente: *elosciloscopio*, 2019.

En la Fig. 24 se muestra la distribución de los pines soportador por este microcontrolador que está en la placa Arduino de tipo Mega; también será importante considerar esta distribución ya que se utilizará una gran cantidad de estos pines para las entradas de los sensores, alimentación de los sensores y colocación del módulo que habilita de internet al Arduino Mega.

3.5.3. Modulo Ethernet

Este módulo tendrá la capacidad de adquirir y guardar todo el sistema de adquisición de datos hacia internet, es decir, se guardan los valores de voltaje, amperaje, temperatura, etc. hacia el servidor de base de datos que se encuentra en internet. Este sistema o modulo que habilita internet a Arduino Mega para interactuar con la base de

datos se comunica a través del protocolo HTTP o Web gracias a la programación de Arduino.

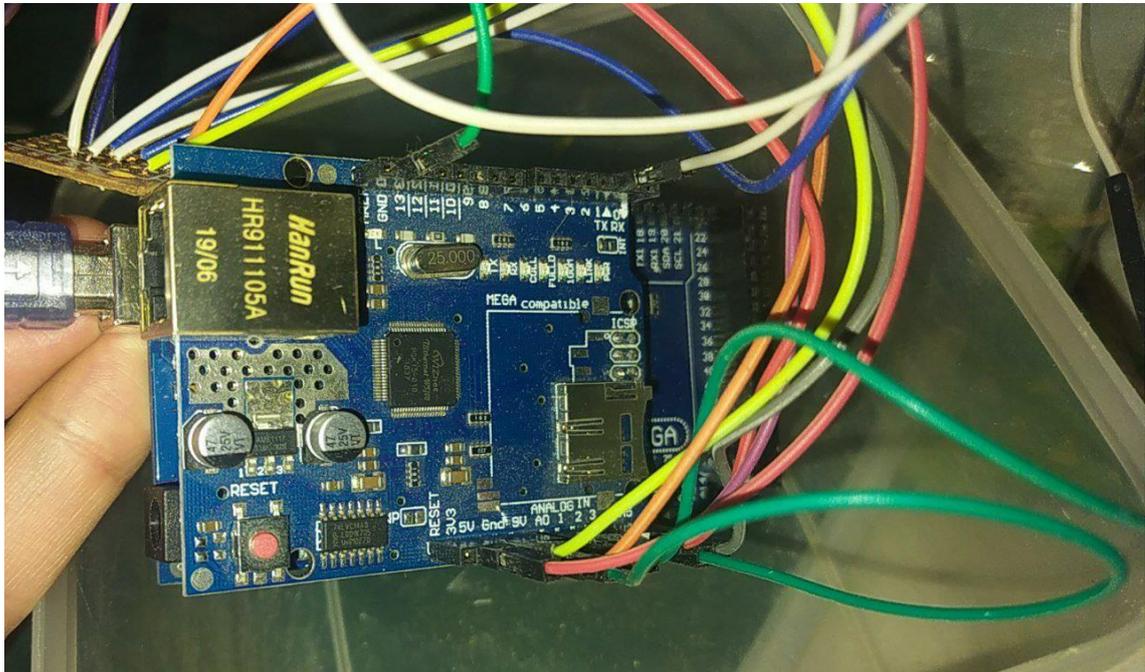


Figura 25: Ethernet Shield W5100.

Elaboración propia.

En la Fig. 25 se muestra este módulo Ethernet Shield modelo W5100 el cual tiene conectado todos los sensores como la alimentación de los mismos. Este módulo se encuentra en la parte superior del Arduino Mega.

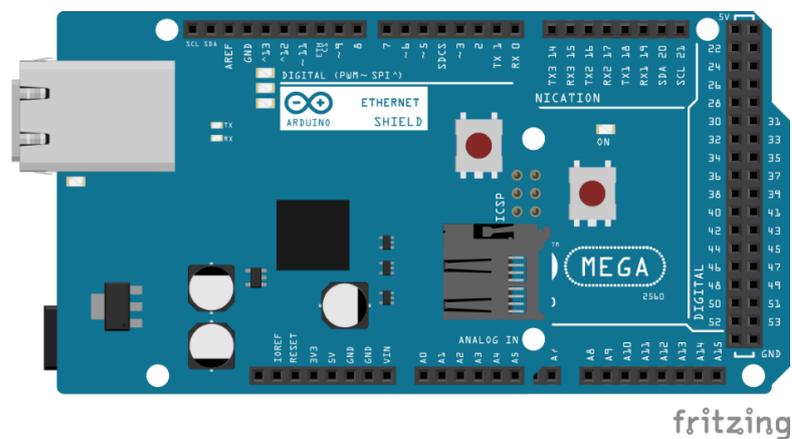


Figura 26: Diagrama de Arduino Mega y módulo Ethernet Shield.

Elaboración propia.

En la Fig. 26 se muestra el diagrama pictórico del módulo completo el cual gobernara el sistema con las lecturas de los sensores y el envío de esta información a la nube.

3.5.4. Carga constante

Para mantener una carga constante y poder medir de forma correcta y eficiente las variables meteorológicas como también la potencia del panel se necesitó de una resistencia de alta potencia para poder sobrellevar la potencia emitida por el panel solar de 100W. La resistencia utilizada es de 8.7 Ohmios, tenemos que el panel tiene una salida de 22 voltios aproximadamente por lo tanto según la ley de Ohm se tiene que dividiremos $22V/8.7\Omega$ dando como resultado 2.53 amperios. Ahora multiplicaremos el voltaje por el amperaje para determinar la potencia de la carga constante que se tendrá, entonces, $22V \times 2.53A$ nos da un resultado igual a 55.66 Watts (Potencia máxima admitida por la resistencia o carga fija es de 5500W).



Figura 27: Carga constante del sistema fotovoltaico.

Elaboración propia.

En la Fig. 27 se muestra la medición respectiva de la resistencia o carga constante que se consideró para el sistema, como muestra la imagen el valor de la carga es de 8.7 Ohmios tomando como referencia los dos puntos extremos de dicha resistencia.

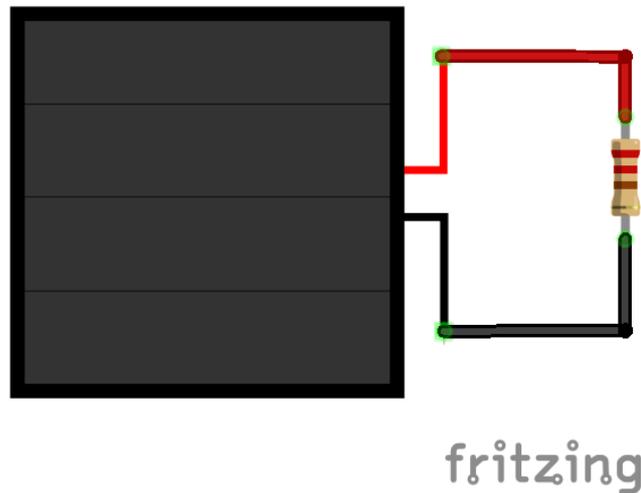


Figura 28: Diagrama de circuito pictórico de la carga y el Panel Solar.

Elaboración propia.

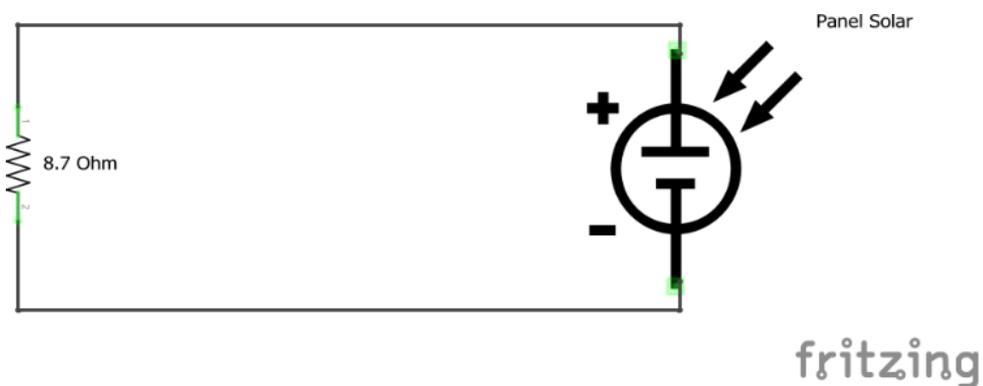


Figura 29: Diagrama circuito esquemático de la carga y el Panel Solar.

Elaboración propia.

El diagrama de la Fig. 28 y Fig. 29 representa la resistencia como el dispositivo resistivo de alta potencia que se utilizó para realizar la investigación. Esta carga funciona como algún dispositivo que estaría constantemente utilizándose sin embargo contiene más precisión ya que la carga es de forma constante.



3.5.5. Sensor de amperaje

Cuando se aplica un campo magnético perpendicular al flujo de corriente, el campo provoca resistencia en la corriente. Esta es la fuerza de Lorentz en acción y se puede observar bien en el efecto Hall. Las corrientes eléctricas se ven afectadas por los campos magnéticos. Cuando se aplica un campo magnético perpendicular al flujo de corriente, el campo provoca resistencia en la corriente. Esta es una manifestación de la fuerza de Lorentz, que empuja los electrones cargados negativamente en la corriente en una dirección dictada por la regla de la mano izquierda. Este movimiento de electrones da como resultado una diferencia de potencial, o voltaje, débil pero medible, perpendicular tanto al flujo de corriente como al campo magnético aplicado. Esto se conoce como efecto Hall, en honor al físico estadounidense Edwin Hall, quien descubrió el fenómeno en 1879. Este efecto es particularmente pronunciado en metales delgados y es fácilmente observable en un plasma de baja densidad (un gas ionizado eléctricamente conductor), como como una luz fluorescente, como en este tutorial. Se observa un efecto similar en los semiconductores, donde el efecto Hall juega un papel importante en el diseño de circuitos integrados en chips semiconductores. En la mayoría de los conductores, como los metales, el efecto Hall es muy pequeño porque la densidad de conducción de los electrones es muy grande y la velocidad de deriva es extremadamente pequeña, incluso para las densidades de corriente más altas que se pueden obtener (Research, 2019). Por lo tanto, el efecto Hall se considera poco importante en la mayoría de los circuitos y dispositivos eléctricos y no se menciona en muchos textos sobre electricidad y magnetismo. Sin embargo, en los semiconductores y en la mayoría de los plasmas de laboratorio, la densidad de corriente es muchos órdenes de magnitud menor que en los metales, y el efecto Hall es correspondientemente mayor y, a menudo, fácilmente observable. Algunos dispositivos para medir campos magnéticos utilizan

semiconductores como elementos sensores y se denominan sondas Hall. El efecto hall es utilizado en el sensor de corriente modelo ACS712T, las características de este sensor son las siguientes:

- Voltaje de operación: 5V.
- Corriente máxima: 5A.
- Contiene señal analógica con ruido bajo.
- Tiempo de respuesta en la salida es de: 5 microsegundos.
- Máximo ancho de banda: 80kHz.
- Contiene un error de: $\pm 1.5\%$
- Resistencia del conductor es de: 1.2M Ω .
- Soporta mediciones de corriente continua como alterna.

El sensor ACS712 es ampliamente utilizado en las industrias, prototipos y otras aplicaciones para adquisición de datos de corriente; las aplicaciones típicas para este tipo de sensor es la medición de la corriente en circuitos controladores de motores, detección y gestión de cargas en circuitos para detectar fallas cuando la corriente sobrepasa los límites establecidos, en esta investigación se utiliza para monitorear la corriente y guardar dichos valores para posteriormente introducirlo en un modelo de Machine Learning. En la Fig. 30 se muestra la conexión del sistema al sensor como también la conexión de los tres pines que están directamente conectados al Arduino para la monitorización de los valores obtenidos por el sensor.

