TECHNICAL UNIVERSITY OF MOLDOVA

FACULTY OF COMPUTERS, INFORMATICS AND MICROELECTRONICS

DEPARTMENT OF SOFTWARE ENGINEERING AND AUTOMATICS

Energy Efficiency System

Mentor: *univ. professor*

______________ Andrei Bragarenco

Academic group: FAF 202

Team Members: ______________ Babcinetcțhi Egor

______________ Moisei Liviu

______________ Purici Marius

______________ Covalevschi Andreea

______________ Gîlca Constantina

Chișinău, 2023

CONTENTS

ABSTRACT

Energy efficiency is a crucial aspect of modern society, as we strive to reduce our carbon footprint and preserve our planet for future generations. As students, we have taken on the challenge of implementing a forecasting-based system that aims to predict the energy output of solar panels. This system will use past training data and weather forecasting for the next 24 hours to accurately predict how much energy a solar panel will produce in the following day, on an hourly basis.

The goal of this project, as clearly stated by our stakeholder, is to provide a reliable and accurate tool for predicting the energy output of solar panels. This will enable individuals and organizations to better plan their energy usage and reduce their reliance on non-renewable energy sources. By using advanced forecasting techniques and machine learning algorithms, we hope to achieve a high level of accuracy in our predictions.

The purpose of this project is to help the owner of the solar panels by reducing costs when the solar panels produce more or less than the predicted value. By providing accurate predictions, we can help individuals and organizations make informed decisions about their energy usage and reduce their overall costs.

In addition to reducing costs for the owner of the solar panels, this project also has broader implications for society as a whole. By promoting the use of renewable energy sources and reducing our reliance on non-renewable sources, we can help to reduce greenhouse gas emissions and combat climate change. This project represents an important step towards a more sustainable future.

As students, we are committed to using our skills and knowledge to make a positive impact on the world. We believe that this project has the potential to make a significant contribution to the field of renewable energy and help pave the way for a more sustainable future.

We have assembled a team of dedicated and talented individuals who are passionate about making a difference. Our team includes experts in fields such as data science, machine

learning, and renewable energy. Together, we are working tirelessly to develop and refine our forecasting system to achieve the highest level of accuracy possible.

We are excited about the potential of this project and are committed to seeing it through to completion. We believe that by working together, we can achieve great things and make a positive impact on the world.

INTRODUCTION

Energy production and consumption are critical components of modern society. With the growing demand for energy and the increasing importance of renewable energy sources, such as solar energy, it has become necessary to improve the management of energy production, distribution, and con- sumption. The traditional energy management systems that rely on manual interventions and lack automation have several limitations, including inefficiency, high cost, and the inability to adapt to the changing needs of consumers. However, with the development of sophisticated Energy Management Systems (EMS) for solar energy, it has become possible to manage and optimize energy production and distribution more efficiently, cost-effectively, and with greater flexibility. In this context, EMS plays a crucial role in automating the entire process, from energy generation to consumption, enabling businesses and households to benefit from solar energy while also reducing their carbon footprint.

This is where automation and energy management systems come in. By integrating automation technology such as SCADA and IoT into solar energy distribution, it is possible to optimize the performance of solar power plants and make them more efficient and profitable. This can involve real-time monitoring and control of energy production and distribution, as well as the collection and analysis of data to improve decision-making processes.

An effective energy management system for solar energy can help reduce operational costs, increase energy production, and enhance the overall performance of a solar power plant. In addition, it can help to reduce carbon emissions and support the transition to a more sustainable energy system.

1. DOMAIN ANALYSIS

1.1 EMS for solar energy

Energy management systems (EMS) have become a crucial part of solar energy systems in recent years. An EMS for solar energy is a software-based system that allows efficient and effective management of energy generation, storage, and consumption from solar panels. It is designed to optimize the energy usage of a solar energy system by tracking and analyzing real-time data such as weather forecasts, electricity consumption patterns, and energy storage levels. By integrating an EMS into a solar management system, solar energy providers and consumers can maximize energy production and minimize energy costs.

One of the key advantages of integrating an EMS into a solar management system is the ability to collect, analyze, and act upon data in real-time. With the help of an EMS, solar energy providers can monitor energy consumption patterns, forecast energy demand, and identify opportunities to optimize energy generation and storage. This enables them to make data-driven decisions that can lead to improved energy efficiency, cost savings, and reduced environmental impact.

The integration of the Internet of Things (IoT) technology into EMS has further enhanced the capabilities of solar energy systems. By adding IoT devices, such as sensors and smart meters, to a solar energy system, EMS can gather even more data, including information about solar panel performance and battery storage levels. This data can be used to further optimize energy usage and identify potential issues with the system.

As solar energy systems continue to grow in popularity, there is a greater need for efficient and effective energy management. The integration of IoT technology into EMS is expected to play a key role in this development. With the increasing availability of IoT devices and the ongoing improvements in connectivity and data analytics, EMS for solar energy systems are becoming more sophisticated and user-friendly.

1.2 SCADA for EMS

SCADA (Supervisory Control and Data Acquisition) can be used to improve energy management systems for solar energy by providing remote monitoring and control of the solar power plant. This allows for real-time monitoring of key parameters such as solar irradiation, temperature, and power output, which can be used to optimize the performance of the solar panels and the overall energy system. SCADA systems can also facilitate fault detection and diagnosis, enabling quick identification and resolution of any issues that may arise. This can result in improved energy efficiency and reduced downtime, leading to greater productivity and profitability. Additionally, the data collected by SCADA systems can be used for performance analysis and forecasting, providing valuable insights into system performance and enabling informed decision-making for future improvements.

With SCADA integrated into a solar energy management system, data can be collected in real- time from various sensors and devices such as solar panels, inverters, and weather stations. This data can be stored and analyzed to provide statistics such as energy production, consumption, and system efficiency. This information can then be used to make informed decisions regarding system maintenance, performance optimization, and energy management strategies. The ability to collect and analyze data in real-time allows for quick identification and resolution of issues, which ultimately leads to increased system efficiency and reduced costs.

Solar energy can use a SCADA system in several ways. A SCADA system can monitor and control various components of a solar energy system such as photovoltaic (PV) panels, inverters, batteries, and other devices. The system can collect data on energy production, consumption, and storage and provide real-time performance monitoring and control. This data can be analyzed to optimize system efficiency and identify potential issues before they become major problems.

SCADA systems can also enable remote monitoring and control of solar energy systems, which is particularly useful for large-scale solar farms or distributed solar installations. With a SCADA system, operators can monitor and control the solar energy system from a central location, reducing the need for on-site personnel and improving overall system reliability.

In addition, SCADA systems can collect data in the form of statistics, such as energy production, consumption, and storage over time. This data can be used for trend analysis and forecasting, enabling operators to plan and optimize the system for maximum efficiency and profitability.

Data that can be collected in the form of statistics from a solar management system include information on the amount of solar energy produced, the energy output of individual solar panels, the amount of energy stored in batteries, and the amount of energy consumed by appliances or devices connected to the system.

By analyzing this data, customers can gain insights into the performance of their solar energy system, identify potential areas for optimization, and make data-driven decisions to increase their profit. For example, customers can use the data to:

- Optimize the placement of solar panels to maximize energy production;
- Identify and fix issues with individual panels that may be underperforming or damaged;
- Schedule the use of energy-consuming appliances during periods of high solar energy production to reduce reliance on grid energy and lower energy bills;
- Monitor the effectiveness of energy storage systems to optimize energy usage and reduce costs;
- Analyze energy usage patterns to identify opportunities to further reduce energy consumption and costs.

By collecting and analyzing data in the form of statistics, customers can make informed decisions about how to optimize their solar energy systems and increase their profits over time.

