



**Prototipo de un módulo de análisis de datos para plantas de generación fotovoltaica
mediante técnicas de aprendizaje automático en el contexto del sistema SCADA Axon
Builder**

Juan Felipe Urbano Ordoñez

Informe de práctica para optar al título de: Ingeniero Electrónico

Asesores

Amado De Jesús Tavera Crespo, Doctor (PhD)

Iván Darío Claros Gómez, Doctor (PhD)

Universidad de Antioquia
Facultad de Ingeniería
Ingeniería Electrónica
Medellín, Antioquia, Colombia
2021

Cita	(Urbano Ordoñez, 2021)
Referencia	Urbano Ordoñez. J. (2021). <i>Prototipo de un módulo de análisis de datos para plantas de generación fotovoltaica mediante técnicas de aprendizaje automático en el contexto del sistema SCADA Axon Builder</i> [Trabajo de grado profesional]. Universidad de Antioquia, Medellín, Colombia.
Estilo APA 7 (2020)	



Centro de documentación de Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/director: Jesús Francisco Vargas.

Jefe departamento: Cesar Augusto Salazar.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Resumen

El aumento de la demanda de energía limpia y renovable ha provocado un auge exponencial en la búsqueda e implementación de sistemas de generación de energía fotovoltaica. Por su parte, Axon Group es una empresa dedicada a la distribución de productos y servicios para la automatización del sector eléctrico, entre ellos el SCADA Axon Builder (AB), el cual es un producto para la supervisión control y adquisición de datos (SCADA/IHM), el propósito de este trabajo fue en primera medida realizar un acercamiento al control de sistemas fotovoltaicos para su posible integración con el AB, por medio del diseño de un prototipo funcional que aplique técnicas de predicción en el contexto de la generación de energía en sistemas fotovoltaicos, lo cual se desarrolló sobre .NET y fue implementado en el lenguaje c#. Para llegar al prototipo final se ejecutaron diferentes etapas entre ellas: la exploración de los sistemas fotovoltaicos, la definición de variables de interés, el prototipado de modelos en Python y la apropiación de librerías e implementación bajo entorno .NET, entre otras.

Índice

	RESUMEN	i
	ÍNDICE	ii
	ÍNDICE DE FIGURAS	iii
1.	INTRODUCCIÓN	1
2.	OBJETIVOS	3
	2.1 Objetivo general	3
	2.2 Objetivos específicos	3
3.	MARCO TEÓRICO	4
4.	METODOLOGÍA	8
	4.1 Investigación	8
	4.1.1 Contexto plantas fotovoltaicas	8
	4.1.2 Sistemas de control para plantas fotovoltaicas	8
	4.1.3 Modelos de predicción aplicados	8
	4.2 Prototipado	8
	4.2.1 Implementación de modelos en Python	8
	4.2.2 Implementación de modelos en C#	9
	4.3.3 Diseño y construcción prototipo con interfaz gráfica en C#	9
5.	PROCEDIMIENTO Y RESULTADOS	10
	5.1 Estado del arte – Modelos de predicción en sistemas fotovoltaicos y selección de variables de interés	10
	5.2 Prototipado	13
	5.2.1 Python	17
	5.2.1.1 Modelo ARIMA	17
	5.2.1.2 Modelo redes neuronales	19
	5.2.2 C#	21
	5.2.2.1 Integración con interfaz gráfica de los modelos	21
6.	CONCLUSIONES	25
7.	REFERENCIAS	27

Índice de figuras

Figura 1 Esquema de una neurona artificial.	6
Figura 2 Red de 4 capas de neuronas artificiales.	7
Figura 3 Muestra de datos (potencia e irradiancia por días)	14
Figura 4 Muestra de datos filtrados (potencia e irradiancia)	15
Figura 5 Matriz de correlación para el Dataset	16
Figura 6 Resultado de la autocorrelación de la serie de datos con frecuencia de 60 min.	17
Figura 7 Resultado de la función auto-arima para determinar el orden del modelo que se usará .	18
Figura 8 Resultado de las predicciones realizadas con el modelo	18
Figura 9 Error absoluto y cuadrático medio para las diferentes series de datos según su frecuencia de submuestreo.....	19
Figura 10 Arquitectura ANN para predicción de potencia en sistemas solares seleccionada	20
Figura 11 Predicción de potencia con ANN	20
Figura 12 Error cuadrático medio en función de las iteraciones del modelo.....	21
Figura 13 Interfaz prototipo funcional implementado en C#	21
Figura 14 Gráfica de variables en el área de visualización.....	22
Figura 15 Lista de Checkbox para selección de variables de interés por el usuario	22
Figura 16 Variable seleccionada por el usuario mediante Checkbox.....	22
Figura 17 Submenú, elección de técnica de entrenamiento.....	23
Figura 18 Selección de parámetros para entrenamiento o carga de modelo.....	23
Figura 19 Predicción de potencia graficada en el área de visualización de datos mediante una ANN.....	24

1. Introducción

La demanda de electricidad a nivel mundial tiene un aumento proporcional al crecimiento poblacional e industrial, a tal punto que el acceso a la misma se considera parte del índice del desarrollo humano dentro de los países. Sin embargo, algunas de las fuentes de energía convencionales generan una cantidad alarmante de contaminación, por lo cual la búsqueda de nuevas fuentes de generación de energía se ha fortalecido. Entre estas alternativas la energía solar se ha constituido como una de las más importantes por su gran aprovechamiento en cualquier lugar del planeta. Este tipo de energía utiliza el principio fotoeléctrico para convertir los fotones provenientes de la luz solar en energía eléctrica.

Colombia se cataloga como un entorno ideal para el desarrollo de proyectos basados en este tipo de energía gracias a su ubicación geográfica con condiciones favorables de radiación solar. Asimismo, el desarrollo de nuevas tecnologías y algunos beneficios tributarios han permitido el auge de estos proyectos. Según las estadísticas de la Unidad de Planeación Minero-Energética (UPME), el 88,3% de las iniciativas radicadas tienen que ver con energía solar, en donde 9 de cada 10 propuestas para generar energía usarán paneles solares, lo que representa un crecimiento del 32% de julio a noviembre de 2017. Es factible que se mantenga la tendencia una vez entre la reglamentación de la Comisión de Regulación de Energía y Gas (CREG) expedida bajo la Resolución 137 de 2020, para el desarrollo de energías no convencionales. La UPME y el Ministerio de Minas y Energía estiman que para antes de 2030 cerca de 10% del consumo energético en Colombia va a provenir de proyectos fotovoltaicos o solares.

Dentro de las plantas de generación fotovoltaicas se encuentran diferentes componentes, entre los que se destacan: inversores, reguladores y celdas solares. Además de su comportamiento intrínseco, la potencia y eficiencia de producción en estos sistemas dependen considerablemente de variables atmosféricas, pues estas juegan un papel muy importante en el desempeño de las celdas solares. En primera medida por la posición del sol y su irradiancia, y además porque la temperatura de la celda afecta la eficiencia de la producción de cada panel. Estas variables ambientales son medidas por las estaciones meteorológicas que comúnmente se instalan junto con los sistemas de generación. Dichas variables, sumadas a las convencionales en los sistemas de generación de energía tales como voltajes, corrientes, frecuencia, entre otras, son monitorizadas y controladas por los sistemas SCADA y constituyen una potencial fuente de información para las operaciones de mantenimiento y control de la producción en un punto óptimo de eficiencia.

Por su parte, Axon Group es una empresa de origen colombiano dedicada a la automatización del sector eléctrico desde hace 12 años. Entre sus líneas de negocios se encuentra el desarrollo de productos de software tales como: Axon Builder (AB), el cual es un software para la supervisión, control y adquisición de datos (SCADA / IHM). Asimismo, Axon Group brinda soluciones para estos procesos de automatización y control en el entorno de las subestaciones y las plantas generadoras de energía. Actualmente, el AB no cuenta con soporte especializado en plantas de generación fotovoltaica, lo cual limita su aplicación en este tipo de contextos. En este marco, se pretende realizar un estudio de viabilidad, así como el desarrollo de un prototipo de un módulo especializado en la visualización y análisis de datos en el contexto de plantas fotovoltaicas. Particularmente, se propone facilitar la aplicación de técnicas de aprendizaje automático sobre los datos, para permitir a los usuarios del SCADA usar modelos de predicción e identificar patrones que puedan ser utilizados para generar proyecciones de generación de energía, teniendo en cuenta los factores ambientales en el entorno del sistema de generación.

En la literatura académica, existe una gran variedad de modelos y técnicas de predicción que podrían ser aplicados a la predicción de potencia dentro de una planta fotovoltaica. Por ello es necesario realizar una primera aproximación de algunas de estas opciones, pensando en la compatibilidad y eficiencia con el modelo actual del sistema AB. El prototipo funcional propuesto funcionará mostrando información gráfica de las variables del sistema, además tendrá la capacidad de generar predicciones de potencia a partir de los modelos seleccionados.

2. Objetivos

2.1 General

Desarrollar un módulo de software especializado, empleando técnicas de predicción de datos y los lenguajes de programación Python y C#, para la visualización y análisis de datos de plantas fotovoltaicas, además de su integración al sistema SCADA Axon Builder

2.2 Específicos

- Identificar el conjunto de variables de interés dentro del proceso de supervisión en sistemas solares por medio de la exploración de sus principales componentes, además de su debida documentación.
- Seleccionar técnicas de análisis de datos que se ajusten a las variables identificadas, con las cuales se puedan predecir datos, generando valor al SCADA Axon Builder en los procesos de gestión de la energía solar.
- Diseño e implementación del módulo teniendo en cuenta las técnicas seleccionadas.