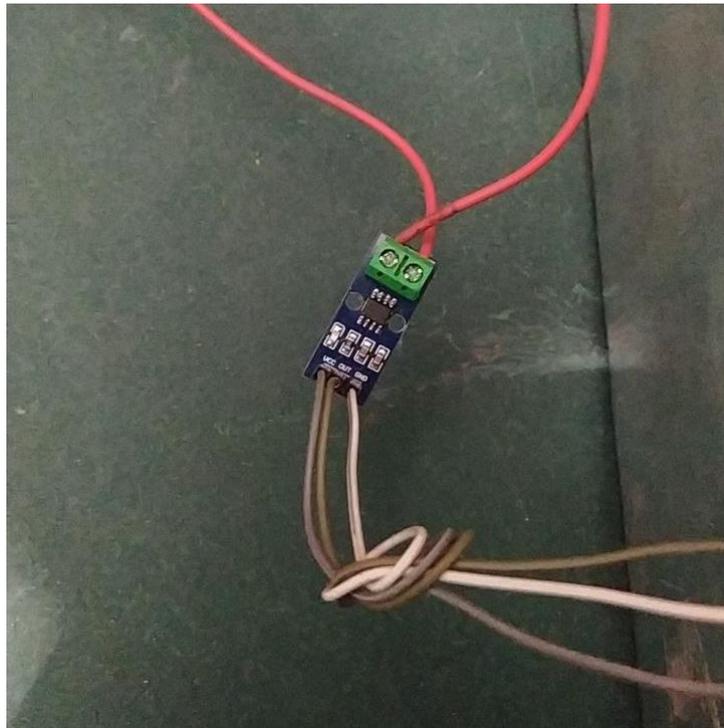
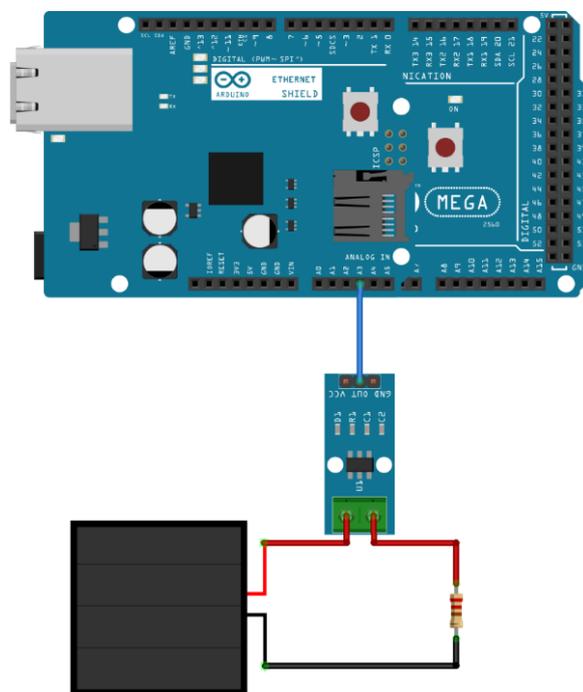


Figura 30: *Sensor de corriente ACS712.*

Elaboración propia.



fritzing

Figura 31: *Diagrama de circuito pictórico del sensor ACS712.*

Elaboración propia.

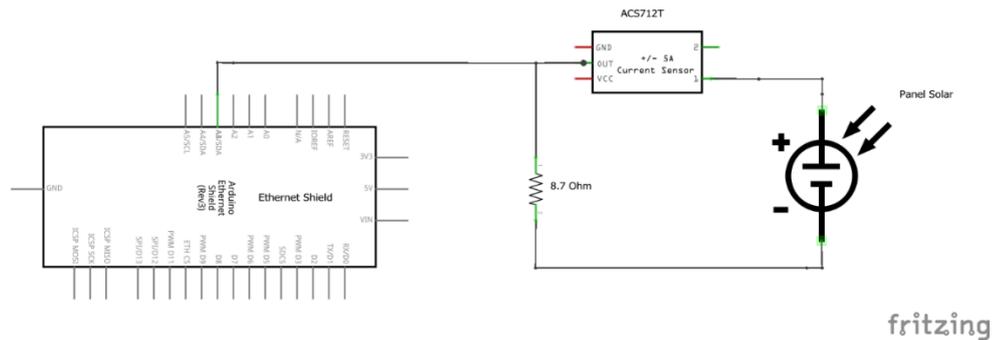


Figura 32: Diagrama circuito esquemático del sensor ACS712.

Elaboración propia.

En la Fig. 31 y la Fig. 32 se muestra las conexiones que se tiene con respecto al sensor de corriente la cual lectura constantemente la corriente que consume la carga constante para que posteriormente pueda enviar esa información al Arduino por medio del módulo Ethernet Shield al pin analógico A3.

3.5.6. Sensor de temperatura y humedad

El sensor utilizado para obtener la temperatura y la humedad es el modelo DHT22 el cual es un sensor digital con un rendimiento optimo y con un costo relativamente reducido a comparación de otros sensores. Dentro del módulo tiene integrado un sensor capacitivo de humedad y un termistor para poder medir el aire que circula por el módulo y luego de recolectar todos estos datos los envía mediante una señal digital. El sensor es soportado por las plataformas de Arduino o Raspberry Pi lo cual lo hace muy simple y sencillo programarlo por medio de software; la distancia máxima recomendada para la alimentación y transmisión de datos del sensor es de veinte metros, las características de dicho sensor son las siguientes:

- Voltaje de alimentación: 5 Voltios.
- Rango de temperatura aptas para medir: -40°C a 80 grados centígrados.
- Precisión de las mediciones: ± 0.5 grados centígrados.
- Resolución de las lecturas: 0.1 grados centígrados.

- Rango de temperatura aptas para medir: 0 a 100% humedad relativa.
- Precisión de las mediciones: 2% humedad relativa.
- Resolución de las lecturas: 0.1% humedad relativa.



Figura 33: *Sensor de temperatura y humedad DHT22.*

Elaboración propia.

En la Fig. 33 se muestra la conexión del sensor de temperatura y humedad DHT22 el cual tiene 3 pines, los extremos sirven para alimentar el sensor con 5 voltios y tierra; el pin del centro sirva para la transmisión de los datos obtenidos por el sensor de manera digital. El sensor se ubica en la parte externa de la caja ya que dentro de la caja no podrá leer de manera correcta y precisa la temperatura y humedad ya que el aire no circula

correctamente, es por ese motivo que el sensor se encuentra fuera de la caja para obtener mejor los factores meteorológicos parecidos al que tiene el panel solar.

Vemos en la Fig. 34 y 35 las conexiones que se tiene con el sensor DHT22 con el Arduino Mega mediante el módulo Ethernet Shield; esta conexión consta de tres pines, dos de ellos se encargan de la alimentación del sensor y el ultimo pin está encargado de enviar la información mediante una señal digital, la información es la humedad y temperatura capturada por el sensor.

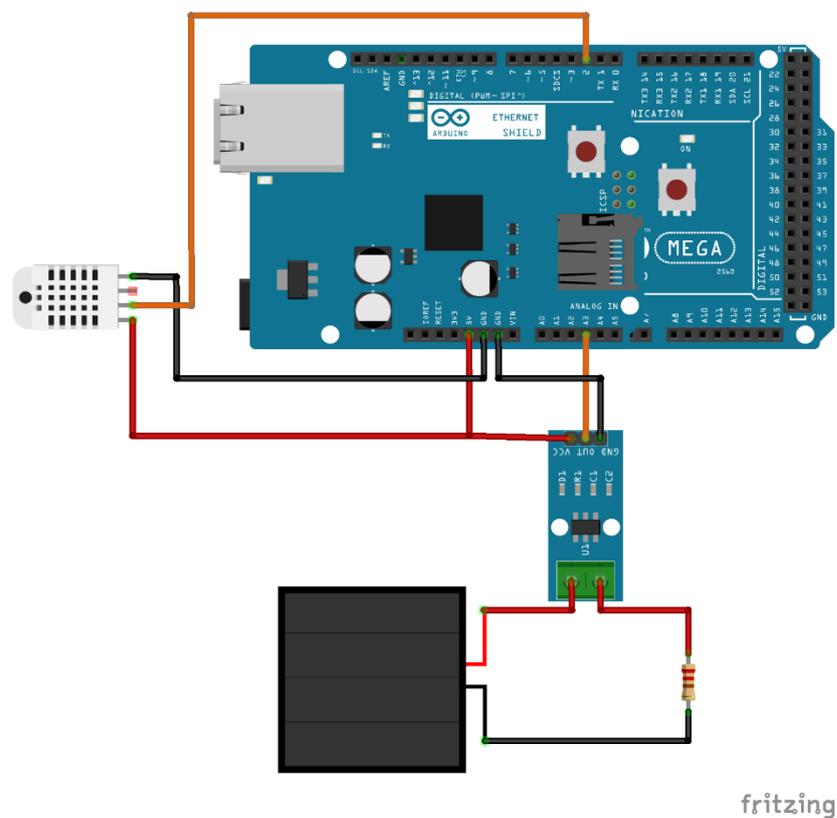


Figura 34: Diagrama de circuito pictórico del sensor DHT22:
Elaboración propia.

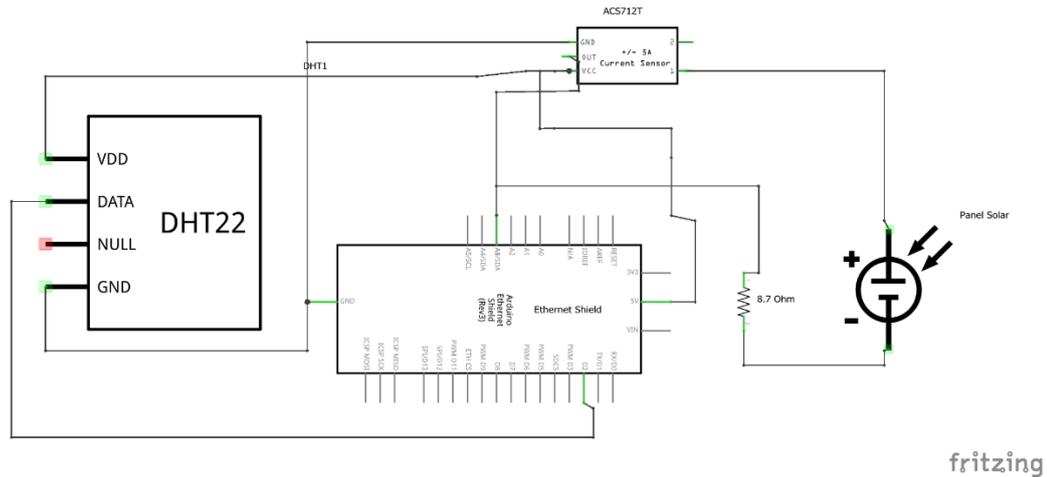


Figura 35: Diagrama de circuito esquemático del sensor DHT22.

Elaboración propia.

3.5.7. Sensor de iluminancia

Para poder calcular o medir la cantidad de luz que hay en un área determinada se utiliza un sensor denominado modulo LDR; este sensor tiene la capacidad de ser un dispositivo cuya internamente contiene una resistencia y varía de acuerdo a la cantidad de luz que recibe. Este módulo LDR es óptimo para el control de iluminación en un ambiente determinado. El módulo contiene alimentación digital (5 voltios); por otro lado, la transmisión de la resistencia puede ser en formato digital como también en analógico, para fines prácticos en esta investigación se utilizará el módulo con el sensor de tipo analógico. Este sensor contiene una alta correlación con la potencia que emitirá el panel fotovoltaico lo cual ayudará de forma positiva al modelo que se implementó para realizar las predicciones.



Figura 36: *Sensor de iluminancia – Módulo LDR.*

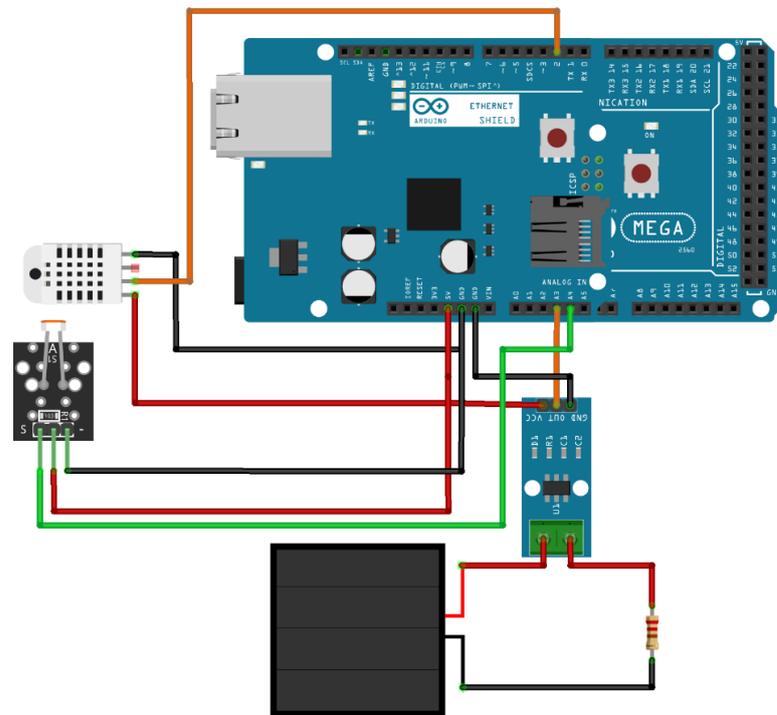
Elaboración propia.

En la Fig. 36 se muestra el módulo LDR sujetado a una estructura blanda para que constantemente este una forma perpendicular al sol para poder medir la iluminancia correctamente; el panel solar también se encuentra en la misma posición garantizando de que las lecturas de los valores son correctas, las características del sensor son las siguientes:

- Voltaje de alimentación: 5V.
- Salida digital y analógica.
- Opamp LM393 en modo comparador.