1.3 SCADA and IoT

The integration of SCADA systems with IoT technology can enhance the efficiency and effectiveness of energy management systems for solar energy. IoT devices such as sensors and smart meters can be deployed to collect and transmit real-time data from solar panels, inverters, and other components. This data can be analyzed using advanced algorithms and machine learning techniques to improve the overall performance of the solar energy system. For example, predictive maintenance can be performed based on the data collected by IoT devices to detect and prevent equipment failures before they occur. Furthermore, the integration of IoT technology with SCADA systems can enable remote monitoring and control of solar energy systems from anywhere in the world. This can help energy managers to optimize the performance of their solar energy systems by adjusting various parameters such as panel orientation, temperature, and battery charging rates.

1.4 Smart grids in solar energy management systems

"The smart grid (SG) is not only a set of smart energy meters involved in the power generation but is a group of various technologies and tools which allow integration, interaction, and control of all elements from the utility to consumer side for the smooth flow of power. The SG enables real-time monitoring of power system through visualization of all the necessary information and then acts accordingly for its smooth functioning, e.g., immediate balancing of demand and supply, bidirectional flow of electricity etc."

The quote accurately captures the essence of a smart grid (SG) as a system of various technologies and tools that enable integration, interaction, and control of all elements involved in power generation and consumption, from the utility to consumer side. By allowing real-time monitoring of the power system, the SG facilitates the visualization of all necessary information and the immediate balancing of demand and supply, thereby promoting the bidirectional flow of electricity for smooth functioning. The SG's ability to integrate and coordinate different components of the energy system, such as renewable energy sources, energy storage systems, and electric vehicles, can lead to greater energy efficiency and reliability, while also enabling more efficient energy use and management.

Additionally, the SG can enable new business models and services that are not possible with the traditional grid, such as peer-to-peer energy trading and demand response programs.

The automation of the SG process, where human intervention is limited to an additional verifica- tion step, is a crucial aspect of the smart grid's functionality. This is achieved through the integration of supervisory control and data acquisition (SCADA) and energy management systems (EMS), which allow for real-time monitoring and control of the grid's components, such as power generation, trans- mission, and distribution.

The automation of the SG process also allows for the rapid detection and response to system failures, reducing downtime and increasing system reliability. SCADA and EMS systems can detect anomalies in the grid's operation, such as voltage fluctuations or equipment failure, and automatically respond to correct them. This reduces the need for human intervention and can lead to significant cost savings and efficiency improvements.

1.5 Smart meters in Solar EMS

”Smart meters are digital electricity meters that accurately mea- sures both electricity consumption and production and commu- nicate this data to the energy supplier. These meters have the ability to communicate the measured data, which provides them the “smart” aspect. Smart meters are predominantly used by en- ergy suppliers as a contribution for more precise and automated billing. Smart meter also allows its integration in home energy management systems through communication protocols where the provided information is related to energy flow and price sig- nals. Smart meters associated with their related infrastructures enable end-users to be included in the smart grid man- agement context, as they provide information about electricity flow mea- surements and energy prices to end-consumers ”

This quote accurately describes the key features and benefits of smart meters. Smart meters are digital devices that accurately measure electricity consumption and production and can communicate this data to energy suppliers, enabling more precise and automated billing. In addition, smart meters can be integrated into home energy management systems through communication protocols, providing information on energy flow and price signals.

One of the key advantages of smart meters is their ability to enable end-users to be included in the smart grid management context, as they provide information about electricity flow measurements and energy prices to end-consumers. This empowers consumers to make more informed decisions about their energy usage, and can also help to reduce overall energy consumption and costs.

Smart meters can also help businesses to increase their revenue by providing real-time data on energy usage and consumption patterns. This information can be used by businesses to identify areas where energy usage can be reduced or optimized, leading to lower energy costs and higher profits. In addition, smart meters can help businesses to participate in demand response programs, where they can receive incentives for reducing their energy consumption during peak demand periods. This can help businesses to reduce their energy bills and earn additional revenue through participation in energy markets.

Furthermore, smart meters can enable businesses to implement dynamic pricing strategies, where the price of energy varies based on demand and supply conditions. This can help businesses to reduce energy costs during low-demand periods and increase revenue during high-demand periods. For example, a business could shift some of its energy usage to off-peak hours when energy prices are lower, leading to significant cost savings.

Overall, the quote highlights the potential benefits of smart meters for both energy suppliers and end-users, and emphasizes the importance of integrating these devices into wider energy management systems to fully realize their potential.

1.6 Objectives

Following the research and the project requirements set together with the stakeholders of this project, the following objectives were identified:

- To develop a system that would predict the energy obtained by the solar batteries in the following days, based on meteorological data;

- To assure access to the statics and observations of the system to the interested users.
- To provide valuable information regarding solar panel's productivity and performance;
- To ensure system safety and reliability.

In according to the expectations of the stakeholder's of this project and the team capabilities, the objectives were identified as achievable and further requirements regarding the system were established.

1.7 Requirements

Non-functional requirements specify the quality attributes of the system.[1] Hence, there were identified 4 main qualities for the energy efficiency system:

- Usability: the system should be easy to use and easy to understand for the end user with minimal technical knowledge and interference;
- Performance: indifferent of the workload, the system should keep it's highest performance rate;
- Availability: the system should be available to the end user in real time;
- Compatibility: the system should be compatible with the environment and not interfere with the other components in terms of performance or workload.

On the other hand, Functional requirements define what a system should actually do. Following are the 3 main Functional requirements:

- Prediction: the system should be predict the amount of energy produced by the solar panels in the following 3 days;
- Visualisation: the predicted data and the real time status should be displayed in a easy to understand way to the end user;
- Maintainability: the system should alert end users in case of unusual activity or occurrence within the tracked components.

The system design follows these requirements in order to achieve the objectives defined.

2. SYSTEM DESIGN

2.1 System architecture

2.1.1 System components

Sensors

Solar inverters typically use several sensors to monitor and control the flow of energy from the solar panels to the electrical grid. Some of the common sensors used in solar inverters include:

- DC voltage and current sensors: These sensors measure the voltage and current output from the solar panels and send this information to the inverter. This data is used to optimize the output of the panels and ensure that the inverter operates within safe limits.
- AC voltage and current sensors: These sensors measure the voltage and current output from the inverter to the electrical grid. This data is used to ensure that the inverter is producing power at the correct frequency and voltage levels.
- Temperature sensors: These sensors measure the temperature of the inverter and the surrounding environment. This data is used to ensure that the inverter does not overheat, which can cause damage to the components and reduce its lifespan.
- Light sensors: These sensors measure the amount of sunlight that is hitting the solar panels. This data is used to optimize the output of the panels and ensure that they are producing power at their maximum capacity.
- Humidity sensors: These sensors measure the level of moisture in the air. This data is used to ensure that the inverter is not exposed to conditions that could cause corrosion or damage to the components.

”MPP Solar Inc” is a company that manufactures and supplies a range of solar products, including MPP solar inverters. Their MPP solar inverters are designed to convert the DC power generated by solar panels into AC power for use in homes, businesses, or for feeding into the grid.

MPP Solar Inc’s inverters typically include MPPT (Maximum Power Point Tracking) technology to ensure maximum energy yield from the solar panels, as well as data monitoring capabilities for remote monitoring and control. They offer a range of inverters with different power ratings, from small residential systems to large commercial and industrial installations.

It’s worth noting that while ”MPP Solar Inc” is a company that manufactures inverters, the term ”MPP solar inverter” is not specific to this particular brand or company, but rather refers to the type of solar inverter that utilizes maximum power point tracking technology.