3. Marco Teórico

Se consideran como energías renovables a todas aquellas que se adquieren a partir de fuentes naturales en las que el recurso sea inagotable, ya sea por la estabilidad de la fuente o por la capacidad para regenerar el insumo de manera natural. En esta categoría se destacan: eólica, geotérmica, hidroeléctrica, solar, mareomotriz, biomasa y biocombustibles [1]. Particularmente, la energía solar fotovoltaica se produce gracias al efecto fotoeléctrico, el cual consiste en la conversión de la luz en electricidad. Este proceso se consigue con algunos materiales que tienen la propiedad de absorber fotones y emitir electrones. Cuando estos electrones libres son capturados, el resultado es una corriente eléctrica que puede ser utilizada como electricidad [2].

El principio fotoeléctrico es aprovechado por los paneles solares, los cuales están organizados en strings, sub-arrays, arrays, que, dependiendo del número y la conexión de estos, reciben estos nombres dentro de un parque solar. Dentro de estos parques son utilizados diferentes dispositivos, tal como el regulador de carga, los inversores, las estaciones meteorológicas, los transformadores y demás elementos convencionales de un sistema de generación eléctrica. Dentro de estos elementos se destaca el PPC (Power Plant Controller) el cual es un controlador basado en un PLC [3], que se encarga de regular la potencia activa y reactiva del parque solar en función de los requerimientos presentados por la distribuidora u operadora de red [4], particularmente en Colombia el factor de potencia debe ser siempre mayor de 0.9, pues las empresas con un valor menor al mencionado sufrirán penalidades.

El PPC se integra con un HMI (interfaz Hombre - Máquina) [5], encargada de ser la unión entre el proceso controlado y los operarios. Su función es mostrar información en tiempo real y facilitar operaciones de control y supervisión (SCADA o Supervisory Control And Data Acquisition) [6]. Los sistemas SCADA se requieren para supervisar y controlar remotamente una instalación de algún proceso, pudiendo integrar datos recogidos desde diferentes sensores, o dispositivos (PLCs), mediante diferentes protocolos de comunicación.

A partir de los datos adquiridos por el sistema SCADA, y de los datos que las estaciones meteorológicas presentes en las instalaciones fotovoltaicas generan, las técnicas de análisis de datos se constituyen como un elemento del cual se puede sacar provecho para obtener múltiples informes detallados, alertas de eventos o, particularmente en los sistemas solares, predicción de producción de energía. Dentro de estas técnicas de análisis de datos se pueden encontrar diferentes metodologías que están siendo aplicadas en la actualidad, tales como redes neuronales, clustering,

algoritmos genéticos, entre otras [7]. Junto con estas, la visualización de datos puede llegar a ser una herramienta muy valiosa, sacando provecho de los datos generados y obtenidos dentro del proceso de generación de energía.

Particularmente, se describe el modelo ARIMA [8] como un modelo autorregresivo integrado de promedio móvil. Es un modelo estadístico que, a partir de variaciones y regresiones de una serie de datos, encuentra patrones para una posible predicción en vista hacia el futuro, las predicciones están basadas en datos pasados dentro de una serie de tiempo y no en variables independientes como es costumbre en las técnicas de predicción.

El modelo contará con 3 parámetros (p, d, q) que son números enteros no negativos que hacen referencia a orden de cada componente del modelo, respectivamente serían la componente autorregresiva (AR), integrada(I) y de media móvil (MA), dentro de este modelo es posible considerar procesos con estacionalidad, pasando a ser SARIMA (seasonal autoregressive integrated moving average).

Un modelo ARIMA toma la expresión de la ecuación 1:

Ecuación 11 Ecuación general de un modelo ARIMA

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)(1 - B)^d X_t = (1 - v_1 B - v_2 B^2 - \dots - v_q B^q) a_t$$

Donde la serie de tiempo puede ser convertida en estacionaria mediante la diferenciación de orden d, además se muestra como el proceso autorregresivo (AR) tiene los primeros p términos de la función de autocorrelación parcial distintos de cero y los demás son nulos. Por último, también se tiene en cuenta que un proceso de medias móviles (MA), los primeros q términos de la función de autocorrelación distintos de cero y los demás son nulos.

Si se habla de redes neuronales (ANN – Artificial Neural Network)[9], se describen como algoritmos que están diseñados para imitar el comportamiento del cerebro humano. En palabras simples, el propósito de estas redes generalmente se reduce a 3 aspectos: aprender de la experiencia, obtener información de las entradas provistas y generalizar lo aprendido en nuevos elementos. Estos aspectos son los que permiten reconocer patrones dentro de un proceso, lo cual es muy valioso dentro de las técnicas de predicción.

Como se puede observar en la figura 1, correspondiente al esquema de una neurona artificial, se muestran algunos componentes importantes. Para su funcionamiento, la suma de las entradas multiplicadas por sus pesos asociados determinará el “impulso nervioso” que recibe la neurona, este valor es procesado al interior de la neurona por medio de una función de activación que devuelve un valor que se envía como salida de la neurona.

La función de activación devuelve una salida que será generada por la neurona dada una entrada o conjunto de entradas. Cada una de las capas que conforman la red neuronal tienen una función de activación que permitirá reconstruir o predecir. Ejemplos de estas funciones son: relu, tangente hiperbólica, lineal, entre otras.

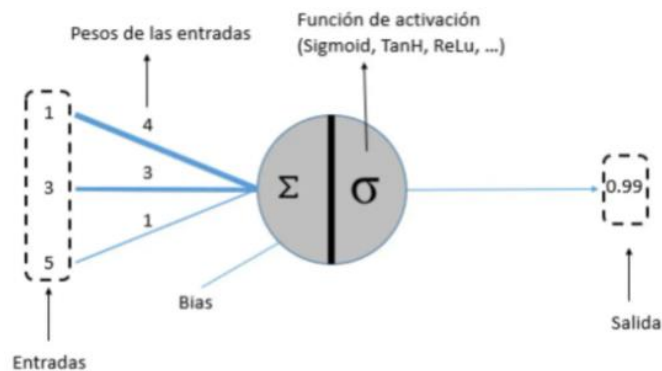


Figura 1 Esquema de una neurona artificial. Fuente: <https://www.xeridia.com/>

Así como el cerebro humano está compuesto por neuronas interconectadas entre sí, una red neuronal se conforma por varias neuronas conectadas entre sí y agrupadas en diferentes niveles denominados capas. En la figura 2 se tiene una red neuronal compuesta por 9 neuronas distribuidas en 4 capas. La primera capa a la izquierda, conocida como la capa de entrada es la encargada de recibir los datos reales, y la última capa de la derecha, es decir la capa de salida es la encargada de proporcionar el resultado visible de la red. Las capas intermedias son conocidas como capas ocultas ya que se desconocen los valores de entrada como de salida.

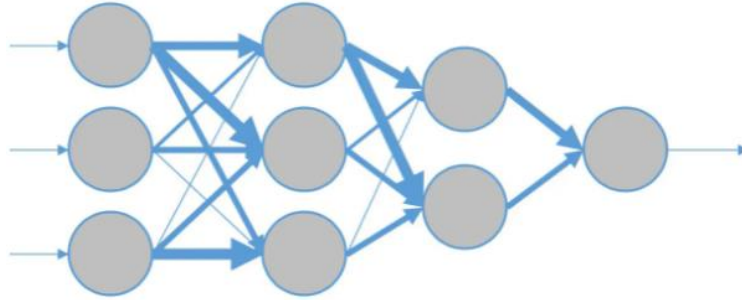


Figura 2 Red de 4 capas de neuronas artificiales. Fuente: <https://www.xeridia.com/>

El proceso de entrenamiento de la red neuronal consiste en ajustar cada uno de los pesos de las entradas de cada una de las neuronas, para que las respuestas de la capa de salida se ajusten en mejor medida a los datos que ya se conocen; por esto, es tan importante contar con datos reales y confiables, pues ellos serán los encargados de transmitirle el conocimiento a la red para que pueda predecir más adelante con entradas diferentes a los datos de entrada.

Los datos generados a través de la predicción con los modelos son comparados con los valores de testeo con lo que generalmente se cuenta en un conjunto de datos, esta comparación se realiza con el RMSE (Root Mean Squared Error) [10] como se muestra en la ecuación 2:

Ecuación 2 RMSE. Ecuación general

$$RMSE = \sqrt{(predicted - observed)^2}$$

La cual hace referencia a la raíz del error cuadrático medio, por su traducción del inglés, y está definida como la medida de desempeño cuantitativa utilizada comúnmente para evaluar métodos de pronóstico, midiendo la cantidad de error que hay entre dos conjuntos de datos.

4. Metodología

Enmarcado en una metodología Agile SCRUM adoptada por el área de R & D en Axon Group, fueron establecidas las tareas necesarias para desarrollar el objetivo principal del proyecto, las cuales pueden ser diferenciadas en dos partes, la investigativa y la de prototipado.