En la Fig. 37 y Fig. 38 se muestra los diagramas de los sistemas con las conexiones respectivas del sensor LDR o sensor de luminosidad, vemos que se tiene como los sensores anteriores alimentación a la fuente del Arduino por medio del Módulo Ethernet Shield, las conexiones de alimentación son 5V y GND, mientras que el tercer pin que se tiene es el encargado de enviar la variación de luminosidad ya que el sensor funciona como una resistencia variable acorde a la luz que capte; este pin donde se envía esta

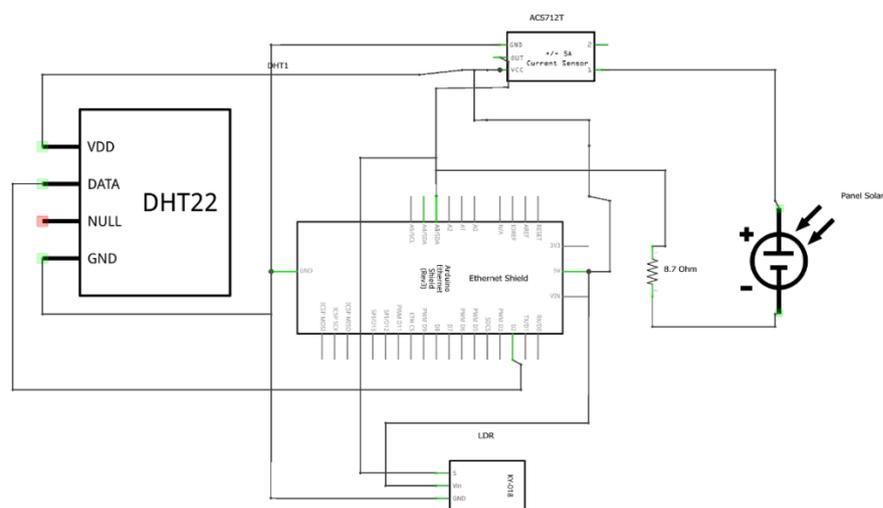
información es de tipo analógico y está conectado al pin número 4 (A4) del módulo Arduino Ethernet Shield.



fritzing

Figura 37: Diagrama de circuito pictórico del sensor LDR.

Elaboración propia.



fritzing

Figura 38: Diagrama de circuito esquemático del sensor LDR

Elaboración propia.

3.5.8. Sensor de radiación UV

Para medir la radiación UV se utilizará el sensor ML8511, este sensor emite una señal de tipo analógica el cual varia depende de la cantidad de luz UV que se detecta en un área determinada; este sensor es ideal para medir índice de radiación UV en condiciones ambientales normales, estaciones meteorológicas, entre otros. La medición de la longitud de onda que se medirá es entre 280 nm a 390nm, este rango cubre la radiación UV de tipo B como también de tipo A. La salida analógica es proporcional a la intensidad de radiación UV (mW/cm^2). Esta señal analógica será conectada al microcontrolador Arduino para poder capturar los valores y ser enviados hacia el servidor.

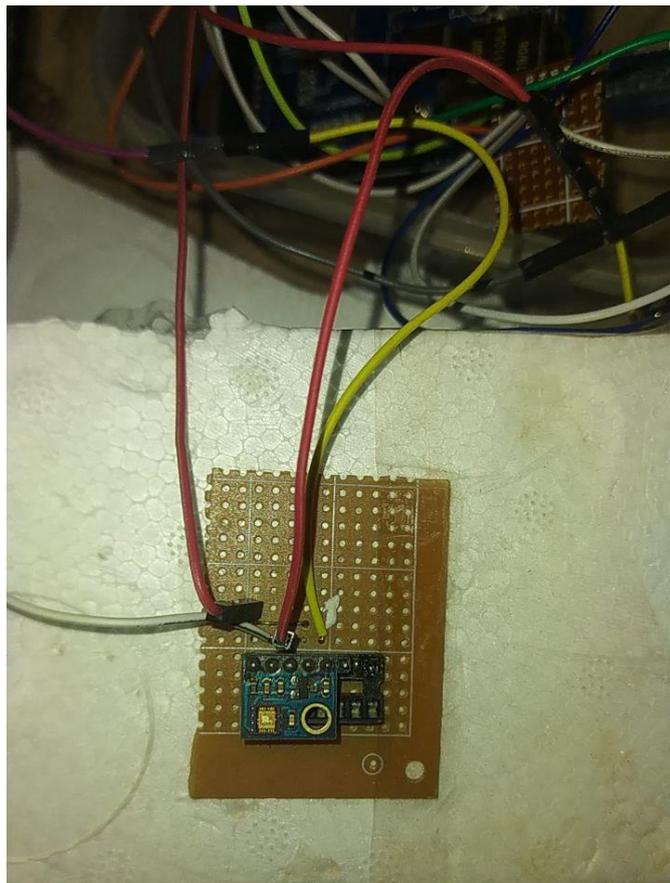


Figura 39: *Sensor de radiación UV - ML8511.*

Elaboración propia.

Como se muestra en la Fig. 39 el sensor de radiación UV está fijado en un material no conductor para evitar lecturas erróneas, vemos que también está soldado en una PCB agujereada para poder fijar de mejor manera el sensor ya que tiene varios pines para que el circuito funcione correctamente.

En la Fig. 40 y Fig. 41 se muestra ya una visión más completa de todo el sistema casi completo ya que aún se tendrá que agregar a la lectura del voltaje de la carga constante, tenemos en ambas imágenes los diagramas de las conexiones que se tiene con respecto a los sensores, se tiene que el sensor de radiación UV - ML8511 contiene cuatro pines, de los cuales dos de estos pines fueron utilizados para la alimentación, es decir 3.3V y GND; posteriormente según el data sheet del fabricante el pin denominado “EN” se conecta a 3.3V como también a la entrada Analógica 1 del Arduino, es decir al pin A1; posteriormente el pin denominado como “OUT” se conecta directamente al pin A0 del Arduino mediante el módulo Ethernet Shield.

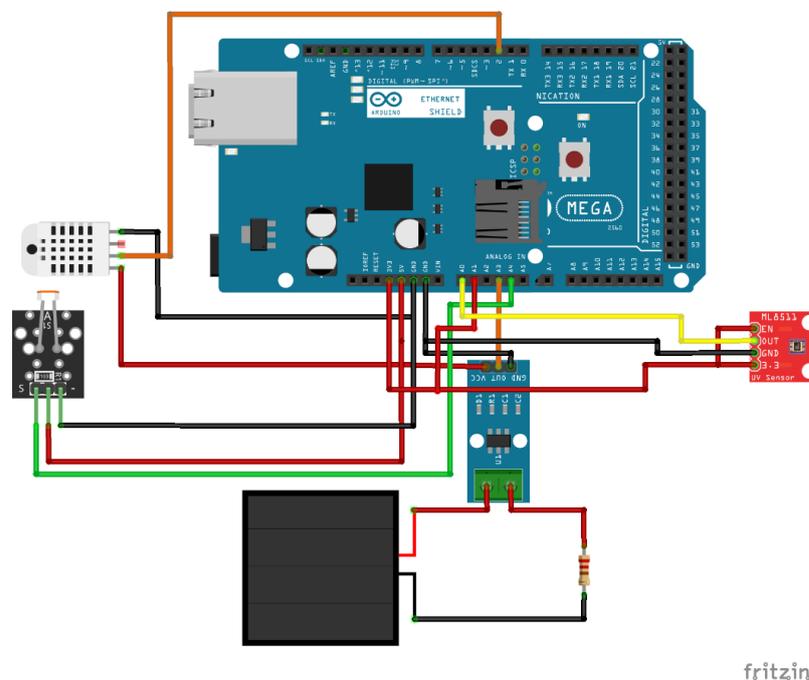


Figura 40: Diagrama de circuito pictórico del sensor UV ML8511.

Elaboración propia.

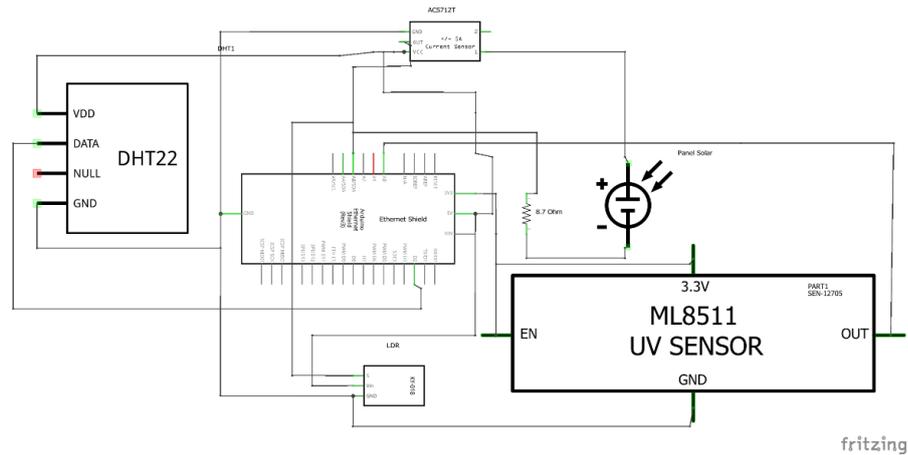


Figura 41: Diagrama de circuito esquemático del sensor UV ML8511.
Elaboración propia.

3.5.9. Sensor de voltaje

Para medir el voltaje del sistema se utilizó un divisor de voltaje, en la Fig. 42 se muestra que tendremos el diagrama mostrado teniendo en cuenta que los valores de la resistencia de R1 es equivalente a 20K, mientras que la resistencia R2 equivale a 5K, teniendo en cuenta que el voltaje del panel es aproximadamente 24 voltios, realizando los cálculos según las fórmulas de la Fig. 41 el voltaje de salida máxima será de 4.8 voltios el cual es admitido por el microcontrolador del Arduino. Esta muestra será monitoreada en un rango de 0 a 1024, siendo 0 un voltaje de mínimo (0 voltios) y 1024 equivale a 4.8 voltios lo cual también indicara que se tiene en el panel el voltaje máximo que es aproximadamente 24 voltios.

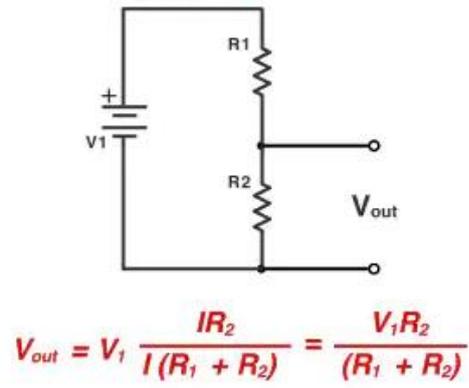


Figura 42: Divisor de voltaje.

Elaboración propia.

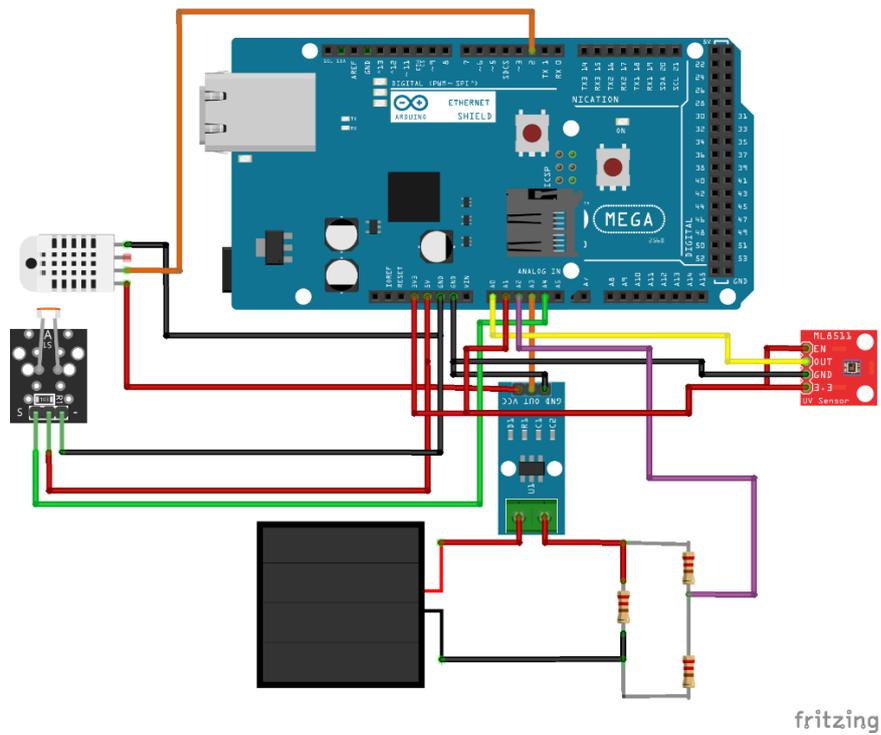


Figura 43: Diagrama de circuito pictórico del divisor de tensión.

Elaboración propia.