Communication Networking

MPP Solar inverters may have various communication interfaces, depending on the specific model and features. Here are some of the common communication interfaces found in MPP Solar inverters:

- RS485: This is a serial communication interface that enables multiple devices to be connected together in a network.
- USB: This is a common interface used for connecting devices such as computers, smartphones, and tablets. Some MPP Solar inverters may have a USB port for data exchange and firmware updates.
- Ethernet: This interface enables the inverter to be connected to a network, such as the internet or a local area network (LAN), for remote monitoring and control.
- WiFi: Some MPP Solar inverters may have built-in WiFi capabilities, which allows the inverter to connect to a wireless network and be monitored and controlled via a smartphone app or web browser.
- Bluetooth: This interface allows the inverter to connect to other Bluetooth-enabled devices, such as smartphones, tablets, or laptops, for data exchange and remote monitoring and control.
- CANbus: This is a communication protocol used in automotive and industrial applications. Some MPP Solar inverters may use CANbus for communication

with other devices in a system, such as battery management systems or energy management systems.

PC Software

MPP Solar provides various PC software for their inverters that allow users to monitor and configure the inverters. The software available may vary depending on the specific model and features of the inverter. Here are some of the common PC software provided by MPP Solar for their inverters:

- SolarPower: This is a Windows-based software that allows users to monitor and configure MPP Solar inverters via a USB or RS232 connection. The software provides real-time monitoring of the inverter's performance, including output power, input voltage, and current, and allows users to adjust various settings, such as battery charging parameters and output voltage.

- WatchPower: This is a Windows-based software that allows users to monitor and configure MPP Solar inverters via a USB or RS232 connection. The software provides real-time monitoring of the inverter's performance, including output power, input voltage, and current, and allows users to adjust various settings, such as battery charging parameters and output voltage. WatchPower also supports remote monitoring and control via Ethernet or WiFi.
- SolarPowerManager: This is a Windows-based software that allows users to manage multiple MPP Solar inverters in a system. The software provides real-time monitoring of the inverters' performance, including output power, input voltage, and current, and allows users to adjust various settings, such as battery charging parameters and output voltage. SolarPowerManager also supports remote monitoring and control via Ethernet or WiFi.
- SolarAssistant: This is a mobile app for Android and iOS devices that allows users to monitor and configure MPP Solar inverters via Bluetooth or WiFi. The

app provides real-time monitoring of the inverter's performance, including output power, input voltage, and current, and allows users to adjust various settings, such as battery charging parameters and output voltage.

The IoT system developed in this project is using WatchPower to monitor and configure inverter

Main Board

Once connected, the Raspberry Pi can be used to monitor and control the MPP Solar inverter, such as reading its operating parameters, setting configurations, or issuing commands. This can be useful for building custom monitoring or control systems, such as a home energy management system or a solar power plant monitoring system.

Some MPP Solar inverters have an RS232 port, which can be used to communicate with the inverter using a suitable communication protocol, such as MODBUS RTU. To connect a Raspberry Pi to the inverter via RS232, one can use a USB-to-RS232 converter cable and install the necessary drivers and software on the Raspberry Pi.

User Interface

Users can interact with a Raspberry Pi in several ways, depending on the intended use case and level of technical expertise. Here are some common ways to interact with a Raspberry Pi:

- Command Line Interface (CLI): The Raspberry Pi runs a Linux-based operating system (such as Raspbian) which can be accessed via the command line interface (CLI). Users can connect to the Raspberry Pi via SSH (Secure Shell) from another computer and use a terminal program to execute commands and manage the system.
- Graphical User Interface (GUI): The Raspberry Pi can also be configured to run a graphical user interface (GUI) desktop environment, such as LXDE, XFCE, or

GNOME. This can be useful for users who prefer a more intuitive and user-friendly interface to manage the system and run applications.

- Web-based Interface: Users can also create web-based interfaces to interact with the Raspberry Pi, using web development technologies such as HTML, CSS, and JavaScript. This can be useful for building web applications that can run on any device with a web browser and connect to the Raspberry Pi remotely.
- External Devices: Users can connect external devices, such as keyboards, mice, and monitors, to the Raspberry Pi's USB or HDMI ports to interact with the system directly. This can be useful for users who prefer a more traditional desktop computing experience.

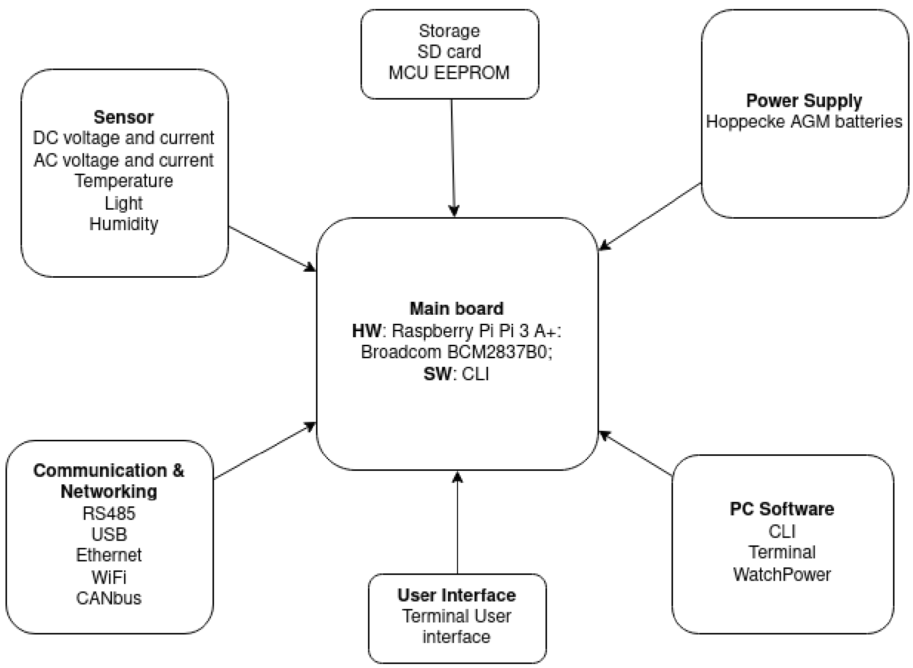

Figure 1: System components diagram

2.1.2 System architecture

The device layer consists of invertor and an RPi module. The invertor collects metric data that includes the input voltage, output voltage and battery, UPS, EMD information. For sending sensor data from an MPP Solar inverter to a Raspberry Pi, it is recommended to establish a communication link between the two devices.

It is recommended to identify the communication interface supported by the inverter by checking its documentation. This could be RS232, USB, Ethernet, or another interface.

To send data from a Raspberry Pi to a remote server via a gateway, data flow would typically follow these steps:

- The Raspberry Pi would collect data from various sensors or other sources.
- The software on the Raspberry Pi would process the data and format it for transmission.
- The Raspberry Pi would send the data to the gateway device using a suitable communication protocol.

By following this data flow, the Raspberry Pi can communicate with a remote server via a gateway device, enabling remote monitoring or control of the device.

The server receives data packet, processes it and generates predictions based on past and current invertor data.