4.1 Investigación

4.1.1. Para entrar en el contexto de los sistemas fotovoltaicos, puntualmente en la predicción de energía y su relación con las variables ambientales, se realizó un estudio previo centrado en la arquitectura física y en los componentes específicos de estos sistemas de generación energética, resaltando la importancia de los inversores y de los controladores de carga dentro del proceso de generación.

4.1.2. La información previa es complementada con el estudio sobre el funcionamiento de los sistemas de control dentro del área de las plantas fotovoltaicas, además de la exploración de los sistemas SCADA disponibles en la industria. El objetivo es tener una base clara de lo que se debe plantear, aterrizando las ideas de los posibles prototipos.

4.1.3. Una vez comprendido el contexto de los sistemas de generación fotovoltaica en un amplio sentido, se procedió a realizar un estado del arte sobre los modelos de predicción que actualmente son estudiados y usados, en el ámbito de los sistemas fotovoltaicos, para la predicción de potencia. A partir de esta información se establecieron dos partes fundamentales para el proceder del proyecto:

- Las técnicas que se seleccionaron para su modelamiento, teniendo en cuenta la precisión y la dinamicidad de estas para modificarse según se requieran sin afectar considerablemente el resultado.
- Las variables críticas que son necesarias para el desarrollo de los modelos, y también para realizar una visualización de estas en el tiempo dentro del módulo.

4.2 Prototipado

4.2.1. Teniendo en cuenta todos los conceptos previamente estudiados y con el fin de realizar la comprobación del entendimiento y manejo de estos, se hizo la implementación de dos de los modelos en el lenguaje Python. Para tal fin, fue necesario usar un Dataset que posteriormente será descrito dentro de este trabajo. Una vez implementados los modelos en Python, es necesario evaluarlos cambiando algunas de sus características.

4.2.2. El software Axon Builder (AB) está construido sobre .NET e implementado en el lenguaje c#, en consecuencia, para poder realizar una aproximación más real de los modelos y así producir la incrustación dentro del AB, se buscó la manera más conveniente de ejecutar este procedimiento, estableciendo Keras.NET como la herramienta con la cual se desarrollaría un modelo de los anteriores en C#.

4.2.3. Posterior a la primera integración de los modelos y de las técnicas de predicción de datos con .NET y con C#, se realizó un prototipo funcional implementado sobre una interfaz gráfica, donde el usuario puede cargar un Dataset y con este mismo realizar una predicción de potencia mostrada el área de visualización de datos.

5. Procedimiento y resultados

5.1 Estado del arte: técnicas de predicción en sistemas fotovoltaicos y selección de variables de interés

Con las bases sobre el funcionamiento de los sistemas fotovoltaicos claras, en lo que corresponde a sus conceptos y los componentes que hacen posible el proceso de generación de energía, se encuentra que la potencia generada es el foco de atención dentro de este proceso, pues la planta de generación debe cumplir con una cuota diaria de producción, para poder estar conectada a la red eléctrica nacional y por consiguiente vender su energía producida. Actualmente existen muchos trabajos e investigaciones en curso, cuyo objetivo principal es generar modelos o técnicas de predicción que sean capaces de predecir la potencia a generar a partir de variables del entorno del sistema.

Para este objetivo, los trabajos establecen horizontes de predicción, los cuales son periodos de tiempo que determinan el momento futuro hasta el cual se pretende realizar la predicción, estos varían entre 4-9 horas, 48-72 horas y 3 a 7 días. Sin embargo, gran parte de trabajos consultados y documentados centran su interés en un horizonte predictivo de 1 día, con el fin de generar la predicción para el día siguiente de trabajo del sistema.

A continuación, se documentan algunos de los trabajos más relevantes en el contexto anteriormente descrito:

- Wang [13] desarrolla un modelo de predicción de la radiación media horaria con un horizonte de 24 horas basado en una red neuronal, siendo destacable que los autores realizan un preprocesamiento de los datos exhaustivo, calculando diferentes parámetros y normalizando variables con el fin de aumentar la eficiencia de la red neuronal. Una de las tareas que realizan es normalizar el valor de la radiación que van a utilizar como entrada. A su vez, incluyen el número de horas de sol, y dos parámetros nuevos (TODmax y NDD) calculados a partir de la diferencia entre la radiación extraterrestre (calculada) y la de superficie (medida el día previo a la predicción). Concretamente TODmax es el valor máximo de la tercera derivada de la diferencia entre ambas radiaciones (de forma que las variaciones bruscas de radiación vienen denotadas por un valor alto de esta variable) y NDD es la raíz de la diferencia cuadrática entre ambas radiaciones normalizadas (útil para clasificar los días en nublados o claros). Las entradas de la red neuronal son la radiación media diaria y la temperatura media diaria para el día previo a la predicción (medidas

localmente), además de los parámetros TODMAX y NDD de dicho día y el día del año al que corresponde la predicción.

- En [14], Chow desarrolla un modelo de predicción para aplicarlo a muy corto plazo (10 y 20 min) y a tiempo real. Dicho modelo predice la potencia media generada para los intervalos comentados en una instalación de 40 kW compuesta por paneles fijos sobre un tejado. La serie temporal que utiliza está compuesta por los datos históricos de 15 días. La red neuronal, que concretamente se trata de una MLP, utiliza un vector de entradas compuesto por el azimut, la altura solar y los valores de temperatura y radiación medidos localmente. El factor de correlación entre la curva de la potencia media generada real y la prevista oscila de 0,81 a 0,98 entre los diferentes días utilizados.
- Mellit [15] presenta un modelo que estima la potencia media generada a tiempo real, así como la potencia media diaria que va a ser generada por un módulo fotovoltaico en dicho día. Se desarrollan dos redes neuronales diferentes aplicando una u otra según sea considerado el día de estudio nublado o soleado. Para clasificar el día utilizan la radiación media diaria del último día, de forma que si es inferior a 400 W/m² se considera nublado y en caso contrario soleado. La serie temporal utilizada para desarrollar el modelo comprende un período de 13 meses. La red neuronal utiliza como entradas la temperatura y la radiación horizontal medidas localmente. La relación entre la potencia estimada y la real tiene un factor de correlación que va del 0,96 al 0,97.
- En [16], Li utiliza un modelo ARMAX para predecir la potencia media diaria que producirá al día siguiente un parque solar fotovoltaico de 2,1 kW de potencia compuesto por paneles fijos. Para ello utilizan como entrada del modelo la radiación horizontal global medida localmente 24 horas antes de la predicción. Para entrenar el modelo se ha utilizado un histórico de datos, de aproximadamente un año y medio, que contiene tanto la radiación horizontal(entrada) como el valor real de la potencia diaria del día siguiente.
- En [De Giorgi 23] se predice la potencia eléctrica media horaria para diversos horizontes hasta 24 horas. Se utilizan datos de potencia horaria recolectados de un sistema fotovoltaico con una potencia nominal total de 960 kWp, instalado en Italia. Medidas horarias de la temperatura del módulo, la temperatura ambiente, la irradiancia en el plano inclinado y su ángulo de inclinación. Se describen tres modelos de predicción: el LS-SVM (Least Square Support Vector Machine), una red neuronal conocida como GMDH (Group Method of Data

Handling), y un nuevo algoritmo híbrido GLSSVM (Group Least Square Support Vector Machine) que se basa en la combinación de los dos primeros modelos. Cada modelo se combina con tres estrategias (Directa, Recursiva y Directa-Recursiva) para cada horizonte de predicción. con el fin de evaluar el rendimiento de los diferentes métodos de predicción se utiliza el error absoluto medio normalizado (NMAE), el error medio normalizado (NME) y la raíz del error cuadrático medio (NRMSE). La salida demuestra que el método GLSSVM con la estrategia Directa-Recursiva permite obtener un error normalizado menor.

Una vez realizado el estado del arte se eligen dos de ellas que puedan ser llevadas a cabo en el contexto de la integración con el Axon Builder (AB): redes neuronales y los modelos de autorregresión. En una primera etapa esta implementación se realizó en Python, pues como es bien sabido, es un lenguaje en el que la ciencia de datos y la inteligencia artificial tiene mayor facilidad de desarrollo.

Además de la selección de técnicas se identificaron las principales variables de interés para el desarrollo de estos modelos. La selección de variables de interés en cualquier sistema de control es una parte fundamental del desarrollo de la solución para el problema planteado, generalmente se cuenta con un conjunto de variables que son candidatas para aportar información al modelo con el que se desea trabajar. La importancia de esta parte radica en el valor de la información que la variable seleccionada pueda aportar y con ello realizar un modelo óptimo. Las variables seleccionadas han sido:

- **Irradiancia:** es el flujo de energía emitida por el sol en forma de ondas electromagnéticas, siendo estas ondas de diferentes tamaños; el sol emite estas radiaciones electromagnéticas a una temperatura de 325 grados centígrados, aproximadamente. La irradiancia emitida por el sol se distribuye desde el infrarrojo hasta el UV. Solo una parte de esta radiación emitida llega a la tierra, debido a la existencia de las capas atmosféricas especialmente la capa de ozono, la parte de la radiación solar que llega a la tierra es conocida como la irradiancia. Cerca de un 80% de los autores del medio tienen en cuenta esta variable, considerando la más importante en el proceso de predicción, pues su relación con la potencia generada es muy estrecha. Además, los valores medios de esta magnitud pueden brindar información importante sobre las zonas y regiones de Colombia que sean aptas para establecer los sistemas fotovoltaicos. Un ejemplo claro de estos análisis es la baja favorabilidad con la

que cuenta la costa pacífica colombiana para albergar este tipo de sistemas, pues es uno de los lugares con menor radiación solar al ser una zona tan húmeda. Esta variable es medida en W/m².