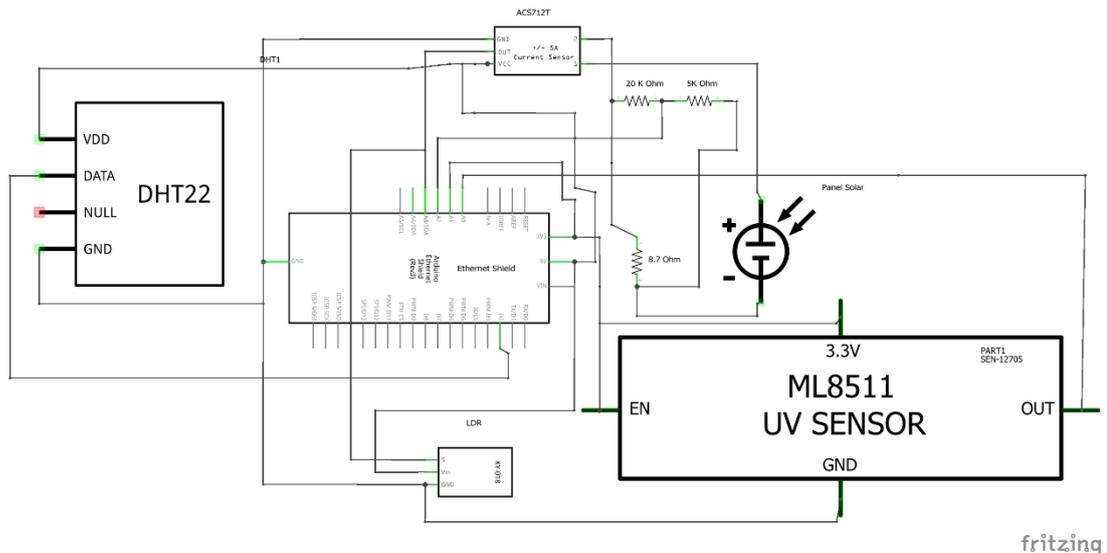


Figura 44: Diagrama de circuito esquemático del divisor de tensión.

Elaboración propia.

En la Fig. 43 y Fig. 44 se grafica los diagramas de conexión que contiene el sistema implementado con todos los sensores administrados por el módulo Arduino e interconectados con la nube mediante el módulo Ethernet Shield; en esta sección se añadió el divisor de tensión que está contenida por dos resistencias conectadas a la carga constante, finalmente este divisor de tensión será leído por el Arduino mediante el pin de modo analógico número 2 o también llamado A2 que se encuentra en le modulo Ethernet Shield.

3.5.10. Raspberry Pi

El dispositivo embebido de modelo Raspberry Pi es como una minicomputadora el cual contiene dentro un sistema operativo de Kernel Linux. Las características de este dispositivo son las siguientes:

- Voltaje de alimentación: 5V.
- Microchip controlador: Ethernet Wiznet W5100.
- Velocidad de transmisión de datos por ethernet: 10 Mbps o 100 Mbps.
- Protocolos soportados: UDP y TCP.

- Compatibles con Arduino Uno y Arduino Mega.

Este dispositivo embebido sirva en esta investigación para controlar mediante un script o algoritmo actuando como un modo de enrutador o router y poder garantizar la comunicación del sistema de adquisición de datos con el servidor de base de datos. En la Fig. 45 se muestra el dispositivo que por medio de la interfaz WiFi se conecta al internet y su puerto ethernet comunica al Arduino.

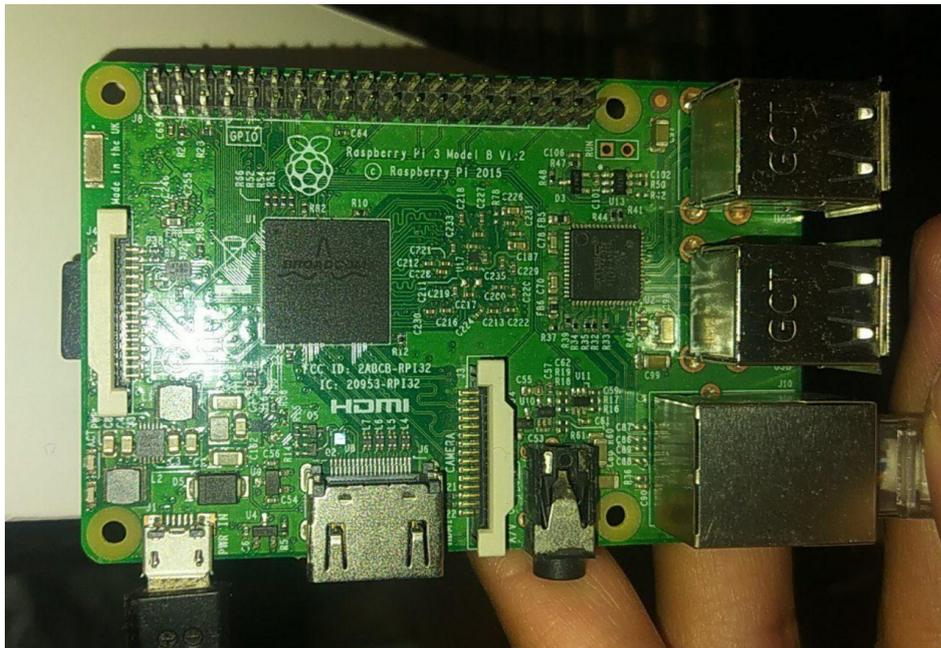


Figura 45: *Raspberry Pi 3 Modelo B.*

Elaboración propia.

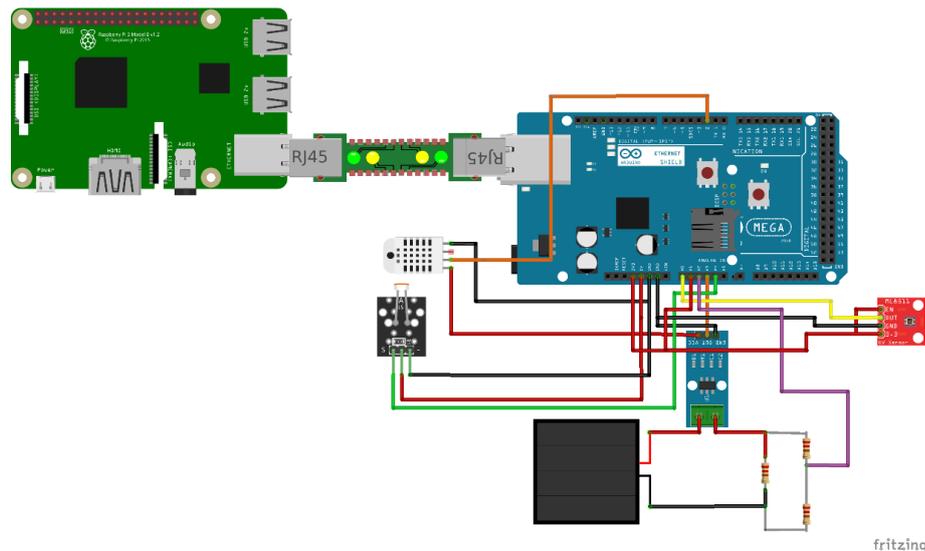


Figura 46: *Diagrama de circuito pictórico Raspberry Pi 3 Modelo B con módulo Arduino Ethernet Shield.*

Elaboración propia.

En la Fig. 46 se tiene el diagrama completo del sistema para la adquisición de datos con los sensores de potencia del panel solar, así como también los sensores que miden los factores meteorológicos y finalmente los dispositivos que comunican a la nube toda la información recolectada y procesada la cual gobierna el Arduino Mega; en la imagen también se puede observar que el módulo Ethernet Shield ayuda a la conexión de internet mediante un cable Ethernet que está conectado a una Raspberry Pi el cual funciona como un router que habilita le internet al sistema de adquisición de datos.

3.5.11. Servidor de internet

La comunicación del internet con el servidor contiene un sistema operativo de distribución Linux, el sistema operativo ejecutado en este caso fue un Ubuntu Server versión 20.04, este servidor contiene un servidor web Apache 2, también contiene una base de datos MariaDB; la programación de la intercomunicación esta ejecutada en el lenguaje PHP. El servidor web Apache permite la creación del servidor HTTP o también conocido como un servidor de tipo web, la base de datos MariaDB se encarga de guardar los registros en tablas como consecuencia de la interacción de información del código



PHP; unificando todas estas tareas y herramientas tenemos el sistema completo de almacenamiento de datos del sistema de adquisición de datos para luego realizar el modelo de Machine Learning.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. PREPROCESAMIENTO DE DATOS

El preprocesamiento de datos para esta investigación será de vital importancia ya que se tendrá que hacer varios procesos para tener finalmente una data limpia y pueda ser las entradas para los modelos propuestos; para lo cual utilizaremos la plataforma Colab que está impulsada por la empresa Google la cual se hace el uso de la versión gratuita para el preprocesamiento como también para la ejecución de los modelos de Machine Learning. Esta plataforma permite ejecutar líneas de código en el lenguaje de Python, que como consecuencia permitirá hacer todo el procesamiento a nivel de software bajo esta plataforma.

En la siguiente figura se muestra las características de la data original la cual tendremos que modificar para obtener valores correctos de voltaje y amperaje.

```
[82] import pandas as pd
data = pd.read_csv("Data.csv", delimiter=";")

[83] data.head(5)
```

| | ID | Temperatura | Humedad | UV | Voltaje Panel | Amperaje Panel | Iluminancia | IP | LUGAR | Date Time |
|---|-------|-------------|---------|------|---------------|----------------|-------------|---------------|-------|------------------|
| 0 | 17036 | 40.4 | 14.0 | 3.83 | 880.50 | 551.65 | 8.80 | 190.42.16.137 | Puno | 06/12/2021 12:41 |
| 1 | 17037 | 42.1 | 13.6 | 4.10 | 886.30 | 551.40 | 8.15 | 190.42.16.137 | Puno | 06/12/2021 12:42 |
| 2 | 17038 | 42.9 | 12.9 | 3.49 | 875.05 | 551.30 | 8.45 | 190.42.16.137 | Puno | 06/12/2021 12:43 |
| 3 | 17039 | 43.8 | 12.7 | 4.24 | 873.40 | 551.20 | 8.35 | 190.42.16.137 | Puno | 06/12/2021 12:44 |
| 4 | 17040 | 44.5 | 12.4 | 3.50 | 860.90 | 551.20 | 8.80 | 190.42.16.137 | Puno | 06/12/2021 12:45 |

```
[84] data.describe()
```

| | ID | Temperatura | Humedad | UV | Voltaje Panel | Amperaje Panel | Iluminancia |
|-------|--------------|--------------|--------------|--------------|---------------|----------------|--------------|
| count | 12770.000000 | 12770.000000 | 12770.000000 | 12770.000000 | 12770.000000 | 12770.000000 | 12770.000000 |
| mean | 23420.500000 | 23.143336 | 42.663516 | 2.912142 | 539.331006 | 527.332138 | 169.326065 |
| std | 3686.525804 | 10.447881 | 26.287258 | 1.889787 | 357.531404 | 16.642199 | 319.050423 |
| min | 17036.000000 | 5.200000 | 8.600000 | 0.620000 | 0.000000 | 507.800000 | 7.450000 |
| 25% | 20228.250000 | 13.800000 | 20.000000 | 1.220000 | 148.850000 | 509.350000 | 12.600000 |
| 50% | 23420.500000 | 22.700000 | 35.300000 | 2.330000 | 745.325000 | 526.325000 | 22.700000 |
| 75% | 26612.750000 | 32.100000 | 63.975000 | 4.130000 | 855.737500 | 544.600000 | 88.087500 |
| max | 29805.000000 | 49.600000 | 99.900000 | 8.910000 | 1023.000000 | 612.350000 | 1021.250000 |

Figura 47: Data original.

Elaboración propia.



Para el siguiente paso utilizare una función denominada “remap” la cual convierte la data de un rango a otro rango muy distinto esto para poder modificar el rango del voltaje y amperaje, los rangos modificados son los siguientes (según la hoja de especificaciones del panel solar):

- Voltaje mínimo obtenido: 0.00
- Voltaje mínimo del panel: 0.00
- Amperaje mínimo obtenido: 507.80
- Amperaje mínimo del panel: 0.00
- Voltaje máximo obtenido: 1023.00
- Voltaje máximo del panel: 18.60
- Amperaje máximo obtenido: 612.35
- Amperaje máximo del panel: 5.38

Se realiza la siguiente fórmula para aplica la conversión entre estos rangos (Penguin, 2013): $NewValue = (((OldValue - OldMin) * (NewMax - NewMin)) / (OldMax - OldMin)) + NewMin$. Luego de aplicar esta conversión tenemos que los valores obtenidos del sistema de adquisición de datos con respecto a las variables de voltaje y amperaje están dentro del rango establecido por el fabricante del panel solar, se muestra en a la siguiente imagen la conversión con la función “remap”.

```
[86] data["V. Panel"] = remap(data["Voltaje Panel"], 0.0, 1023.0 , 0, 18.6 )
data["A. Panel"] = remap(data["Amperaje Panel"], 507.8, 612.35, 0, 5.38 )
data.head(5)
```

| | ID | Temperatura | Humedad | UV | Voltaje Panel | Amperaje Panel | Iluminancia | IP | LUGAR | Date Time | V. Panel | A. Panel |
|---|-------|-------------|---------|------|---------------|----------------|-------------|---------------|-------|------------------|-----------|----------|
| 0 | 17036 | 40.4 | 14.0 | 3.83 | 880.50 | 551.65 | 8.80 | 190.42.16.137 | Puno | 06/12/2021 12:41 | 16.009091 | 2.256461 |
| 1 | 17037 | 42.1 | 13.6 | 4.10 | 886.30 | 551.40 | 8.15 | 190.42.16.137 | Puno | 06/12/2021 12:42 | 16.114545 | 2.243596 |
| 2 | 17038 | 42.9 | 12.9 | 3.49 | 875.05 | 551.30 | 8.45 | 190.42.16.137 | Puno | 06/12/2021 12:43 | 15.910000 | 2.238451 |
| 3 | 17039 | 43.8 | 12.7 | 4.24 | 873.40 | 551.20 | 8.35 | 190.42.16.137 | Puno | 06/12/2021 12:44 | 15.880000 | 2.233305 |
| 4 | 17040 | 44.5 | 12.4 | 3.50 | 860.90 | 551.20 | 8.80 | 190.42.16.137 | Puno | 06/12/2021 12:45 | 15.652727 | 2.233305 |

Figura 48: Conversión de rangos.