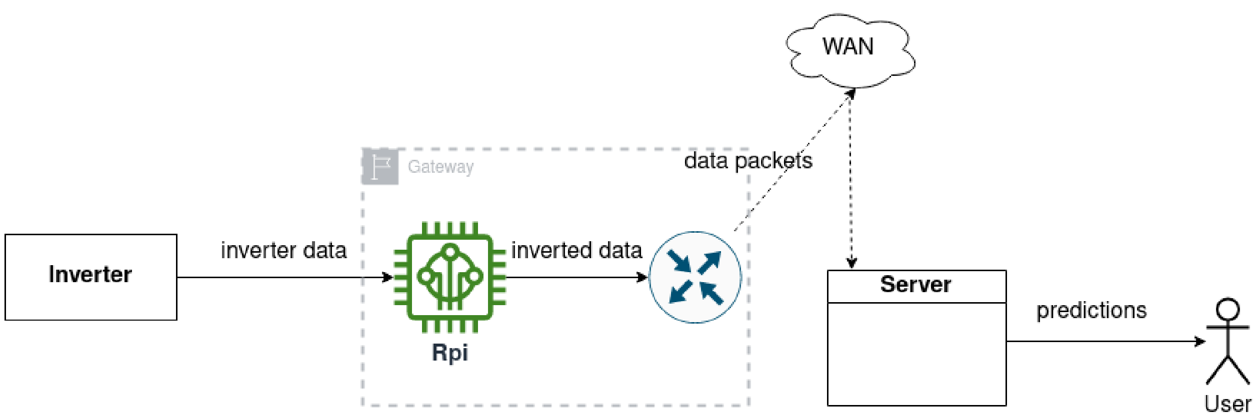

Figure 2: System architecture of the IoT System

The server itself is a subsystem of components that provide the main functionalities of the whole data-processing system.

Message broker

After the remote server receives the data forwarded by the gateway device, the data can be processed and stored in a database or other storage system. However, to enable real-time processing and analysis of the data, it may be necessary to pass the data to a message broker such as RabbitMQ.

A message broker acts as an intermediary between producers and consumers of data, allowing data to be queued up and delivered in a reliable, scalable manner. By passing data from the remote server to a message broker, it becomes possible to process the data in real-time, trigger alerts or notifications based on the data, and enable other advanced functionality.

It is important to store the intermediate data in a queue, such as a message broker, to prevent data loss and ensure reliability of the communication. By queuing the data, it can be stored in the queue until the receiving end is ready to process it, ensuring that no data is lost or dropped due to network issues or processing delays. Additionally, message brokers often offer features such as message persistence, message acknowledgement, and message filtering, which can further enhance the reliability and flexibility of the data flow.

Aggregator Service

After the data is passed to a message broker, an aggregator can be set up to read the data from the message broker's queue, validate it, and store it in a local database.

The aggregator can be a separate software component that runs on a different server or even on the same server as the message broker. It subscribes to the data topic in the message broker and receives data as it arrives. The aggregator then performs validation on the data to ensure that it is in the expected format and meets any other required criteria. If the data is valid, it is stored in a local database for further processing or analysis.

Storing the data in a local database has several benefits. First, it provides a backup of the data in case of issues with the remote server or message broker. Second, it allows for faster access to the data for analysis or processing, as the data does not need to be fetched from a remote server or message broker each time it is needed. Finally, it enables the aggregator to perform additional processing or analysis on the data before passing it to other systems or applications.

Prediction Service

After the aggregator stores the data in the local database, a prediction server can be set up to read the persisted data from the database and use it to generate predictions of energy production based on weather forecast and machine learning.

The prediction server can be a separate software component that runs on a different server or on the same server as the aggregator. It reads the sensor data from the local database and combines it with weather forecast data to generate predictions of energy production. Machine learning algorithms can be used to analyze historical sensor data and weather data to generate accurate predictions.

The prediction server then stores the predictions back into the database, along with other relevant metadata such as the time of prediction and the weather forecast used. The predictions can be used for a variety of purposes, such as optimizing energy usage, predicting energy demand, or predicting equipment failures.

By using a prediction server to generate predictions of energy production, the data flow becomes more intelligent and adaptive. The prediction server uses machine learning algorithms to analyze historical data and generate accurate predictions, enabling better decision-making and more efficient use of energy resources.

Analytics service

After the prediction server generates predictions of energy production, the data can be passed on to an analytics service for further processing. The analytics service can take the predicted energy production data and generate prediction reports, which provide energy production predictions for the next day, broken down by hour.

Web Page

The final step in the data flow is the web page, which provides users with a user interface to view the energy production predictions, filter by preferences, configure the prediction service, and export data. The web page can be designed using HTML, CSS, and JavaScript, and can be served by the analytics service using a web framework like Flask or Django. Users can access the web page using a web browser and interact with the various features provided.

The web page can display the energy production predictions for the next day, broken down by hour, along with additional information such as weather forecasts and historical energy production data. Users can filter the predictions by various criteria, such as geographical location, time of day, or type of energy source.

Users can also configure the prediction service by adjusting parameters such as the forecasting model used, the input data sources, and the prediction intervals. These changes can be submitted to the prediction service through the web page and will take effect in subsequent prediction reports.

Finally, the web page can provide users with the ability to export the prediction data in various formats, such as CSV or JSON, for further analysis or integration with other systems. Users can specify the data range and format and download the data directly from the web page.

By providing a user-friendly web page, the analytics service can enable users to easily access and interact with the energy production prediction data, empowering them to make informed decisions and optimize their energy usage and resource allocation.

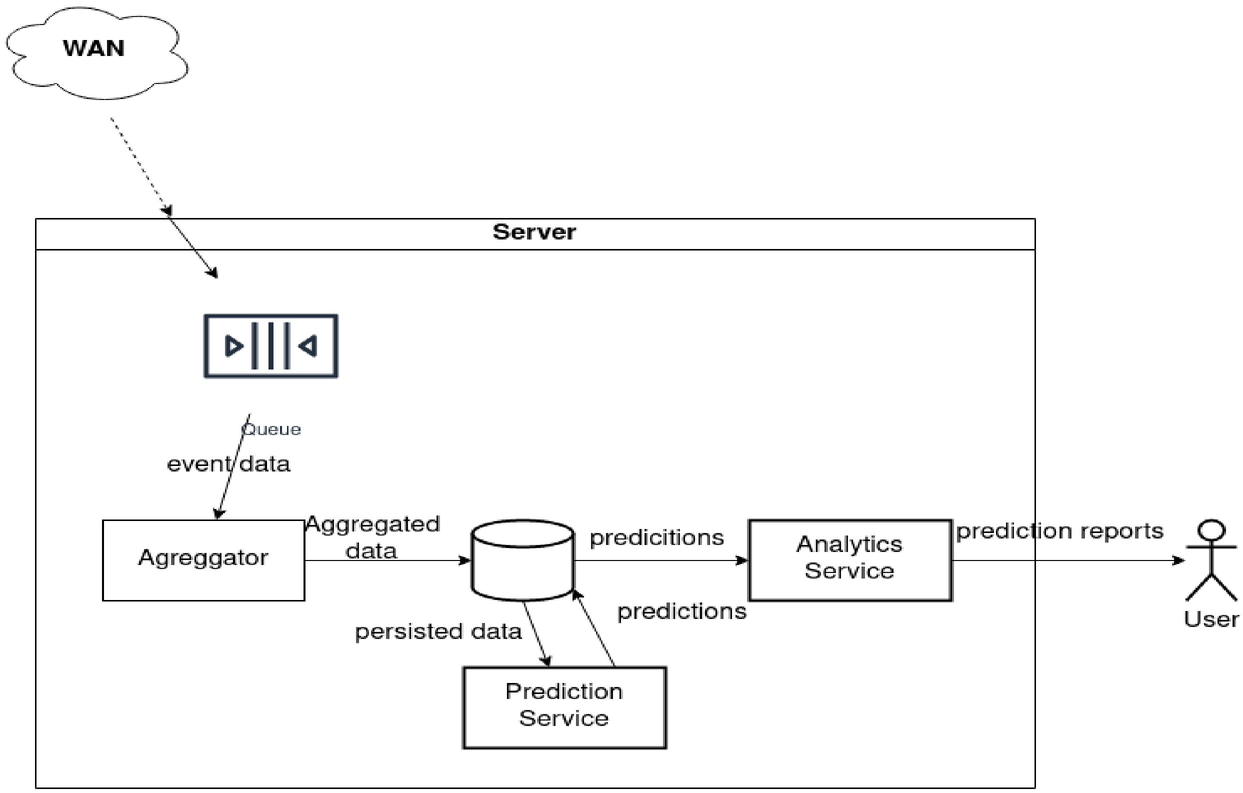

Figure 3: System architecture of the server

2.2 Tools & Methods

To implement the prediction server described above, a combination of software tools and technologies can be used. Some possible tools and technologies include:

- Python programming language: Python is a popular language for data analysis and machine learning, and is often used in developing predictive models.
- TensorFlow or PyTorch: These are popular machine learning frameworks that can be used to develop predictive models.
- Flask or Django: These are web application frameworks that can be used to build REST APIs for accessing data and serving predictions.
- PostgreSQL or MongoDB: These are popular databases that can be used to store data and predictions.