- Temperatura del panel: puede afectar directamente la eficiencia del panel pues estos no están pensados para trabajar bajo condiciones extremas; además, una temperatura muy elevada podría dar indicios de algunas fallas directamente enfocadas en el panel, como algún cortocircuito.
- Temperatura ambiente: al igual que la anterior variable brinda información sobre la eficiencia que pueda llegar a tener un panel dependiendo del entorno, el punto de trabajo óptimo es de los 24 grados centígrados.
- Voltaje: es una medida importante para el sistema, pues para el buen funcionamiento de éste, se debe establecer el voltaje requerido. Corresponde al potencial eléctrico que puede ser aprovechado por los dispositivos conectados al sistema autónomo, o el potencial que se entregará a la red eléctrica.
- Corriente: al igual que el voltaje asegura el correcto funcionamiento del sistema, en conjunto con el voltaje y los inversores.
- Potencia: definida como la cantidad de trabajo efectuado por unidad de tiempo, la potencia es el requerimiento más importante para entregar a través del sistema fotovoltaico. La potencia puede ser medida en diferentes puntos del sistema y aporta información en cada uno de estos puntos, desde medirla en los inversores, para comprobar el correcto funcionamiento de las cadenas de paneles, hasta medirla en la salida del sistema para asegurar la demanda de potencia que se necesita.
- Tiempo: las marcas temporales usadas en los Datasets son de gran ayuda en el momento de realizar una representación gráfica de todas las variables mencionadas anteriormente.

5.2 Prototipado

Después de estudiar las diferentes técnicas presentes en la literatura de investigación, se eligen dos de ellas (ARIMA, ANN), con el fin de generar modelos funcionales de cada una de ellas usando como herramienta de desarrollo el entorno Jupyter, por medio del lenguaje Python. Para cualquier implementación de un modelo que use inteligencia artificial, se hace necesario contar con un Dataset apropiado para tal fin; en el desarrollo de este trabajo, esta fue una de las limitaciones más

grandes para poder desarrollar modelos con un análisis profundo. Los Datasets deben contar con valores de las variables en un amplio rango de condiciones, específicamente para un trabajo de predicción de potencia en una planta de generación fotovoltaica, es necesario contar con valores a lo largo de todo el año, pues el entrenamiento del modelo debe contar con la mayor cantidad de valores posibles con todas las condiciones que se puedan presentar en el entorno.

Para este trabajo el Dataset usado, fue el facilitado por MEng. Oscar Ocampo, Ingeniero de la Universidad de Antioquia, cuyo trabajo de maestría centro su estudio en el “Monitoreo de la eficiencia de paneles fotovoltaicos bajo condiciones de lluvia y polvo” [11], donde se realizó un montaje correspondiente para enviar datos tomados de un panel solar a la nube por medio de un Arduino y una SIMCARD. La limitación anteriormente mencionada se encuentra en dos factores principales: reducido tamaño de datos, mediciones con sensores electrónicos sobre Arduino. Este conjunto de datos obtenido desde un montaje experimental se encuentra presente en [12] y pertenece al grupo de investigación TESLA de la Universidad de Antioquia; en él, se miden y recolectan algunas de las variables importantes de este proceso, entre ellas: Fecha [Y-M-D], Hora [h:m: s], Irradiancia [W/m²], Temperatura del panel [°C], Temperatura ambiente [°C], Humedad [%], Voltaje [V], Corriente [A], Potencia [W], Consumo [KWh], Eficiencia [%]. En la figura 3 se grafica una muestra del conjunto de datos proporcionado, donde se muestra la potencia generada y además la irradiancia correspondiente a esos días de generación.

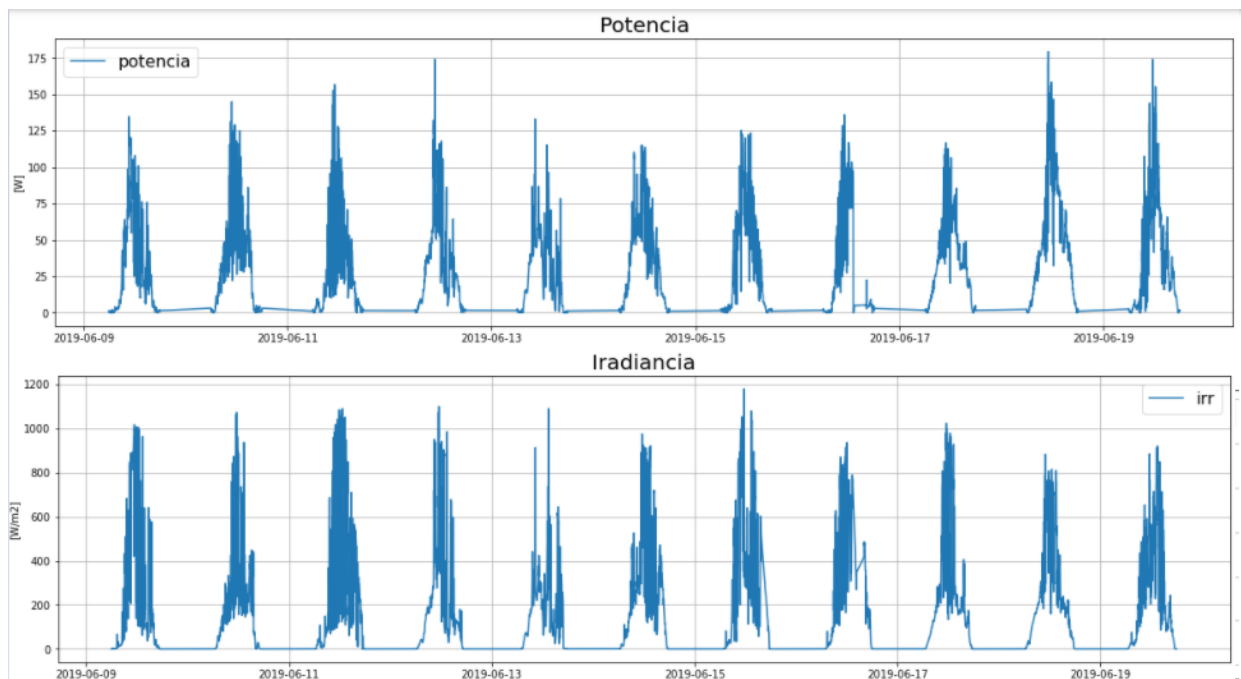


Figura 3 Muestra de datos (potencia e irradiancia por días)

Para las variables del conjunto de datos, la frecuencia de medición es de aproximadamente un minuto por muestra y se consigna cada una de estas variables mencionadas en un archivo .xlsx. Una vez obtenido el conjunto de datos, es común realizar un preprocesamiento de los datos, con el fin de ajustar los datos de la manera más conveniente para la solución del problema planteado. Para ello, una vez cargados los datos en el entorno de trabajo, se realizan diferentes DataFrames con algunas modificaciones realizadas a los datos, entre estas modificaciones se puede encontrar: normalización y escalamiento de datos, filtrado de datos por medio de un filtro de media móvil, submuestreo de datos, entre otros.

Como se observa en la figura 4 y a diferencia de la primera muestra graficada, se nota claramente como la limpieza de los datos generada por el filtro de media móvil muestra con mayor claridad el comportamiento de las variables, pues en el proceso de medición del sistema, las variaciones aportan mucho ruido indeseable, además se realizó un escalamiento de los datos para trabajar con ellos con mayor facilidad.

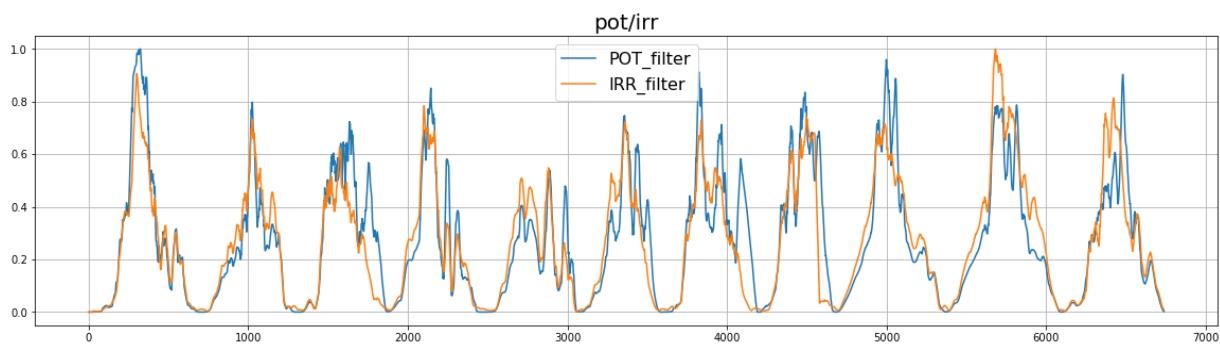


Figura 4 Muestra de datos filtrados (potencia e irradiancia)

Teniendo en cuenta algunos datos del conjunto y como parte del preprocesamiento y análisis primario de los datos se muestra en la figura 5 la matriz de correlación de los datos más sobresalientes, donde como era de esperarse la irradiancia marca una clara correlación con la potencia generada, y a su vez con las temperaturas medidas.