Elaboración propia.

Luego de hacer la conversión multiplicaremos el voltaje obtenido con la corriente o también llamado amperaje, ya que al multiplicar estas dos variables nos dará como resultado la potencia del sistema fotovoltaico. La siguiente imagen muestra en la segunda caja de código tenemos la descripción de la data casi preprocesada; vemos que la potencia se encuentra en la última columna teniendo el rango de 0W como mínimo valor y 99W como máximo valor, esta relación es correcta ya que el panel tiene una potencia máxima de 100 W.

```
[87] data["Potencia"] = data["V. Panel"]*data["A. Panel"]
data.head(5)
```

| | ID | Temperatura | Humedad | UV | Voltaje Panel | Amperaje Panel | Iluminancia | IP | LUGAR | Date Time | V. Panel | A. Panel | Potencia |
|---|-------|-------------|---------|------|---------------|----------------|-------------|---------------|-------|------------------|-----------|----------|-----------|
| 0 | 17036 | 40.4 | 14.0 | 3.83 | 880.50 | 551.65 | 8.80 | 190.42.16.137 | Puno | 06/12/2021 12:41 | 16.009091 | 2.256461 | 36.123890 |
| 1 | 17037 | 42.1 | 13.6 | 4.10 | 886.30 | 551.40 | 8.15 | 190.42.16.137 | Puno | 06/12/2021 12:42 | 16.114545 | 2.243596 | 36.154536 |
| 2 | 17038 | 42.9 | 12.9 | 3.49 | 875.05 | 551.30 | 8.45 | 190.42.16.137 | Puno | 06/12/2021 12:43 | 15.910000 | 2.238451 | 35.613747 |
| 3 | 17039 | 43.8 | 12.7 | 4.24 | 873.40 | 551.20 | 8.35 | 190.42.16.137 | Puno | 06/12/2021 12:44 | 15.880000 | 2.233305 | 35.464878 |
| 4 | 17040 | 44.5 | 12.4 | 3.50 | 860.90 | 551.20 | 8.80 | 190.42.16.137 | Puno | 06/12/2021 12:45 | 15.652727 | 2.233305 | 34.957308 |

```
[88] data.describe()
```

| | ID | Temperatura | Humedad | UV | Voltaje Panel | Amperaje Panel | Iluminancia | V. Panel | A. Panel | Potencia |
|-------|--------------|--------------|--------------|--------------|---------------|----------------|--------------|--------------|--------------|--------------|
| count | 12770.000000 | 12770.000000 | 12770.000000 | 12770.000000 | 12770.000000 | 12770.000000 | 12770.000000 | 12770.000000 | 12770.000000 | 12770.000000 |
| mean | 23420.500000 | 23.143336 | 42.663516 | 2.912142 | 539.331006 | 527.332138 | 169.326065 | 9.806018 | 1.005097 | 13.871087 |
| std | 3686.525804 | 10.447881 | 26.287258 | 1.889787 | 357.531404 | 16.642199 | 319.050423 | 6.500571 | 0.856385 | 14.043445 |
| min | 17036.000000 | 5.200000 | 8.600000 | 0.620000 | 0.000000 | 507.800000 | 7.450000 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 20228.250000 | 13.800000 | 20.000000 | 1.220000 | 148.850000 | 509.350000 | 12.600000 | 2.706364 | 0.079761 | 0.516803 |
| 50% | 23420.500000 | 22.700000 | 35.300000 | 2.330000 | 745.325000 | 526.325000 | 22.700000 | 13.551364 | 0.953271 | 7.519014 |
| 75% | 26612.750000 | 32.100000 | 63.975000 | 4.130000 | 855.737500 | 544.600000 | 88.087500 | 15.558864 | 1.893678 | 29.096287 |
| max | 29805.000000 | 49.600000 | 99.900000 | 8.910000 | 1023.000000 | 612.350000 | 1021.250000 | 18.600000 | 5.380000 | 98.632304 |

Figura 49: Descripción de la conversión y cálculo de la potencia.

Elaboración propia.

Para la siguiente imagen se nota que eliminamos las variables que no podrán ser la entrada para los modelos ya que son de tipo String como también no son de importancia

como el “ID” por ejemplo, las variables se eliminan para aplicar directamente el modelo posteriormente y tener la data limpia, las variables eliminadas son: "ID", "Voltaje Panel", "Amperaje Panel", "IP", "LUGAR", "Date Time", "V. Panel", "A. Panel". Posterior a este paso finalmente ya se tiene la data preprocesada y lista para utilizar y aplicar el modelo así que pasamos al guardado de la nueva data la cual se le guarda con el nombre de “data_preprocesada.csv”.

```
[95] data_preprocesada = data.drop(["ID", "Voltaje Panel", "Amperaje Panel", "IP", "LUGAR", "Date Time", "V. Panel", "A. Panel"], axis=1)
data_preprocesada.head(5)
```

| | Temperatura | Humedad | UV | Iluminancia | Potencia |
|---|-------------|---------|------|-------------|-----------|
| 0 | 40.4 | 14.0 | 3.83 | 8.80 | 36.123890 |
| 1 | 42.1 | 13.6 | 4.10 | 8.15 | 36.154536 |
| 2 | 42.9 | 12.9 | 3.49 | 8.45 | 35.613747 |
| 3 | 43.8 | 12.7 | 4.24 | 8.35 | 35.464878 |
| 4 | 44.5 | 12.4 | 3.50 | 8.80 | 34.957308 |

```
[96] data_preprocesada.describe()
```

| | Temperatura | Humedad | UV | Iluminancia | Potencia |
|-------|--------------|--------------|--------------|--------------|--------------|
| count | 12770.000000 | 12770.000000 | 12770.000000 | 12770.000000 | 12770.000000 |
| mean | 23.143336 | 42.663516 | 2.912142 | 169.326065 | 13.871087 |
| std | 10.447881 | 26.287258 | 1.889787 | 319.050423 | 14.043445 |
| min | 5.200000 | 8.600000 | 0.620000 | 7.450000 | 0.000000 |
| 25% | 13.800000 | 20.000000 | 1.220000 | 12.600000 | 0.516803 |
| 50% | 22.700000 | 35.300000 | 2.330000 | 22.700000 | 7.519014 |
| 75% | 32.100000 | 63.975000 | 4.130000 | 88.087500 | 29.096287 |
| max | 49.600000 | 99.900000 | 8.910000 | 1021.250000 | 98.632304 |

```
[98] data_preprocesada.to_csv('data_preprocesada.csv', sep=';', index=False)
```

Figura 50: Eliminar variables y guardar data preprocesada.

Elaboración propia.

4.2. MODELOS DE ENSAMBLE

El conjunto de modelos de aprendizaje automático es la referencia del concepto de un modelo de tipo ensamble. Cada modelo nos mostrara una proporción u otra predicción. Combine las predicciones de varios modelos para obtener una sola predicción. La ventaja obtenida al combinar varios modelos es que cada modelo funciona desde el punto de vista de la palabra, por lo que el error tiende a ser compensado. Esto lleva a un mejor error en la generalización. Para la investigación realizada se tiene tres modelos de

este tipo, los cuales son Bagging, Gradient Boosting y también XGboost, estos modelos serán alimentados por la misma data y comprando los resultados que generan.

El primer paso para realizar la regresión aplicando los modelos propuestos como se muestra en la siguiente imagen se cargó el data set al entorno de Colab, luego se visualiza la información de los datos con el comando “.info ()”, vemos que se tiene 12770 datos por cada variable en el data set.

```
[113] import pandas as pd
data = pd.read_csv("data_preprocesada.csv", delimiter=";")
data.head(5)
```

| | Temperatura | Humedad | UV | Iluminancia | Potencia |
|---|-------------|---------|------|-------------|-----------|
| 0 | 40.4 | 14.0 | 3.83 | 8.80 | 36.123890 |
| 1 | 42.1 | 13.6 | 4.10 | 8.15 | 36.154536 |
| 2 | 42.9 | 12.9 | 3.49 | 8.45 | 35.613747 |
| 3 | 43.8 | 12.7 | 4.24 | 8.35 | 35.464878 |
| 4 | 44.5 | 12.4 | 3.50 | 8.80 | 34.957308 |

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12770 entries, 0 to 12769
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  ---            -
0   Temperatura     12770 non-null  float64
1   Humedad         12770 non-null  float64
2   UV              12770 non-null  float64
3   Iluminancia     12770 non-null  float64
4   Potencia        12770 non-null  float64
dtypes: float64(5)
memory usage: 499.0 KB
```

Figura 51: Cargar y visualizar data.

Elaboración propia.

Posteriormente como se muestra en la imagen siguiente dividiremos la cantidad de datos, el 80% de datos será para entrenar el modelo y el 20% restante está establecido para la etapa de test del modelo, así podremos asegurarnos de que el modelo se entrenó correctamente. También en la imagen se muestra una semilla o llamado en el código como

“random_state” el cual nos ayudará a aleatorizar la data que será tomada en porcentajes para el entrenamiento y test (80% y 20% respectivamente), esta aleatorización es de vital importancia ya que garantiza al modelo de tener un sobreajuste en su aprendizaje con respecto a la predicción.

4.2.1. Bagging

Para ejecutar el modelo Bagging de tipo regresión ya que este caso en específico tendrá que cumplir una regresión al tener un rango de valores a calcular como salida, ya que si se tendría una salida binaria se optaría por utilizar el modelo de tipo clasificador. Luego de llamar a la librería se tendrá que establecer el modelo con los parámetros que se quiere parametrizar, para esta investigación se asigna el hiperparametro “n_estimators=500” el cual es el número de árboles potenciados por gradiente. Equivalente al número de rondas de refuerzo.

```
[179] from sklearn.ensemble import BaggingRegressor  
  
regresion = BaggingRegressor(n_estimators=500)  
regresion.fit(X_train,y_train)  
bagg_predict = regresion.predict(X_test)
```

Figura 52: Regresión con modelo Bagging.

Elaboración propia.

Vemos en la siguiente imagen que el resultado de la regresión de tipo Bagging; la precisión del modelo con la data de entrenamiento alcanza un 98.5%, mientras que en la data de test alcanza un 89.3%.

```
[180] from sklearn.metrics import mean_squared_error, mean_absolute_error, mean_absolute_percentage_error
import numpy as np

mse = mean_squared_error(y_test, bagg_predict)
rmse = np.sqrt(mean_squared_error(y_test, bagg_predict))
mae = mean_absolute_error(y_test, bagg_predict)
bagg_score_test = regresion.score(X_test, y_test)
bagg_score_train = bagg_reg.score(X_train, y_train)

print("MAE: %f" % (mae))
print("RMSE: %f" % (rmse))
print("MSE: {:.4f}".format(mse))
print("Score Train: %f" % (bagg_score_train))
print("Score Test: %f" % (bagg_score_test))

MAE: 1.809840
RMSE: 4.455827
MSE: 19.8544
Score Train: 0.985140
Score Test: 0.893488
```

Figura 53: Resultados del modelo Bagging.

Elaboración propia.

En la siguiente imagen se muestra la predicción de los datos con los datos reales graficados para tener mejor un panorama, los gráficos fueron graficado desde el dato número 300 al 500 para no saturar los puntos a graficar y así poder tener una gráfica más entendible y con posibilidad analizar y notar fácilmente que el modelo se ajusta correctamente en la mayoría de los casos en la predicción de la potencia del sistema fotovoltaico implementado para esta investigación, se utilizó la librería “matplotlib” para poder graficar estos datos mostrados en la siguiente imagen.