- RabbitMQ or Apache Kafka: These are message brokers that can be used to exchange messages between different components of the data flow.
- Docker or Kubernetes: These are containerization platforms that can be used to package and deploy the different components of the data flow.

By combining these tools and technologies, it is possible to build a scalable and robust data flow that includes a prediction server for generating predictions of energy production.

2.2.1 AutoML

Azure Automated Machine Learning (AutoML) is a tool that allows data scientists, analysts, and developers to build machine learning models with high scale, efficiency, and productivity while sus- taining model quality. It automates the time-consuming and iterative tasks of machine learning model development using breakthrough research from Microsoft Research division 1.

With Azure AutoML, one can automatically build and deploy predictive models using either the no-code UI or the SDK. It supports a variety of automated machine learning tasks such as classifica- tion, regression, time-series forecasting, text classification, multilabel text classification, named entity recognition, image classification, object detection, and image segmentation 1.

Azure AutoML also provides built-in capabilities such as easy data exploration, automatic prepro- cessing, and intelligent feature engineering using deep neural networks to handle large datasets and improve model scores. It also offers text data featurization in 100 languages with the included BERT deep-learning architecture 1.

One can use Azure AutoML with multiple Microsoft products for faster insights regardless of machine learning skill level. It also provides built-in support for experiment run summaries and detailed metrics visualizations to better understand models and compare model performance. one can use model interpretability to evaluate ML model fit for raw and engineered features and to get insights into feature importance 1.

Overall, Azure AutoML is a powerful tool that can help to accelerate model creation and improve productivity while building highly accurate machine learning models.

2.2.2 Prophet

Prophet is a forecasting procedure implemented in R and Python. It was developed by Facebook's Core Data Science team and is now open-source software available for download on CRAN and PyPI 1.

Prophet is based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data, shifts in the trend, and typically handles outliers well 1.

Prophet is fast and provides completely automated forecasts that can be tuned by hand by data scientists and analysts. It allows to generate reliable forecasts for planning and goal setting quickly. It can also use human-interpretable parameters to improve ther forecast by adding ther domain knowl- edge 1.

Overall, Prophet is a powerful tool for producing high-quality forecasts for time series data.

2.2.3 XGBoost

XGBoost (eXtreme Gradient Boosting) is an open-source software library that provides a regularizing gradient boosting framework for various programming languages such as C++, Java, Python, R, Julia, Perl, and Scala. It works on Linux, Windows, and macOS 1.

XGBoost is designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework and provides a parallel tree boosting (also known as GBDT, GBM) that can solve many data science problems in a fast and accurate way 2.

XGBoost has gained much popularity and attention recently as the algorithm of choice for many winning teams of machine learning competitions. It can be used on a single machine as well as on distributed processing frameworks such as Apache Hadoop, Apache Spark, Apache Flink, and Dask 1.

Overall, XGBoost is a powerful tool for building highly accurate machine learning models using gradient boosting techniques.

2.2.4 OpenWeatherApi

One free weather forecasting API that can be integrated into the current prediction system is Open- WeatherMap.

OpenWeatherMap provides current weather data, hourly and daily forecasts, and historical weather data for various locations worldwide. The API can be accessed using a simple HTTP request and can return data in various formats, including JSON, XML, and HTML.

To integrate OpenWeatherMap into the current prediction system, the prediction server can make periodic API requests to retrieve the current weather data and forecast data for the relevant locations. This data can then be combined with the sensor data from the inverters and fed into the machine learning algorithms to generate energy production predictions.

The API request can be made using a client library, such as Requests for Python, which can simplify the process of making HTTP requests and parsing the API response. The client library can be integrated into the prediction server code, which can periodically call the API and retrieve the latest weather data.

Additionally, OpenWeatherMap provides various tools and features that can enhance the prediction system, such as historical weather data analysis, weather alerts, and satellite imagery. These features can be integrated into the analytics service to provide users with a more comprehensive and accurate energy production prediction system.

Overall, by integrating a free weather forecasting API like OpenWeatherMap, the prediction system can improve its accuracy and reliability, leading to better energy production optimization and resource allocation.

2.2.5 AWS Greengrass

AWS Greengrass is an edge computing solution provided by Amazon Web Services (AWS). It enables IoT devices to run AWS Lambda functions and communicate with the AWS cloud even when they are disconnected from the internet.

With AWS Greengrass, the prediction service and IoT system can benefit in several ways:

- Reduced Latency: AWS Greengrass enables the prediction service to process data on the edge, reducing the time required to send data to the cloud and receive a response. This reduced latency can be especially important in cases where real-time decisions need to be made based on sensor data.
- Improved Reliability: By allowing IoT devices to operate locally even when they are disconnected from the cloud, AWS Greengrass can improve the reliability of the system. This means that the prediction service can continue to function even in situations where there is limited or no connectivity to the cloud.
- Enhanced Security: AWS Greengrass can improve the security of the IoT system by allowing data to be processed and analyzed locally on the edge, rather than being sent to the cloud for processing. This can reduce the risk of sensitive data being intercepted during transmission.

- Improved Scalability: AWS Greengrass can help the prediction service to scale more efficiently by allowing Lambda functions to be run locally on IoT devices, reducing the load on the cloud.
- Reduced Costs: By processing data on the edge, AWS Greengrass can reduce the amount of data that needs to be sent to the cloud, resulting in lower data transfer costs.

In summary, AWS Greengrass can help to improve the prediction service and IoT system by reducing latency, improving reliability, enhancing security, improving scalability, and reducing costs.

2.2.6 Grafana

Grafana is an open-source platform used for data visualization and monitoring. It can help improve the IoT system and prediction service by providing a unified dashboard for displaying real-time data and alerts from various sources.

Grafana supports a wide range of data sources, including time-series databases, relational databases, and cloud-based platforms. It also provides a variety of visualization options, such as graphs, charts, and tables, that can be customized to display data in a way that is easy to understand.

Using Grafana, the user can create custom dashboards that display real-time data from various sources, including the prediction service and IoT devices. The dashboards can be configured to display alerts and notifications based on predefined thresholds or rules, providing a way for the user to proactively monitor the system.

Furthermore, Grafana supports the integration of machine learning models, which can be used to further enhance the prediction service. The user can train machine learning models using historical data and then integrate them with Grafana to provide real-time predictions and insights.

Overall, Grafana is a powerful tool for monitoring and visualizing data from IoT devices and prediction services. It can help improve the system by providing a unified dashboard for real-time data visualization, alerting, and machine learning integration.