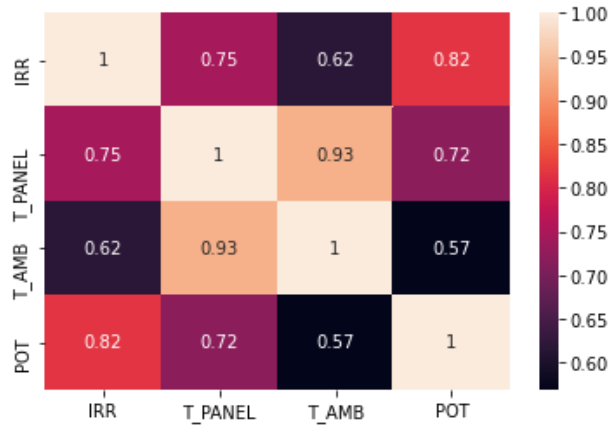


Figura 5 Matriz de correlación para el Dataset

Como es común en el desarrollo de modelos de predicción, el conjunto de datos es dividido en un porcentaje de 80% y 20% entre subconjuntos de entrenamiento y de prueba respectivamente, los datos de entrenamiento, como se mencionó anteriormente, son los encargados de proporcionarle los patrones característicos del proceso al modelo, para que a partir de ellos pueda generar predicciones y posteriormente compararlas con los datos de prueba. Dentro de esta comparación y para los modelos de regresión se hace necesario obtener algunas métricas de evaluación de los modelos, entre las cuales se encuentran:

- Error medio absoluto: es la media de las diferencias absolutas entre el valor objetivo y el predicho. Al no elevar al cuadrado, no penaliza los errores grandes, lo que la hace no muy sensible a valores anómalos, por lo que no es una métrica recomendable en modelos en los que se deba prestar atención a éstos.
- Error cuadrático medio: una de las medidas más utilizadas en tareas de regresión. Es simplemente la media de las diferencias entre el valor objetivo y el predicho al cuadrado. Al elevar al cuadrado los errores, magnifica los errores grandes, por lo que hay que utilizarla con cuidado cuando tenemos valores anómalos en nuestro conjunto de datos.

Como se mencionaba anteriormente, para los modelos de regresión, y a diferencia de los modelos de clasificación, es casi imposible predecir el valor exacto, entonces se busca estar lo más cerca posible del valor real.

5.2.1 Prototipado en Python

5.2.1.1 Modelo ARIMA

Antes de empezar con la creación del modelo ARIMA, en primera medida se hace necesario analizar la naturaleza de la serie de datos, donde como es evidente y se muestra en la figura 1 y 2, se observa que hay tendencias repetitivas cada día, lo cual es previsto por la irradiancia solar a lo largo del mismo, esta repetición de las tendencias marca cierta estacionalidad de los datos, por lo que el modelo a generar pasará de ARIMA a SARIMA.

Una vez hecho este primer análisis, en lo que procede a encontrar el mejor modelo, es decir, el que genere una predicción más precisa, se establecen diferentes tipos de datos para entrenamiento y prueba de este modelo, para este caso se ha re-muestreado el Dataset principal con distintas frecuencias generando nuevas series de datos, con cada una de ellas se creará un modelo diferente para comparar sus resultados.

A Continuación, se describen los pasos para cada una de las series mencionadas. Particularmente, a detalle para la serie que mejor desempeño tuvo, es decir la que se submuestreo a una frecuencia de una hora (60 min).

- 1) Se emplea la función de autocorrelación en la serie de datos, la cual es una herramienta estadística que, como su nombre lo indica, se basa en la medida de la autocorrelación entre las observaciones de una serie de tiempo con el fin de encontrar patrones repetitivos dentro de una señal; es útil para encontrar esta periodicidad de la señal enmascarada bajo algún tipo de ruido, así como también para ubicar su frecuencia fundamental. En la figura 6 se observan dos gráficas arrojadas por esta función, donde claramente y de acuerdo con lo mencionado anteriormente, se evidencia la frecuencia principal de 24, recordando que se está trabajando con la serie de datos re-muestreada por hora, lo que nos indica una repetición del patrón por cada día.

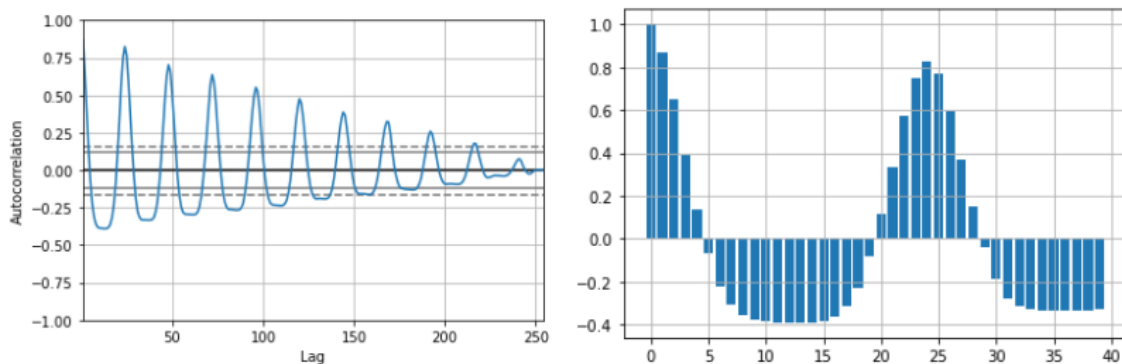


Figura 6 Resultado de la autocorrelación de la serie de datos con frecuencia de 60 min.

2) En este punto, es conveniente realizar la prueba de Dickey-Fuller [22] aumentada con el fin de confirmar la estacionariedad de la serie de tiempo, esta prueba estadística arroja un valor p el cual debe ser menor que 0.05 ($p < 0.05$) si la serie es estacionaria, para el caso de la serie de datos con la que se está trabajando se obtuvo un valor $p = 8.456 e-17$. En este paso también se utiliza la función auto ARIMA, la cual arroja el orden que mejor se adapta a los datos para el modelo ARIMA, este orden es independiente de la estacionariedad mencionada anteriormente. En la figura 7, se muestran algunas iteraciones, además del resultado final para la serie de datos usada:

```

Performing stepwise search to minimize aic
ARIMA(2,0,2)(0,0,0)[0] intercept : AIC=1964.821, Time=1.57 sec
ARIMA(0,0,0)(0,0,0)[0] intercept : AIC=2408.284, Time=0.01 sec
ARIMA(1,0,0)(0,0,0)[0] intercept : AIC=2049.194, Time=0.05 sec
ARIMA(0,0,1)(0,0,0)[0] intercept : AIC=2183.236, Time=0.10 sec

Best model: ARIMA(3,0,1)(0,0,0)[0] intercept

```

Figura 7 Resultado de la función auto-arima para determinar el orden del modelo que se usará

3) Teniendo en cuenta lo resultados de los pasos anteriores, se procede a crear y entrenar el modelo SARIMAX, posteriormente se procede a generar predicciones de datos, comparándolos con el conjunto de testeo que previamente fue dividido, obteniendo resultados gráficos como el mostrado en la figura 8:

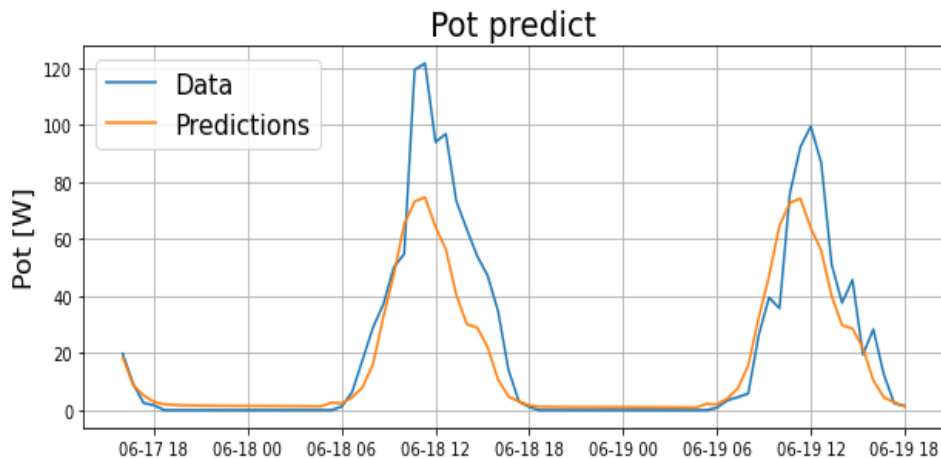


Figura 8 Resultado de las predicciones realizadas con el modelo

Con estos pasos realizados como último procedimiento se calculan los valores del error medio absoluto y del error cuadrático medio, arrojando los siguientes valores:

- Mean Absolute Percent Error: 8.13704
- Root Mean Squared Error: 14.4657

Como se mencionó anteriormente estos mismos pasos fueron realizados para las diferentes series de datos, los resultados se muestran en la tabla 1, y en la figura 9 se observa la comparativa gráfica de estos mismos resultados.