```
import matplotlib.pyplot as plt

x_ax = range(len(y_test[300:500]))
plt.figure(figsize=(24,10))
plt.plot(x_ax, y_test[300:500], lw=1.5, color="orange", label="Potencia Original")
plt.plot(x_ax, bagg_predict[300:500], lw=1.5, color="green", label="Potencia Predecida")
plt.legend()
plt.show()
```

Figura 54: Código para graficar valores reales y predichos.

Elaboración propia.

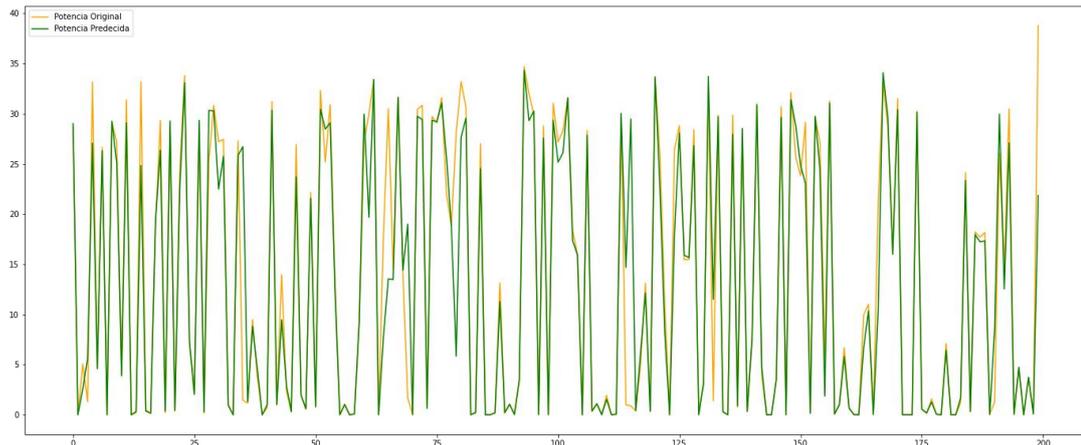


Figura 55: Datos predichos y datos reales.

Elaboración propia.

4.2.2. Gradient Boosting

En la siguiente imagen tenemos la aplicación del modelo Gradient Boosting de tipo regresión también ya que las predicciones serán de tipo numérica desde 0 a 100 aproximadamente ya que el panel fotovoltaico que monitorizo el sistema de adquisición de datos contiene ese valor de potencia máxima (100W). Se tiene también que el hiperparametro ajustado fue “n_estimators=500” el cual es el número de árboles potenciados por gradiente; equivalente al número de rondas de refuerzo como también se ajustó el “max_depth” el cual es la máxima cantidad de árboles para el aprendizaje del modelo, el valor asignado es de 5 árboles de profundidad.

```
[7] from sklearn.ensemble import GradientBoostingRegressor  
  
regresion = GradientBoostingRegressor(n_estimators=500, max_depth=5)  
regresion.fit(X_train,y_train)  
bagg_predict = regresion.predict(X_test)
```

Figura 56: Regresión con modelo Gradient Boosting.

Elaboración propia.

Posteriormente se entrena al modelo con los hiperparametros ajustados con los valores de entrenamiento, este proceso tarda en durar aproximadamente 120 segundos, dependiendo el tipo de hardware será asignado a la cuenta en donde se está realizando en el entrenamiento del modelo que se está ejecutando.

En la siguiente imagen se muestra los resultados generados por el modelo mencionado anteriormente; se nota que la precisión con los datos de entrenamiento fue de 98.1% mientras que los datos de test mostraron una precisión del 88.0% dando lugar a un rendimiento óptimo para la predicción de la potencia en base a las variables meteorológicas adquiridas.

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, mean_absolute_percentage_error
import numpy as np

mse = mean_squared_error(y_test, bagg_predict)
rmse = np.sqrt(mean_squared_error(y_test, bagg_predict))
mae = mean_absolute_error(y_test, bagg_predict)
bagg_score_test = regresion.score(X_test, y_test)
bagg_score_train = regresion.score(X_train, y_train)

print("MAE: %f" % (mae))
print("RMSE: %f" % (rmse))
print("MSE: {:.4f}".format(mse))
print("Score Train: %f" % (bagg_score_train))
print("Score Test: %f" % (bagg_score_test))
```

MAE: 2.089756
RMSE: 4.727685
MSE: 22.3510
Score Train: 0.981437
Score Test: 0.880094

Figura 57: Resultados del modelo Gradient Boosting.

Elaboración propia.

Finalmente, en la siguiente la gráfica siguiente se tiene los datos predichos y datos reales graficados para una comparación más practica y sencilla.

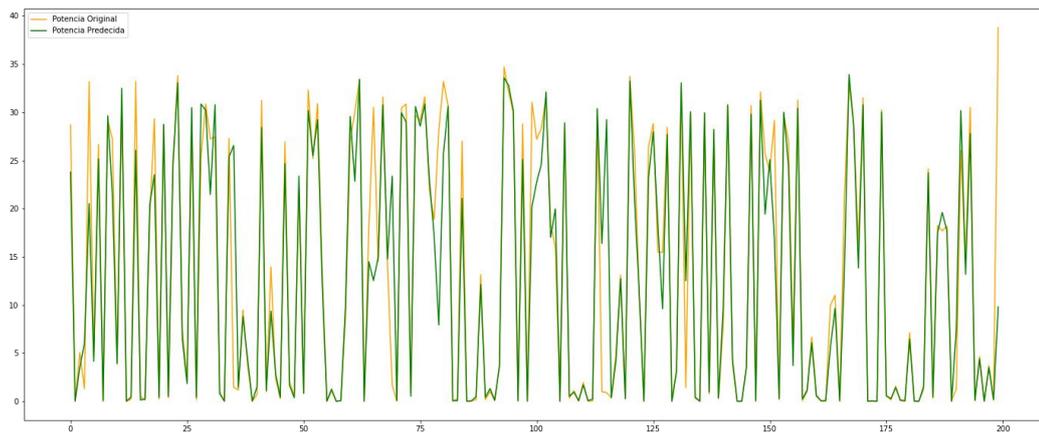


Figura 58: Datos predichos y datos reales.

Elaboración propia.

4.2.3. XGBoost

Para el último modelo denominado XGBoost como se muestra en la siguiente imagen se ajusta el hiperparametro “n_estimators=500” el cual es el número de árboles potenciados por gradiente; equivalente al número de rondas de refuerzo como también se ajustó el “max_depth” el cual es la máxima cantidad de árboles para el aprendizaje del modelo, el valor asignado es de 5 árboles de profundidad. Posteriormente se entrena el modelo para mostrar los resultados.

```
[10] import xgboost as xgb

regresion = xgb.XGBRegressor(n_estimators=500, objective="reg:squarederror", max_depth=5)
regresion.fit(X_train,y_train)
bagg_predict = regresion.predict(X_test)
```

Figura 59: Regresión con modelo XGBoost.

Elaboración propia.

En la siguiente imagen se puede visualizar que la precisión de la data de entrenamiento es de 97.8%, mientras que le precisión de la data de test es equivalente a 88.6%, dando lugar a un rendimiento bueno y fiable para aplicaciones en predicciones a este tipo de sistemas.

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, mean_absolute_percentage_error
import numpy as np

mse = mean_squared_error(y_test, bagg_predict)
rmse = np.sqrt(mean_squared_error(y_test, bagg_predict))
mae = mean_absolute_error(y_test, bagg_predict)
bagg_score_test = regresion.score(X_test, y_test)
bagg_score_train = regresion.score(X_train, y_train)