3. IMPLEMENTATION

3.1 Device Abstraction

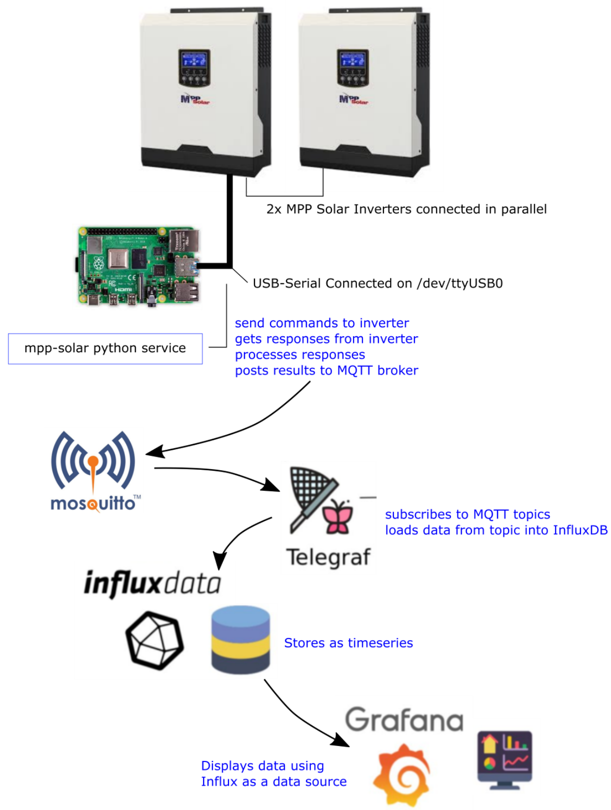

Figure 4: System data flow

The integration of MPP Solar inverters with a Raspberry Pi (RPi) begins with the physical hardware setup. The MPP Solar inverters are installed and connected to the solar panels according to the manufacturer's instructions. The RPi is then connected to the MPP Solar inverter using a suitable interface such as a USB cable or RS485 communication. It is essential to ensure that the RPi is properly powered and connected to the internet.

Once the hardware is in place, the software setup is initiated. The RPi is configured with an appropriate operating system like Raspbian or Ubuntu. Next, the necessary dependencies and libraries for the data collection program are installed. It is advised to refer to the README file of the provided GitHub repository for specific instructions on the required packages and installations.

To access the program responsible for gathering data from the MPP Solar inverter, the repository from the provided GitHub link is cloned. This is done by opening the terminal on the RPi and executing the command ”git clone https://github.com/jblance/mpp-solar.git”.

For enhanced portability and ease of deployment, the ”mpp-solar” service can be containerized using Docker. Docker provides a lightweight, isolated environment for running applications. The Docker image is built based on the provided Dockerfile, which encapsulates the dependencies and configurations required by the ”mpp-solar” service. By utilizing Docker, the deployment process is simplified, and the service can be easily replicated across different environments.

The MQTT (Message Queuing Telemetry Transport) protocol is employed to transmit data from the RPi to the server. To facilitate this, an MQTT broker must be set up on the server side. Pop- ular MQTT broker software, such as Mosquitto, can be used for this purpose. The MQTT broker is configured with appropriate credentials, including a username and password, as well as network settings.

With the configuration in place, the program is executed on the RPi to initiate data collection and MQTT communication. The command ”python3 mpp-solar.py” is executed in the terminal, navigating to the repository folder. This command establishes communication with the MPP Solar inverter and starts collecting relevant data points. The program formats the collected data into MQTT messages and publishes them to the configured MQTT broker

3.2 Network Interconnection Cloud

This chapter focuses on the network interconnection and security aspects of the MPP Solar data collection system. It describes the communication between the client (mpp-solar service) and the server (AWS ECS) via Ethernet using the MQTT protocol. Additionally, it highlights the deployment of the Mosquitto MQTT broker on AWS and the security measures implemented using the AWS ECS firewall.

The communication between the MPP Solar inverters and the Raspberry Pi (client) is established over Ethernet. Ethernet provides a reliable and high-speed connection, allowing for efficient data transmission between the devices. The Raspberry Pi is connected to the local network, enabling seamless interaction with the MPP Solar inverters.

To establish the connection between the RPi and the MQTT broker, the program's configuration must be adjusted. The "config.json" file within the cloned repository folder is modified to include the necessary information for connecting to the MQTT broker. This includes specifying the IP address, port number, username, and password. Other configuration parameters can be updated as per the specific requirements of the system.

To facilitate MQTT communication, the Mosquitto MQTT broker is deployed on AWS. Mosquitto is an open-source MQTT broker that can be installed and configured on AWS infrastructure. The Mosquitto broker is responsible for receiving MQTT messages published by the client (mpp-solar service) and forwarding them to the appropriate subscribers on the server side.

To ensure the security of the communication, the AWS ECS (Elastic Container Service) firewall is employed. AWS ECS provides a secure and scalable platform for hosting the server components of the MPP Solar data collection system. The ECS firewall allows fine-grained control over inbound and outbound traffic, enabling secure communication between the client and the server.

The secure communication flow between the client and the server can be summarized as follows:

- The client (mpp-solar service) establishes an Ethernet connection with the MPP Solar inverters, collecting relevant data.
- The client formats the collected data into MQTT messages and publishes them to the Mosquitto MQTT broker deployed on AWS.

- The server (AWS ECS) subscribes to the MQTT topics and receives the published data from the Mosquitto broker.
- The ECS firewall ensures that only authorized communication is allowed between the client and the server, securing the data transmission.

To further enhance security, data encryption and authentication mechanisms can be implemented. TLS (Transport Layer Security) can be employed to encrypt the MQTT communication between the client and the server. Client authentication using certificates or username/password credentials can be enforced, ensuring that only authorized clients can publish data to the MQTT broker.

In conclusion, this chapter discussed the network interconnection and security aspects of the MPP Solar data collection system. It highlighted the use of Ethernet for communication between the client and the server, and the MQTT protocol for efficient data transmission. The deployment of the Mosquitto MQTT broker on AWS and the security measures implemented with the AWS ECS firewall were described. Additionally, it emphasized the importance of data encryption, authentication, and monitoring to ensure a secure and reliable data collection system.

3.3 Database

This chapter focuses on the database integration and data storage aspects of the MPP Solar data collection system. It describes the deployment of the Telegraf service and InfluxDB database on the AWS server. The chapter also discusses how Telegraf collects data from Mosquitto and sends it to InfluxDB for storage. Additionally, it highlights the advantages of using InfluxDB in IoT systems.

Telegraf is a plugin-driven server agent that is deployed on the AWS server. It is responsible for collecting data from various sources, including the Mosquitto MQTT broker. Telegraf can be configured to subscribe to specific MQTT topics and extract the relevant data. It acts as a bridge between the MQTT broker and InfluxDB, ensuring seamless data ingestion.

InfluxDB is a high-performance, open-source time-series database designed for efficiently storing and querying time-stamped data. It is well-suited for IoT systems, including the MPP Solar data collection system, due to its ability to handle large volumes of data and provide fast and flexible querying capabilities.

The workflow for data collection and storage can be summarized as follows:

- Telegraf, running as a separate container on the AWS server, subscribes to the MQTT topics and receives the published data from the Mosquitto broker.
- Telegraf processes the received data and converts it into the InfluxDB line protocol format. The processed data is then sent to InfluxDB for storage, where it is organized based on the defined measurements, tags, and fields.
- InfluxDB stores the time-series data efficiently, allowing for quick retrieval and analysis.

InfluxDB offers several advantages for IoT systems like the MPP Solar data collection system:

- Time-Series Data Handling: InfluxDB is specifically designed to handle time-stamped data, making it ideal for capturing and storing real-time data from IoT devices, including energy production from the inverters.
- High Performance: InfluxDB is optimized for high write and query performance, enabling effi- cient storage and retrieval of large volumes of data.
- Scalability: InfluxDB can scale horizontally to handle increasing data loads, allowing the system to accommodate future growth in the number of connected devices and data points.
- Flexible Data Model: InfluxDB's flexible data model allows for the storage of different types of data, including real-time sensor readings, predicted energy production, and other relevant metrics.
- SQL-Like Query Language: InfluxDB provides a SQL-like query language (InfluxQL) and a powerful query API for retrieving and analyzing data. This facilitates easy integration with visualization tools like Grafana and prediction services.