Frec. remuestreo [min]	Mean absolute error	Mean squared error	ARIMA order	Seasonal order	Model fit time [sg]
10	13,93026662943870	25,24009043771580	3,0,3	144	962
20	10,19654945701300	18,52783869724420	2,0,2	72	82
30	8,53955151971666	15,84461019926820	2,0,1	48	18
40	8,13704017369487	14,46577073456160	2,0,2	36	9
60	8,13704017369487	14,46577073456160	3,0,1	24	2

Tabla 1 Resultados para las diferentes series de datos según su frecuencia de submuestreo.

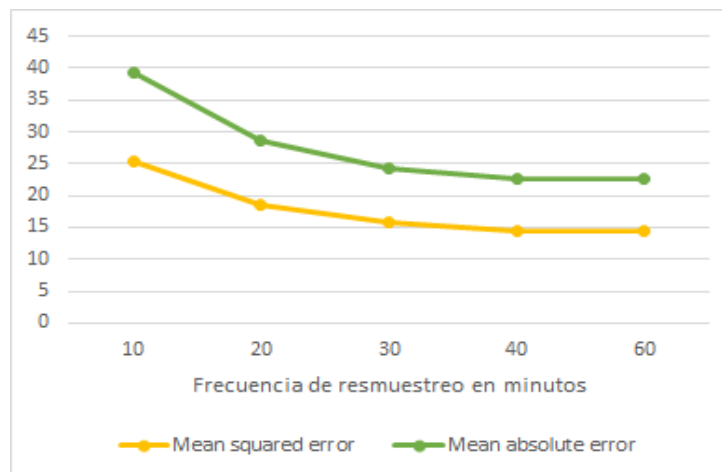


Figura 9 Error absoluto y cuadrático medio para las diferentes series de datos según su frecuencia de submuestreo.

5.2.1.2 Modelo de redes neuronales

Para la creación de la red neuronal artificial (ANN) basada en la predicción de datos de potencia para plantas solares, se usa el mismo Dataset mencionado para el modelo ARIMA, con ayuda de librerías de Python como Numpy, Keras y TensorFlow se realiza la creación de la ANN con diferentes parámetros iniciales. Se realizan varias pruebas entre diferentes arquitecturas de redes neuronales con el fin de establecer la mejor de ellas, encontrando que el mejor desempeño se obtuvo para la red presentada en la figura 10, la cual se compone de 3 neuronas en su capa de entrada, las cuales corresponden a las variables que pueden aportar información al proceso de predicción; en este caso la irradiancia, la temperatura del panel y la temperatura ambiente. En las dos siguientes

capas ocultas de la red neuronal se tienen 10 neuronas y a la salida una sola neurona, en todas las capas de la red neuronal, cada neurona cuenta con una función de activación tipo relu, y miden su pérdida con el error cuadrático medio.

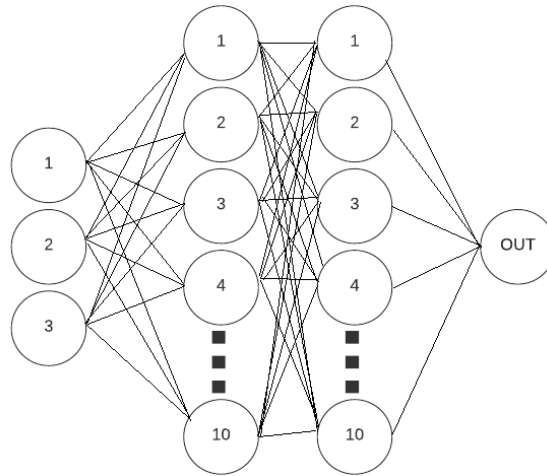


Figura 10 Arquitectura ANN para predicción de potencia en sistemas solares seleccionada

La variable más importante dentro de este proceso de generación de energía es la irradiación solar por su correlación con la potencia generada. La temperatura del panel y ambiental también tienen una buena correlación con la variable de salida, por lo cual, las 3 variables mencionadas anteriormente serán las entradas de la red neuronal. El entrenamiento de la red se hace con 300 iteraciones, y con un número de 64 datos por cada ciclo de iteración (`batch_size`) para su aprendizaje. Se logró obtener un resultado considerablemente bueno, que se observa en la figura 11, donde se observa la comparativa con los datos reales y los obtenidos a partir de la predicción del modelo descrito. En la figura 12 se observa el comportamiento de la función de error a través de las iteraciones, obteniendo como resultado final un error cuadrático medio $MSE = 0.0659$, después del proceso de validación, lo que indica que el proceso de predicción es muy acertado y los valores obtenidos son muy cercanos a los reales. Se usó el Data Frame con los datos escalados.

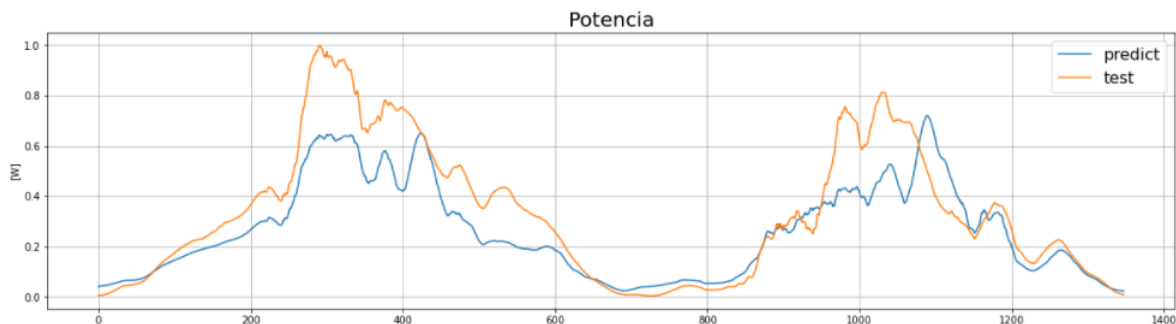


Figura 11 Predicción de potencia con ANN

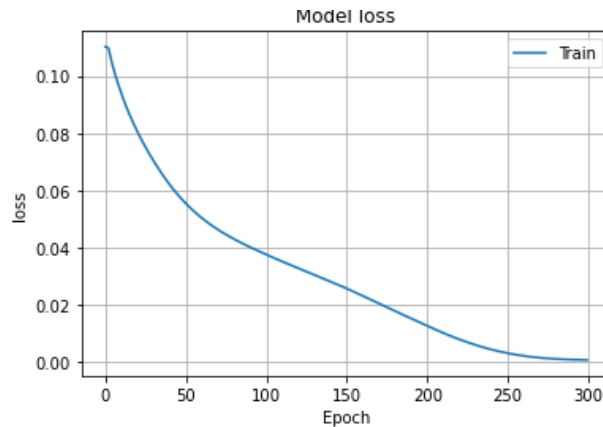


Figura 12 Error cuadrático medio en función de las iteraciones del modelo

5.2.2 Integración con .NET en lenguaje C#

5.2.2.1 Integración con interfaz grafica

Para la integración con .NET, se realizó una interfaz gráfica como la que se muestra en la figura 13, desarrollada en lenguaje C# con Windows Forms. Al ser un prototipo básico, cuenta con funcionalidades sencillas que pueden dar pie a algunas más avanzadas cuando se integre con el software AB. Se puede observar un área de visualización de datos, así como también submenús de carga de datos y de selección de modelo para su entrenamiento y predicción.

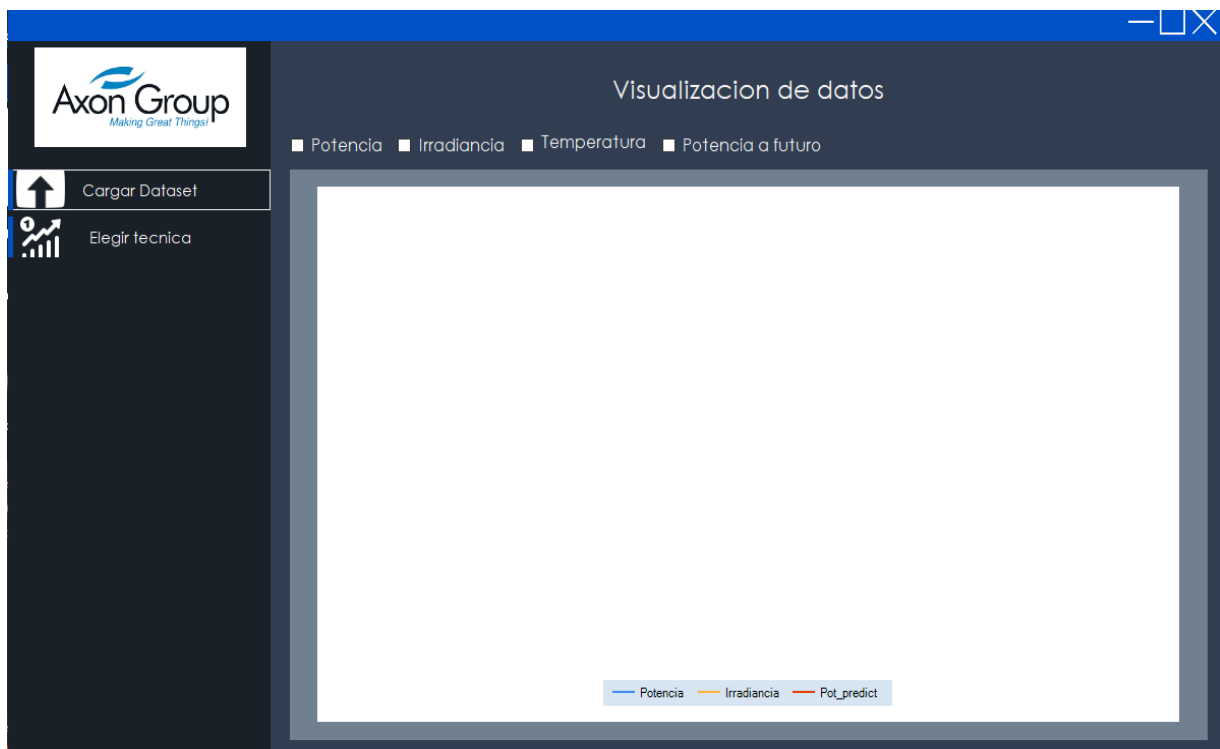


Figura 13 Interfaz prototipo funcional implementado en C#

Una vez realizada de carga de datos mediante una ventana de diálogo de selección de archivos, se grafican las variables más relevantes en el área de visualización (figura 14), donde a partir de checkbox's (figura 15) es posible seleccionar solo las variables que el usuario esté interesado en observar (figura 16).