print("MAE: %f" % (mae))
print("RMSE: %f" % (rmse))
print("MSE: {:.4f}".format(mse))
print("Score Train: %f" % (bagg_score_train))
print("Score Test: %f" % (bagg_score_test))
```

MAE: 2.050168
RMSE: 4.619583
MSE: 21.3405
Score Train: 0.978296
Score Test: 0.885515

Figura 60: Resultados del modelo XGBoost.

Elaboración propia.

En la siguiente imagen también se muestra los datos reales como también los datos predichos por el modelo XGBoost para demostrar visualmente que el modelo se ajusta bien a los datos y al objetivo, sin embargo, también se muestra en el gráfico que no se comete un sobreajuste lo cual es positivo para los resultados finales.

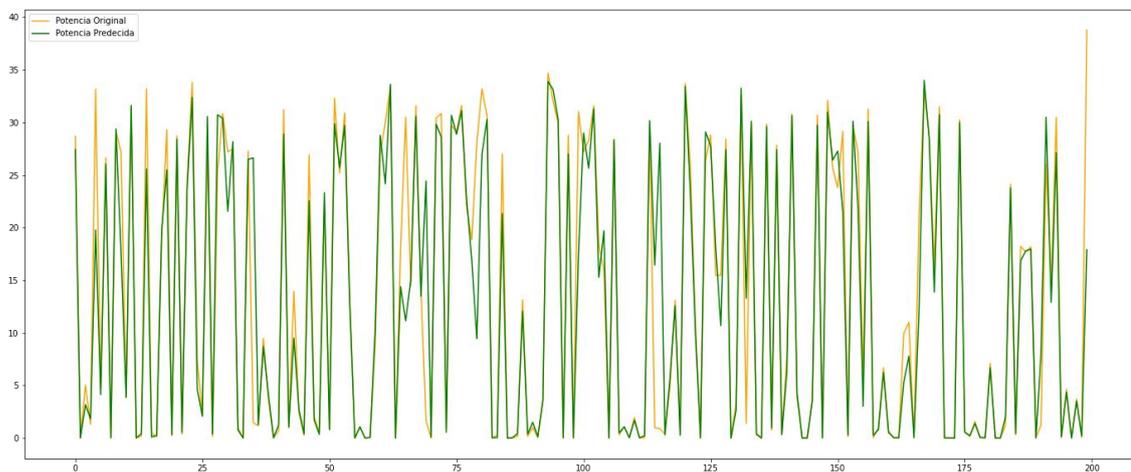


Figura 61: Datos predichos y datos reales.

Elaboración propia.

En la siguiente imagen se muestra un fragmento de la gráfica del árbol de decisión que contiene el modelo XGBoost el cual muestra con una serie de árboles las decisiones que toma el modelo de acuerdo a los datos de entrada que tiene como datos,

posteriormente llega un punto donde decide el resultado final que el la predicción o datos predichos que calcula en base al aprendizaje.

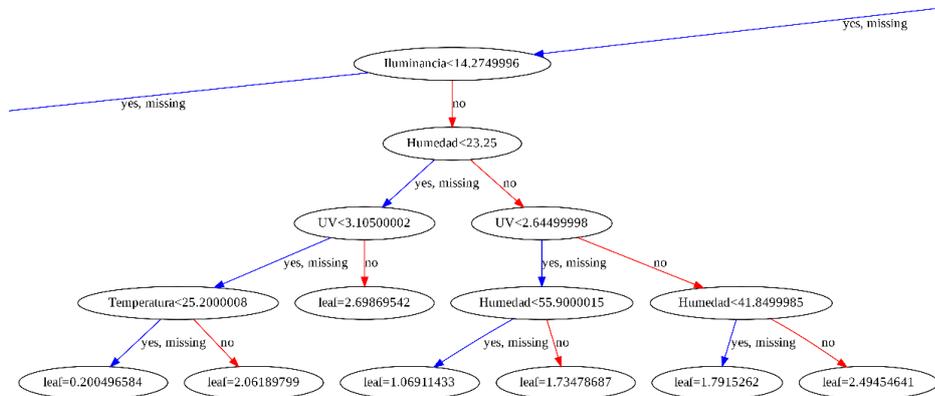


Figura 62: Fragmento del árbol de decisión del modelo XGBoost.

Elaboración propia.

4.3. COMPARANDO MODELOS

Teniendo todos los modelos y sus resultados, se compara estos resultados y poder así determinar qué modelo se ajusta mejor a la data y el sistema de predicción que se pretende establecer en este trabajo de investigación. Se muestra en la Tabla 1 los resultados; se tiene la métrica MAE ($\frac{1}{n} \sum_{j=1}^n |Y_j - \hat{Y}_j|$) que mide el error absoluto es la cantidad de error en sus medidas. Es la diferencia entre el valor medido y el valor verdadero o real, por otro lado, MSE ($\sqrt{\frac{1}{n} \sum_{j=1}^n (Y_j - \hat{Y}_j)^2}$) o también llamado error cuadrático medio es una métrica básica para medir la diferencia cuadrática de errores entre valores predichos este error también es conocido como desviación cuadrática media, le métrica RMSE () es la desviación estándar de los valores residuales (errores de predicción) el cual indicara el nivel de concentración de los datos en la línea de mejor ajuste al modelo que se esté aplicando, finalmente la precisión es el porcentaje de aciertos que tuvo el modelo con respecto a la predicción de los datos, mientras más cerca del 100%

se encuentre el modelo es más óptimo, sin embargo si llegaría al 100% se considera que el modelo está sobreajustado por lo tanto no es un modelo válido estadísticamente.

Tabla 1: Resultados de los modelos.

| | Bagging | Gradient Boosting | XGBoost |
|-----------------------|----------------|--------------------------|----------------|
| MAE | 1,810168 | 2,089756 | 2,050168 |
| RMSE | 4,442227 | 4,727685 | 4,619583 |
| MSE | 19,733400 | 22,351 | 21,3405 |
| Accuracy Train | 0,985170 | 0,981437 | 0,978296 |
| Accuracy Test | 0,894137 | 0,880094 | 0,885515 |

Elaboración propia.

En la Tabla 1 notamos que el modelo con mejores resultados es Bagging con una precisión del 89.4% en el data set de test el cual es el más relevante ya que esa data no interactuó en la creación del modelo por lo tanto tiene más variabilidad y es la más óptima para corroborar de que un modelo se entrenó de manera correcta.

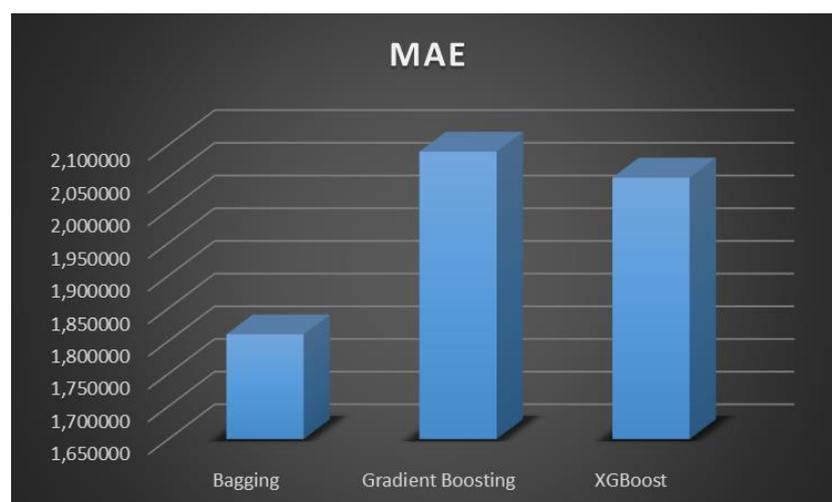


Figura 63: Resultados de la métrica MAE.

Elaboración propia.

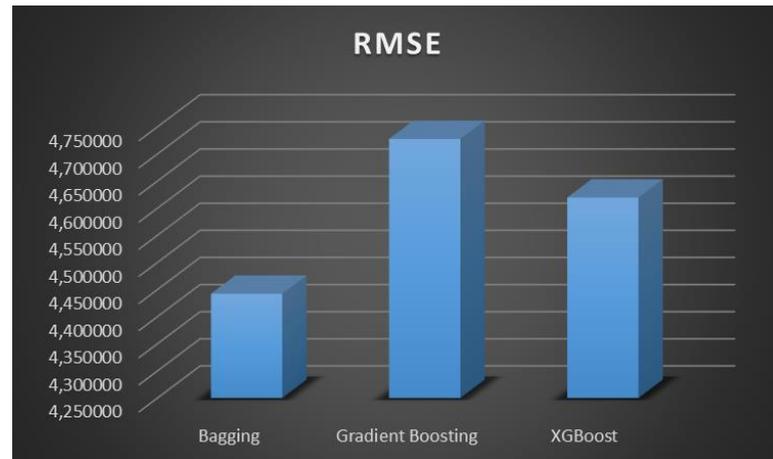


Figura 64: Resultados de la métrica RMSE.

Elaboración propia.

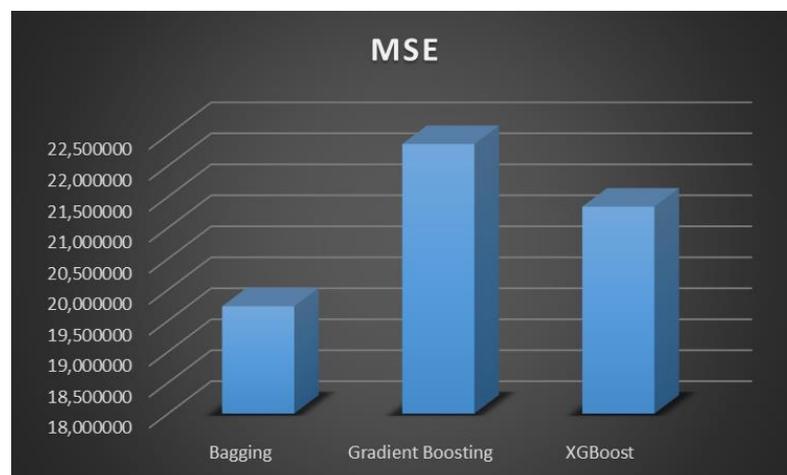


Figura 65: Resultados de la métrica MSE.

Elaboración propia.

En la Fig. 63, Fig. 64 y Fig. 65 se muestra que el mejor resultado contiene el modelo Bagging, dejando al modelo XGBoost en segundo lugar y por último el modelo Gradient Boosting, esto muestra que el modelo Bagging contiene un error menor a comparación de los demás modelos para la predicción de la potencia en base a las variables meteorológicas de entrada.

En la Fig. 66 se muestra que la precisión de los modelos teniendo que Bagging es el modelo líder con respecto a la predicción de los datos de la variable de salida, sin embargo, esta precisión es de la data de entrenamiento lo cual no es fiable para el

rendimiento con datos nuevos para el modelo, como segundo lugar se tiene al modelo Gradient Boosting y finalmente el modelo XGBoost con menos rendimiento.



Figura 66: Resultados de la métrica Accuracy con data de entrenamiento.
Elaboración propia.

En la Fig. 67 se muestra el verdadero rendimiento ante datos nuevos de los modelos; se muestra que el modelo Bagging es el que mejor se ajusta a las predicciones de la potencia en el sistema fotovoltaico, como segundo lugar se tiene a XGBoost con un rendimiento optimo y finalmente el modelo con menos acierto es Gradient Boosting. Esta métrica es más fiable ya que fue calculada con la data de test la cual es data que no fue influenciada en el aprendizaje del modelo.

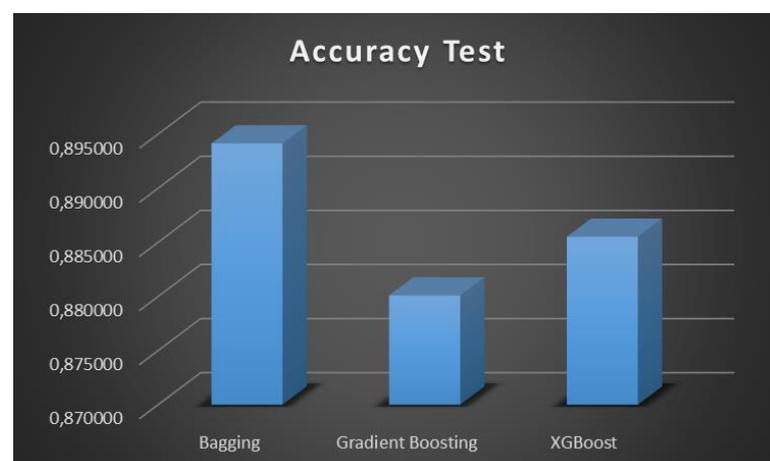


Figura 67: Resultados de la métrica Accuracy con data de test.
Elaboración propia.



V. CONCLUSIONES

El primer objetivo del diseño de un sistema de adquisición de datos se logró estableciendo un prototipo propio, aún queda pendiente incrementar más variables meteorológicas o factores que influyen directamente en la recolección de energía mediante celdas fotovoltaicas para hacer un modelo con más variables de entrada; el prototipo presentado en esta investigación funciona correctamente y con muy poca manipulación del sistema en conjunto ya que todos los procesos están automatizados. Los datos adquiridos mediante los 20 días fueron óptimos para el entrenamiento de los modelos, ya que en su totalidad se recolectó 12770 datos de cada una de las variables de entrada (Temperatura, Humedad, Radiación UV e Iluminancia) mientras que la variable de salida fue la potencia del sistema fotovoltaico.

El segundo objetivo también fue plenamente alcanzado al implementar los modelos de ensamble en los datos adquiridos por el sistema de adquisición de datos, los modelos fueron Bagging, Gradient Boosting y XGBoost; de acuerdo a los resultados obtenidos el mejor modelo para la predicción de la potencia del panel solar fue Bagging teniendo como resultado la métrica MAE equivalente a 1.81, por otro lado el RMSE fue de 4.44, un MSE de 19.73, la precisión del modelo con la data de entrenamiento fue de 98.52% y finalmente la precisión con la data de test alcanzó el 89.41% lo que se traduce en un rendimiento bastante alto y óptimo sin tener problemas de sobreajuste; el modelo que alcanzó mejores resultados fue ajustado con un hiperparámetro el cual incrementó el rendimiento del modelo sin comprometer la memorización de los datos, el hiperparámetro ajustado fue “n_estimators” con un valor igual a 500, el cual es equivalente al número de rondas de refuerzo.



VI. RECOMENDACIONES

La predicción del modelo se pudo realizar correctamente con altos valores de precisión en los distintos casos con respecto a los algoritmos utilizados en esta investigación, sin embargo, este sistema puede ser mejorado implementando el sistema de adquisición de datos con sensores más sofisticados utilizando también tecnología IoT pero que los dispositivos contengan mayor precisión y como consecuencia los modelos de Machine Learning de tipo ensamble puedan aprender de mejor manera eliminando así pequeños sesgos generados por los sensores; los modelos de ensamble puede ser optimizado mediante la validación cruzada de sus hiperparámetros, esto comprende mayor tiempo de entrenamiento y test sin embargo incrementaría la precisión del modelo para las predicciones.



VII. REFERENCIAS BIBLIOGRÁFICAS

- Nilsson, N. (2005). *INTRODUCTION TO MACHINE LEARNING*. California: Universidad Stanford.
- Achachi, L., & Fernando, D. (2021). *Diseño de un algoritmo utilizando Machine Learning para la predicción de la radiación solar en el sector de Lasso*. Universidad Técnica de Cotopaxi: UTC.
<https://doi.org/http://repositorio.utc.edu.ec/handle/27000/8014>
- Andrade Caicedo, R., & Sanyer Espinoza, O. (2021). *Diseño e implementación de un control basado en redes neuronales artificiales para la posición optima de paneles solares de un sistema fotovoltaico en la escuela primaria Camino al Espacio*. Repositorio Institucional de la Universidad Politécnica Salesiana.
<https://doi.org/http://dspace.ups.edu.ec/handle/123456789/21906>
- Arno, S., Klaus, J., Olindo, I., René, S., & Miro, Z. (2016). *Solar energy: The physics and engineering of photovoltaic conversion, technologies and systems*. Inglaterra: UIT Cambridge Ltd.
- Aston, Z., Zachary, L., Mu, L., & Alexander, S. (2022). *Dive into Deep Learning*. Pensilvania: Carnegie Mellon University.
- AutoSolar. (27 de 01 de 2021). *AutoSolar*. <https://autosolar.pe/aspectos-tecnicos/controlador-de-carga-solar-controlador-mppt-usos-y-ventajas>
- Bentejaca, C., Csorgo, A., & Martinez, G. (2021). A Comparative Analysis of XGBoost. *Artificial Intelligence Review*, <https://doi.org/10.1007/s10462-020-09896-5>.
- Burkov, A. (2019). *The Hundred - Page Machine Learning Book*. Quebec: Andriy Burkov.



- cablematic. (2022). *cablematic*. <https://cablematic.com/en/products/30a-ac712-electronic-current-sensor-AJ030/>
- Castellanos Taita, M. A. (2020). *Modelo de predicción de curvas i-v para paneles solares bajo condiciones difíciles de sombra basado en machine learning*. Univesidad de los Andes Colombia. <https://doi.org/https://repositorio.uniandes.edu.co/flexpaper/handle/1992/50625/24033.pdf?sequence=1&isAllowed=y>
- Corso, C. L., Pereyra, M. F., Martínez, G., Gibellini, F., & Ciceri, L. (2017). *Sistema de soporte de decisión para la gestión de fallos en equipos industriales, basado en métodos de ensamble*. VI Simposio Argentino de Informática Industrial (SII) - JAIIO 46 (Córdoba, 2017). <https://doi.org/http://www.clei2017-46jaiio.sadio.org.ar/sites/default/files/Mem/SII/sii-17.pdf>
- Cruz, E. (2022). *MakerElectronico*. <https://www.makerelectronico.com/producto/dht11-sensor-temperatura-humedad/>
- Daimiel Borrallo, C. (2020). *Análisis predictivo de fallos en plantas fotovoltaicas mediante técnicas de Machine Learning*. Universidad de Sevilla. Departamento de Organización Industrial y Gestión de Empresas I. <https://doi.org/https://hdl.handle.net/11441/108810>
- Dapper Heinen, E. (2018). *Redes Neurais Recorrentes e XGBoost Aplicados à Previsão de Radiação Solar no Horizonte de Curto Prazo*. Universidade Federal de São Carlos – UFSCar.
- Deisenroth, P., Faisal, A., & Ong, C. (2022). *MATHEMATICS FOR MACHINE LEARNING*. Cambridge: Cambridge University Press.



- Díaz, S., & Guerrero, G. (2018). *Modelos de Ensamble de Estrategias de Sistemas Inteligentes Aplicadas al Diagnóstico Médico*. Maestría de Ingeniería de Sistemas y Computación, Universidad Javeriana.
- elosciloscopio. (01 de 01 de 2019). *elosciloscopio*. <https://elosciloscopio.com/tutorial-arduino-mega-pinout/>
- Emilio, M. D. (2013). *Data Acquisition Systems From Fundamentals to Applied Design*. Italia: Springer New York Heidelberg Dordrecht London.
- Energy, V. (2022). *Convertidor CC/CC Orion aislado*. The Netherlands: Victron Energy B.V. .
- Ferreira, A., & Figueiredo , M. (2012). Boosting Algorithms: A Review of Methods, Theory, and Applications. En A. F. Figueiredo, *Boosting Algorithms: A Review of Methods, Theory, and Applications* (pág. Chapter 3). Portugal: Instituto de Telecomunicaciones, Instituto Superior Técnico y Technical University of Lisbon.
- Forbes. (10 de 11 de 2021). *Forbes Staff*. Forbes Staff: <https://forbes.pe/negocios/2021-11-10/inversion-en-proyectos-de-energia-solar-en-el-peru-podrian-superior-los-us1-400-millones-al-2030/>
- Gnoza Tansini, N., & Barberena Allietti, M. E. (2018). *Estudio de factibilidad del uso de Machine Learning con múltiples fuentes de datos en el pronóstico del tiempo*. Universidad ORT Uruguay, Facultad de Ingeniería. <https://doi.org/https://bibliotecas.ort.edu.uy/bibid/87198>



- González, F., Vilorio, R., Garcia, M., Calzada, J., & Arízaga, A. (2010). MEDICIÓN DE TEMPERATURA DE UNA ESTUFA SOLAR SK-14 PARA PARA OPTIMIZAR SU EFICIENCIA. *Semana Internacional de Energia Solar*.
- Hunt, J. (2019). *Advanced Guide to Python 3 Programming*. Reino Unido: Springer Nature Switzerland.
- Keithley. (2001). *Data Acquisition and Control Handbook*. U.S.A.: Keithley Instrument.
- Lad, S., Mahajan, B., Mandkulkar, R., Pugaonkar, V., & Sailakshmi, K. (2017). *Data Acquisition Handbook*. United States: Measurement Computing Corporation.
- learn, s. (01 de 01 de 2022). *scikit-learn*. scikit-learn: https://scikit-learn.org/stable/supervised_learning.html#supervised-learning
- Lipták, B. (2012). *Process Software and Digital Networks*. London: Taylor & Francis Group.
- Lutins, E. (01 de 08 de 2017). *Towards Data Science*. <https://towardsdatascience.com/ensemble-methods-in-machine-learning-what-are-they-and-why-use-them-68ec3f9fef5f>
- Martínez, R. (2020). *El secreto detrás de una tesis: Cómo logré convertir mi experiencia al hacer una tesis en la solución que todo profesional necesita*. Biblioteca Nacional del Perú. <https://doi.org/ISBN-10 : 6120050590>
- Mitchell, R., & Eibe, F. (2017). Accelerating the XGBoost algorithm using GPU computing. *PeerJ Computer Science*.
- Morde, V. (07 de 04 de 2019). *Towards Data Science*. <https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d>



- MPPTSolar. (2021). *MPPTSolar*. mpptsolar.com/es/esquema-funcionamiento-inversor.html
- Ordoñez Palacios, L., León Vargas, D., Bucheli Guerrero, V., & Ordoñez Eraso, H. (2020). Solar Radiation Prediction on Photovoltaic Systems Using Machine Learning Techniques. *Revista Facultad de Ingeniería*, 29, 54. <https://doi.org/https://doi.org/10.19053/01211129.v29.n54.2020.11751>
- Outsourcing, D. &. (25 de Abril de 2017). *Design & Outsourcing*. <https://www.medicaldesignandoutsourcing.com/ultraviolet-light-sensor-for-wearable-devices/>
- Package, P. (01 de 01 de 2022). *XGBoost Python Package*. https://xgboost.readthedocs.io/en/stable/python/python_api.html
- Pelayo, M. H. (2016). *Diseño de una estructura soporte para un módulo integrado de placa fotovoltaica*. Madrid: Universidad Politécnica de Madrid.
- Penguin. (20 de 03 de 2013). *wake-up-neo.net*. <https://www.wake-up-neo.net/es/python/convertir-un-rango-de-numeros-otro-rango-manteniendo-la-relacion/957627295/>
- PROMETEC. (2022). *prometec*. <https://prometec.mx/producto/ldr-sensor-de-luz/>
- python. (01 de 01 de 2022). *python.org*. [python.org: https://www.python.org/downloads/](https://www.python.org/downloads/)
- Ramírez Ríos, J. F. (2016). *Diseño y construcción de un sistema de adquisición de datos para la caracterización de una celda solar*. Universidad Distrital Francisco José de Caldas. <https://doi.org/http://hdl.handle.net/11349/5804>



- Research, N. S. (17 de 06 de 2019). *nationalmaglab*.
<https://nationalmaglab.org/education/magnet-academy/watch-play/interactive/hall-effect-robotop>. (2022). *robotop*. <https://www.robotop.lv/en/svetovye/793-gy-8511-ml8511-uvb-uv-rays-sensor.html>
- Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. (2014). *Metodología de la investigación*. McGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V. <https://doi.org/ISBN:978-1-4562-2396-0>
- Schapire, R., & Freund, Y. (2012). *Boosting: Foundations and Algorithms*. Londres: Instituto Tecnológico de Massachusetts.
- Sen, J. (2021). *Machine Learning Algorithms, Models and Applications*. Londres: Andries Engelbrecht.
- Shalev, S., & Ben, D. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge: Cambridge University Press.
- Smola, A., & Vishwanathan. (2008). *Introduction to Machine Learning*. Cambridge: Universidad de Cambridge.
- Stefan, K. (2006). *Solar Electric Power Generation - Photovoltaic Energy Systems*. Rio de Janeiro: Springer-Verlag Berlin Heidelberg.
- Unit. (01 de 01 de 2021). *Unit Electronics*. <https://uelectronics.com/producto/arduino-ethernet-shield-w-5100/>
- Wallender, L., & Allen, S. (25 de 02 de 2022). *Forbes*. Forbes Advisor: <https://www.forbes.com/advisor/home-improvement/monocrystalline-vs-polycrystalline-solar-panels/>



White, J. (2016). *The Python Book*. Londres: Imagine Publishing Ltd.

Zhu, X., & Goldberg, A. (2009). Introduction to Semi-Supervised Learning. *Morgan & Claypool Publishers*, <https://doi.org/10.2200/S00196ED1V01Y200906AIM006>.



ANEXOS

ANEXO 1 - MUESTRA DE LA DATA PREPROCESADA

Temperatura;Humedad;UV;Iluminancia;Potencia

40.4;14.0;3.83;8.8;36.123889656971414

42.1;13.6;4.1;8.15;36.15453561149511

42.9;12.9;3.49;8.45;35.61374748923954

43.8;12.7;4.24;8.35;35.46487766618845

44.5;12.4;3.5;8.8;34.95730843006828

45.1;12.0;3.81;9.05;34.53622244250247

45.2;11.4;3.76;9.8;33.57054741098212

45.2;11.6;3.89;9.65;34.02001049519588

45.2;11.9;3.41;10.9;32.790572549019615

45.1;11.9;3.3;11.4;33.12971300378247

44.5;11.4;2.43;14.55;28.896768018781813

43.3;12.9;2.12;16.1;26.24528714403719

42.0;12.1;2.34;15.05;28.59075288900481

40.7;13.7;3.08;11.8;33.629479240033035

40.0;14.7;3.51;10.55;34.64886772749012

39.8;14.4;3.74;10.75;35.56141406025825

39.9;14.6;3.42;10.3;34.407346637102684



40.1;14.0;3.78;10.0;35.327899465240655

40.4;14.0;3.82;9.8;35.29432972479455

40.8;13.8;3.6;10.05;34.58174461979909

41.2;13.1;3.46;10.7;34.16627697926178

41.4;13.5;3.35;11.3;34.41396141037347

ANEXO 2 – CODIGO PREPROCESAMIENTO