InfluxDB stores the collected data from the inverters, which can be visualized in real-time using Grafana. Grafana is a popular open-source data visualization tool that integrates seamlessly with InfluxDB. It allows users to create customizable dashboards and visualizations, enabling real-time monitoring of energy production and other system metrics.

In addition to real-time data visualization, the stored data in InfluxDB is utilized by the prediction service to display predicted energy production by the inverters. By analyzing historical data and ap- plying machine learning algorithms, the prediction service forecasts energy production trends, helping in optimizing energy usage and planning.

3.4 Machine Learning Forecasting Model

Since machine learning algorithms use historical data as input to predict new output values[2], a research was conducted on data available online. Upon discovery, two of the many choices were selected for training: an data set that would help predict the production of solar energy depending on the climacteric values and another one that would help forecast the production based three previous years of historic data. Furthermore, the data sets will be analysed individually and conclusions will be drawn accordingly.

First thing was analyzed the correlation between different features from the data. Since they were quite many, sixteen to be exact, a feature selection algorithm was needed. The kydavra library was used, since its a python sci-kit learn inspired package for feature selection.[3] From the library, P Value Selector proved to be the best one both in speed and efficiency, reducing the number of features to seven. In the analysis process of the final seven features, these three were selected:

- Maximum Temperature, having a correlation of 0.36;
- Precipitation, having a correlation of -0.42;
- Cloud Cover, having a correlation of -0.74.

Those values mean that the discharge of energy from the solar panels are closely related to the maxim temperature and precipitation level outside, as well as the cloud coverage from the sky. Hence, their value will help the model the most in predicting the value of generated energy.

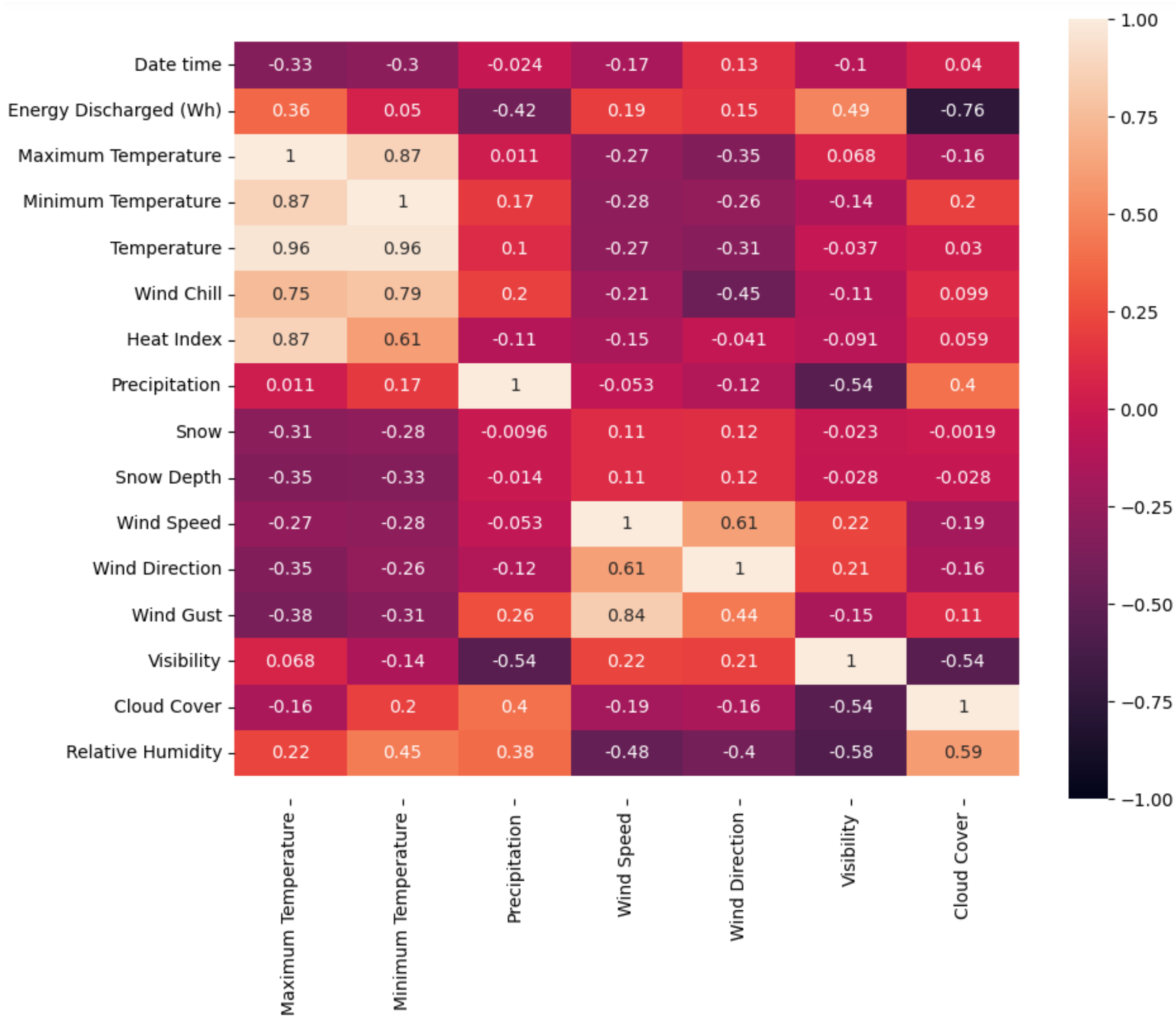

Figure 5: Correlation heatmap of features versus target column

To have a balanced model it is needed to train it on balanced data. Hence, the data balance analysis in fig. 6. Balance in Data Science means Gaussian Normal Form, which should have more average data and less exceptional data on either negative part or positive one. Tho slightly skewed and having a lot of spikes in data distribution. the features tend to reassemble somewhat the normal distribution. Of course, real-life doesn't have normal distribution as we imagine it, hence the form of data is more than acceptable.