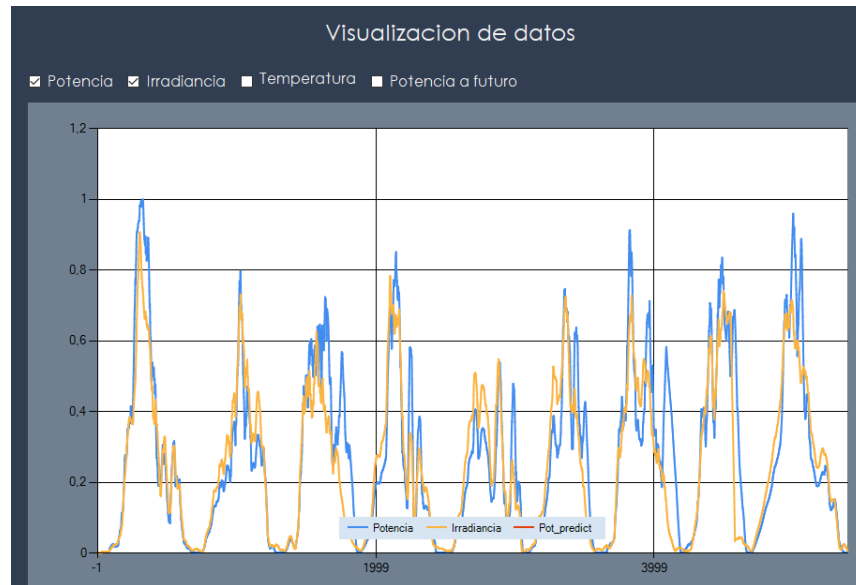


Figura 14 Gráfica de variables en el área de visualización

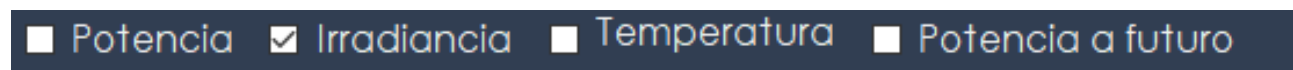


Figura 15 Lista de Checkbox para selección de variables de interés por el usuario

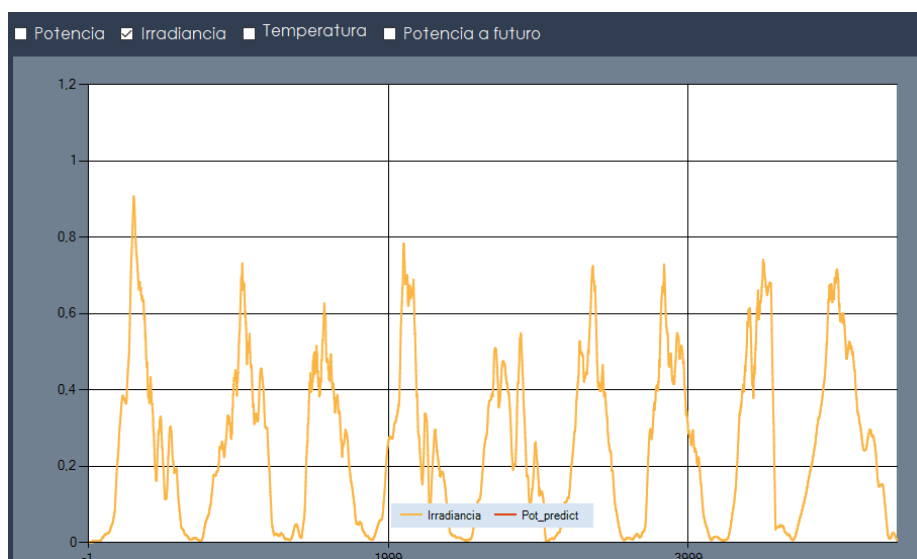


Figura 16 Variable seleccionada por el usuario mediante Checkbox

Una vez realizada la carga de datos el usuario puede ir al submenú mostrado en la figura 17, donde se presentan diferentes técnicas para realizar la predicción de potencia, en este trabajo solamente se implementó el modelo de redes neuronales, para dicha predicción, una vez seleccionada la opción correspondiente se muestra un diálogo(figura 18) donde el usuario tiene 3 opciones: entrenar el modelo con parámetros predeterminados por el programador, entrenar dicho modelo con los parámetros que ingrese por medio de la ventana de diálogo o cargar un modelo previamente entrenado en formato de documento “.h5” el cual es el formato destinado por la librería Keras, de Python ,para guardar modelos previamente entrenados.



Figura 17 Submenú, elección de técnica de entrenamiento.

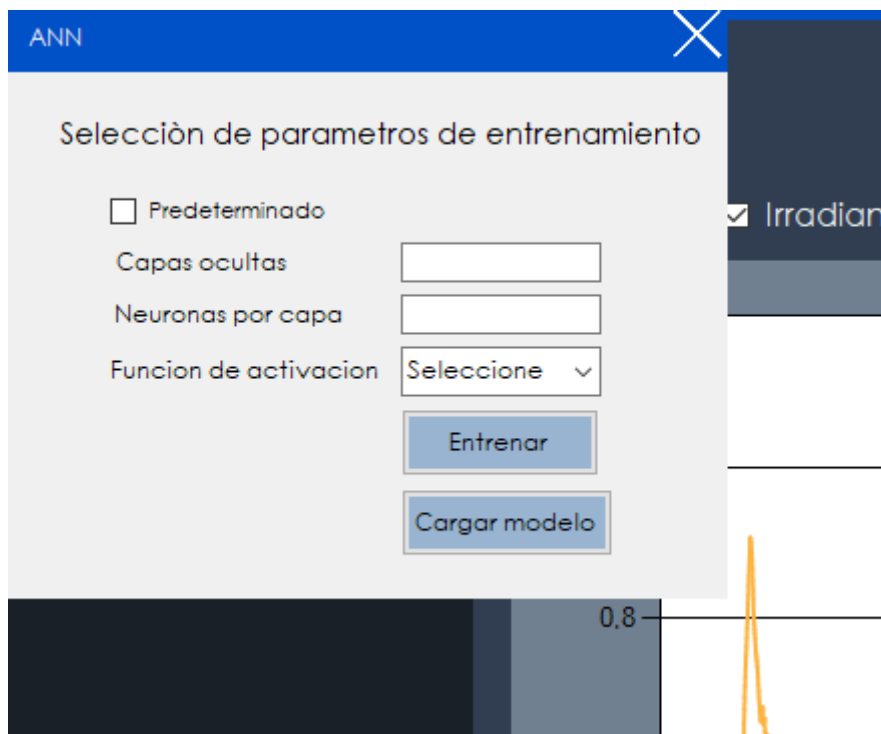


Figura 18 Selección de parámetros para entrenamiento o carga de modelo

Una vez realizada la elección a conveniencia, en el área de visualización va a ser posible observar como una variable más la predicción de potencia. (figura 19)

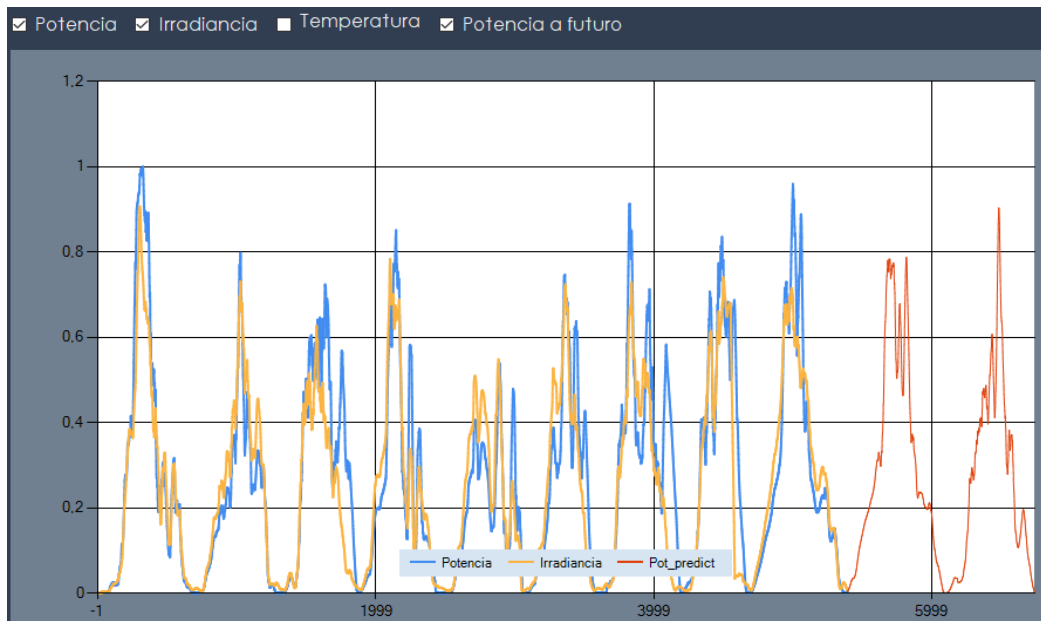


Figura 19 Predicción de potencia graficada en el área de visualización de datos mediante una ANN