```
# -*- coding: utf-8 -*-
```

```
import pandas as pd
```

```
data = pd.read_csv("Data.csv", delimiter=";")
```

```
data.head(5)
```

```
data.describe()
```

```
def remap( x, oMin, oMax, nMin, nMax ):
```

```
#range check
```

```
if oMin == oMax:
```

```
print ("Warning: Zero input range")
```

```
return None
```

```
if nMin == nMax:
```

```
print ("Warning: Zero output range")
```

```
return None
```

```
#check reversed input range
```



```
reverseInput = False

oldMin = min( oMin, oMax )

oldMax = max( oMin, oMax )

if not oldMin == oMin:

reverseInput = True

#check reversed output range

reverseOutput = False

newMin = min( nMin, nMax )

newMax = max( nMin, nMax )

if not newMin == nMin :

reverseOutput = True

portion = (x-oldMin)*(newMax-newMin)/(oldMax-oldMin)

if reverseInput:

portion = (oldMax-x)*(newMax-newMin)/(oldMax-oldMin)

result = portion + newMin

if reverseOutput:

result = newMax - portion

return result

data["V. Panel"] = remap(data["Voltaje Panel"], 0.0, 1023.0 , 0, 18.6 )

data["A. Panel"] = remap(data["Amperaje Panel"], 507.8, 612.35, 0, 5.38 )
```



```
data.head(5)
```

```
data["Potencia"] = data["V. Panel"]*data["A. Panel"]
```

```
data.head(5)
```

```
data.describe()
```

```
data_preprocesada = data.drop(["ID", "Voltaje Panel", "Amperaje Panel", "IP",  
"LUGAR", "Date Time", "V. Panel", "A. Panel"], axis=1)
```

```
data_preprocesada.head(5)
```

```
data_preprocesada.describe()
```

```
data_preprocesada.to_csv('data_preprocesada.csv', sep=';', index=False)
```



ANEXO 3 – CODIGO MODELOS DE ENSAMBLE

```
# -*- coding: utf-8 -*-

import pandas as pd

data = pd.read_csv("data_preprocesada.csv", delimiter=";")

data.head(5)

data.info()

from sklearn.model_selection import train_test_split

X = data.drop(["Potencia"], axis=1)

y = data["Potencia"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=2022)

"""## Bagging"""

from sklearn.ensemble import BaggingRegressor

regresion = BaggingRegressor(n_estimators=500)

regresion.fit(X_train,y_train)

bagg_predict = regresion.predict(X_test)

from sklearn.metrics import mean_squared_error, mean_absolute_error,
mean_absolute_percentage_error

import numpy as np

mse = mean_squared_error(y_test, bagg_predict)
```



```
rmse = np.sqrt(mean_squared_error(y_test, bagg_predict))

mae = mean_absolute_error(y_test, bagg_predict)

bagg_score_test = regresion.score(X_test, y_test)

bagg_score_train = regresion.score(X_train, y_train)

print("MAE: %f" % (mae))

print("RMSE: %f" % (rmse))

print("MSE: {:.4f}".format(mse))

print("Score Train: %f" % (bagg_score_train))

print("Score Test: %f" % (bagg_score_test))

import matplotlib.pyplot as plt

x_ax = range(len(y_test[300:500]))

plt.figure(figsize=(24,10))

plt.plot(x_ax, y_test[300:500], lw=1.5, color="orange", label="Potencia Original")

plt.plot(x_ax, bagg_predict[300:500], lw=1.5, color="green", label="Potencia
Predecida")

plt.legend()

plt.show()

"""## Gradient Boosting"""

from sklearn.ensemble import GradientBoostingRegressor
```



```
regression = GradientBoostingRegressor(n_estimators=500, max_depth=5)

regression.fit(X_train,y_train)

bagg_predict = regression.predict(X_test)

from sklearn.metrics import mean_squared_error, mean_absolute_error,
mean_absolute_percentage_error

import numpy as np

mse = mean_squared_error(y_test, bagg_predict)

rmse = np.sqrt(mean_squared_error(y_test, bagg_predict))

mae = mean_absolute_error(y_test, bagg_predict)

bagg_score_test = regression.score(X_test, y_test)

bagg_score_train = regression.score(X_train, y_train)

print("MAE: %f" % (mae))

print("RMSE: %f" % (rmse))

print("MSE: {:.4f}".format(mse))

print("Score Train: %f" % (bagg_score_train))

print("Score Test: %f" % (bagg_score_test))

import matplotlib.pyplot as plt

x_ax = range(len(y_test[300:500]))

plt.figure(figsize=(24,10))

plt.plot(x_ax, y_test[300:500], lw=1.5, color="orange", label="Potencia Original")
```



```
plt.plot(x_ax,  bagg_predict[300:500],  lw=1.5,  color="green",  label="Potencia
Predecida")

plt.legend()

plt.show()

"""## XGBoost"""

import xgboost as xgb

regresion  =  xgb.XGBRegressor(n_estimators=500,  objective="reg:squarederror",
max_depth=5)

regresion.fit(X_train,y_train)

bagg_predict = regresion.predict(X_test)

from  sklearn.metrics  import  mean_squared_error,  mean_absolute_error,
mean_absolute_percentage_error

import numpy as np

mse = mean_squared_error(y_test, bagg_predict)

rmse = np.sqrt(mean_squared_error(y_test, bagg_predict))

mae = mean_absolute_error(y_test, bagg_predict)

bagg_score_test = regresion.score(X_test, y_test)

bagg_score_train = regresion.score(X_train, y_train)

print("MAE: %f" % (mae))

print("RMSE: %f" % (rmse))
```



```
print("MSE: {:.4f}".format(mse))

print("Score Train: %f" % (bagg_score_train))

print("Score Test: %f" % (bagg_score_test))

import matplotlib.pyplot as plt

x_ax = range(len(y_test[300:500]))

plt.figure(figsize=(24,10))

plt.plot(x_ax, y_test[300:500], lw=1.5, color="orange", label="Potencia Original")

plt.plot(x_ax, bagg_predict[300:500], lw=1.5, color="green", label="Potencia
Predecida")

plt.legend()

plt.show()

xgb.to_graphviz(regresion)

format = 'png'

image = xgb.to_graphviz(regresion)

image.graph_attr = {'dpi':'400'}

image.render('XGboost', format = format)
```

ANEXO 4 – FRAGMENTO DEL ÁRBOL DE DECISIÓN DEL MODELO

XGBOOST