A partir de este prototipo funcional y de los modelos estudiado e implementados previamente es posible generar algunas líneas futuras o mejoras para la continuación de estos:

- En lugar de realizar una carga de datos por medio de un archivo, lo ideal sería obtener datos en tiempo real de las instalaciones fotovoltaicas, con ayuda, principalmente, de la estación meteorológica, dentro de estos datos se encontraría las temperaturas (ambiental y del panel), la irradiancia, entre otras.
- Personalización de gráficas mostradas, con ajustes como rangos de tiempos y mayor número de variables disponibles.
- Integración del prototipo como servicio de invocación cíclica dentro del sistema SCADA Axon Builder.
- Usar modelos de inteligencia artificial para otros fines dentro del contexto de la automatización del sector eléctrico (ej. detector de fallas con técnicas de aprendizaje automático).
- Combinación de técnicas de predicción; a partir de un modelo ARIMA generar predicciones de irradiancia para usar como entrada de una red neuronal.
- Integración con API de predicción de irradiancia.

6. Conclusiones

- La planeación en cualquier proyecto es fundamental, tener un plan de trabajo desde el comienzo del mismo evita incurrir en errores de estimación de tiempos, así como también errores en los alcances. La metodología Agile Scrum permite tener un enfoque más claro del paso a paso que se realiza en cada parte del proyecto. Para llevar a cabo este proyecto, los planes de trabajo y los sprints de la metodología fueron planificados teniendo en cuenta todas las especificaciones que cada tarea necesita, pues con ellas fue posible seguir el camino adecuado para conseguir el objetivo final, con las especificaciones requeridas y necesarias.
- La exploración de sistemas fotovoltaicos a nivel personal y de la empresa es algo nuevo, pues no se había profundizado sobre ello. Por lo cual, se hizo necesario realizar una documentación con toda esta información pertinente, que abarca desde los temas más básicos de estos sistemas, hasta los pequeños detalles que puedan resultar importantes en este entorno de la generación de la energía. Dicha documentación y exploración fue necesaria para tener un contexto muy claro de lo que se quería realizar, por ello resultó de gran provecho y de gran importancia para el proyecto.
- Hoy en día la inteligencia artificial tiene un impacto muy grande en la mayoría de los sectores industriales, adoptar técnicas de predicción puede llegar a generar un valor adicional a los productos de una empresa, es por ello que se estudió toda esta viabilidad en el contexto del Axon Builder, con el fin de evaluar posibles líneas futuras de integración con estas técnicas.
- El desarrollo de modelos de predicción y de técnicas de inteligencia artificial resulta más sencillo en el lenguaje Python, pues cuenta con librerías que tienen como objetivo permitirle al programador hacer más fácil su implementación. El Axon Builder está implementado en .NET sobre el lenguaje C#, en este lenguaje, no es tan factible implementar dichos métodos de predicción, por lo que se hizo necesario encontrar alternativas para cumplir con el objetivo, entre ellas: Keras.NET, comunicación por medio de archivos .h5 (modelos) y comunicación por medio de ejecutables de Python.
- Después de analizar los modelos implementados se encuentra una ventaja particular para cada uno de ellos. En el caso del modelo ARIMA, no es necesario tener los datos de la irradiancia para poder predecir los comportamientos que pueda llegar a tener la variable de

interés, pues la predicción usa valores pasados, sin embargo, esto afecta la precisión del modelo. Para el caso de las redes neuronales es lo contrario, al contar con la irradiancia y demás variables de entrada, se puede obtener una predicción más precisa.

- El manejo de los datos es un factor fundamental dentro del desarrollo de modelos de inteligencia artificial. Para el desarrollo de este trabajo, como se mencionó anteriormente, tener una base de datos sólida, amplia y confiable fue una limitante, no obstante, a los datos con los cuales se trabajó, se hizo necesario realizar un preprocesamiento. Escalar los datos, normalizar y remuestrear los puede ser de gran utilidad. Un caso específico de esto se da en la creación del modelo ARIMA, pues fue necesario realizar un muestreo con una frecuencia menor a la inicial debido a que la capacidad computacional no era suficiente si no se realizaba dicho procedimiento.

7. Referencias Bibliográficas

- [1] Twidell, J., & Weir, T. (2015). Renewable energy resources. Routledge.
- [2] Sorokin, A. A., Bobashev, S. V., Feigl, T., Tiedtke, K., Wabnitz, H., & Richter, M. (2007). Photoelectric effect at ultrahigh intensities. *Physical review letters*, 99(21), 213002.
- [3] M. A. Laughton, D. J. Warne (ed), *Electrical Engineer's Reference book*, 16th edition, Newnes, 2003 Chapter 16 Programmable Controller.
- [4] Integrated Dynamics, Inc. 2018. Power Plant Controller (PPC) for Grid-Connected Power Plants with Energy Storage.
- [5] Zolotova, I., & Landryová, L. (2000). Scada/HMI systems and emerging technologies. *IFAC Proceedings Volumes*, 33(1), 17-20.
- [6] Penin, A. R. (2011). *Sistemas Scada*. Marcombo.
- [7] Taylor, J. K., & Cihon, C. (2004). *Statistical techniques for data analysis*. CRC Press.
- [8] De Arce, R., & Mahía, R. (2003). *Modelos Arima*. Programa CITUS: Técnicas de Variables Financieras.
- [9] López, R. F., & Fernandez, J. M. F. (2008). *Las redes neuronales artificiales*. Netbiblo.
- [10] Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE). *Geoscientific Model Development Discussions*, 7(1), 1525-1534.
- [11] Ocampo, O. (2020). *Monitoreo de la eficiencia de paneles fotovoltaicos bajo condiciones de lluvia y polvo*. Universidad de Antioquia.
- [12] Fuente base de datos: <http://iot.kontrolar.co/TeslaUdeA/>. Grupo de investigación TESLA UdeA.
- [13] F. Wang, Z. Mi, S. Su, H. Zhao, "Short-Term Solar Irradiance Forecasting Model Based on Artificial Neural Network Using Statistical Feature Parameters", *Energies*, vol. 5 (5), pp. 1355-1370, 2012.
- [14] S.K.H. Chow, E.W.M. Lee, D.H.W. Li, "Short-term prediction of photovoltaic energy generation by intelligent approach", *Energy and Buildings*, vol. 55, pp. 60-667, 2012.
- [15] A. Mellit, S. Saïlam, S.A. Kalogirou, "Artificial neural network-based model for estimating the produced power of a photovoltaic module", *Renewable Energy*, vol. 60, pp. 71-78, 2013.
- [16] Y. Li, Y. Su, L.J. Shu, "An ARMAX Model for Forecasting the Power Output of a Grid Connected Photovoltaic System," *Renewable Energy*, vol. 66, pp. 78-89, 2014.

- [17] Paoli, Ch., Voyant, C., Muselli, M., Nivet, M. (2009). Forecasting of preprocessed daily solar radiation time series using neural networks. *Solar Energy* 84 (2010) 2146–2160.
- [18] European Photovoltaic Industry Association. (2018). “Solar Photovoltaic Electricity Empowering the World”. *Solar Generation* 6. Disponible en http://pvtrin.eu/assets/media/PDF/Publications/Other%20Publications/36.SolarGeneration6__2011.pdf
- [19] Shahsavari, A., Bagherzadeh, S.A., Afrand, M. (2021). Application of artificial intelligence techniques in prediction of energetic performance of a hybrid system consisting of an earth-air heat exchanger and a building-integrated photovoltaic/thermal system. *Journal of Solar Energy Engineering, Transactions of the ASME* 143(5),051002.
- [20] Elsheikh, A.H., Sharshir, S.W., Abd Elaziz, M., Guilan, W., Haiou, Z. (2019). Modeling of solar energy systems using artificial neural network: A comprehensive review. *Solar Energy* 180, pp. 622-639.
- [21] Wang, F., Xuan, Z., Zhen, Z., Wang, T., Shi, M. (2020). A day-ahead PV power forecasting method based on LSTM-RNN model and time correlation modification under partial daily pattern prediction framework. *Energy Conversion and Management* 212,112766.
- [22] Cheung, Y. W., & Lai, K. S. (1995). Lag order and critical values of the augmented Dickey–Fuller test. *Journal of Business & Economic Statistics*, 13(3), 277-280.
- [23] De Giorgi, M. Malvoni, M. Congedo. P. (2017). Forecasting of PV Power Generation using weather input data-preprocessing techniques.