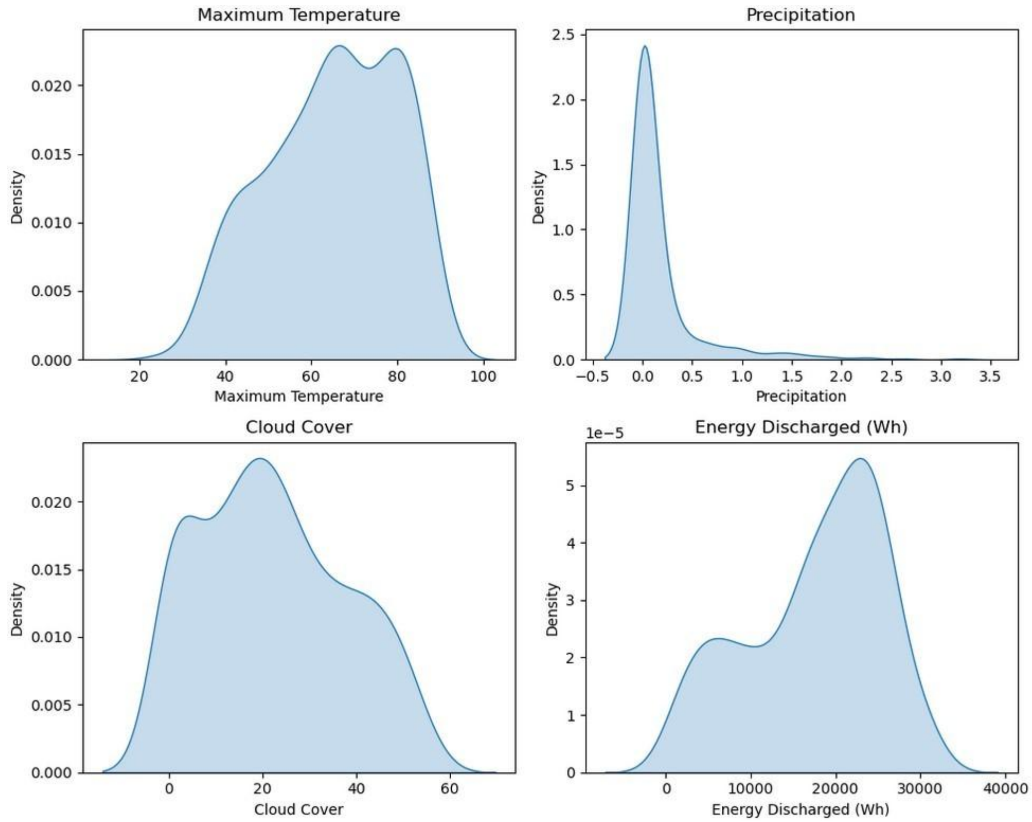

Figure 6: The Distribution of Data

In order to make an informed conclusion on whether the data set is good or not for our scope, the analysis of energy production in time is needed. The scope is to see if the distribution follows a logical explanation. The graphic provided in Fig. 7 is the result of this inquiry.

It oscillates in a normal way since at night there is minimal to no energy production and during the day the values are high, especially around noon. As it also can be understood, the energy production in May-July is higher than in November-January period, since the sun exposure in winter is lower than summer.

In conclusion the data indeed follows a logical explanation and can be used for our model. After training and testing, the Random Forest Regressor model proved to be the most efficient algorithm.

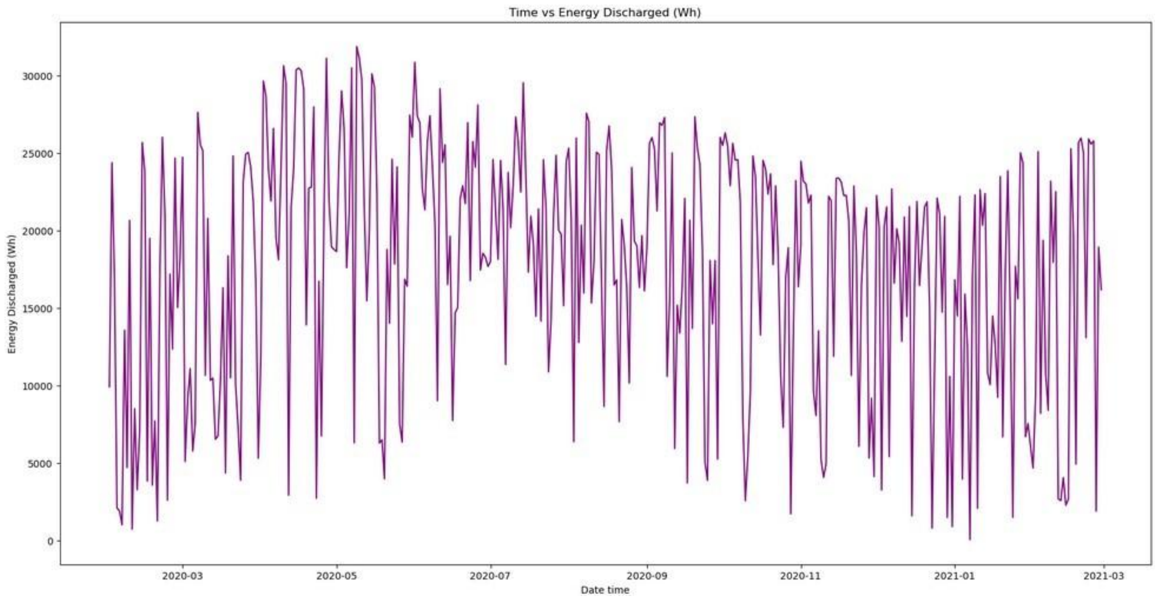

Figure 7: The Distribution of Data

Random forest is a type of supervised learning algorithm that uses ensemble methods (bagging) to solve both regression and classification problems. The algorithm operates by constructing a multitude of decision trees at training time and outputting the mean/mode of prediction of the individual trees. The fundamental concept behind random forest is the wisdom of crowds wherein a large number of uncorrelated models operating as a committee will outperform any of the individual constituent models. The reason behind this is the fact that the trees protect each other from their individual errors. Within a random forest, there is no interaction between the individual trees. A random forest acts as an estimator algorithm that aggregates the result of many decision trees and then outputs the most optimal result. [4]

The second data set is way simpler than the first, since a forecasting model needs nothing more than the date and the production column. The data starts from may 2019 and goes till may 2023, showcasing data on the length of four years.

Unlike in the case of tabular data, time series data are characterized by their seasonality and trend. Seasonality means the period at which the data is behaving in the same manner over and over again. In the case of solar energy production, the seasonality expected should be yearly, since each year summer produces higher energy and winter produces lower.

The trend characterizes the behaviour of the data either rising or decreasing. It should be expected that the trend for the month of may is in raise.

To perform those analytic steps, the kats library was used. Kats is a toolkit to analyze time series data, a lightweight, easy-to-use, and generalizable framework to perform time series analysis. Time series analysis is an essential component of Data Science and Engineering work at industry, from understanding the key statistics and characteristics, detecting regressions and anomalies, to forecasting future trends. [5]

In Fig. 8 is displayed the seasonality and trending of the data set chosen. As expected, the seasonality of the data set is one year, since the pattern is repeating itself from the first month of the next year. Also, the trending is rising constantly, which can be motivated by the global warming that raises the exposure of earth to the sun, making it favourable for solar energy production.

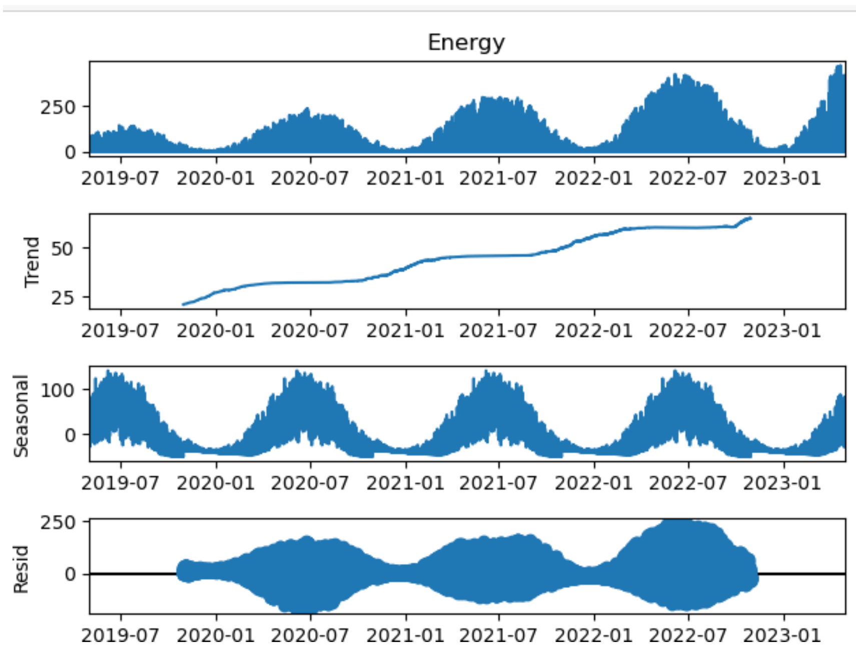

Figure 8: Seasonality of data

After experimenting with several forecasting models from the kats library, the prophet algorithm showcased the best results of 83% accuracy, hence being chosen for the final forecasting feature of the system.

3.5 Backend

The back end of the application consists of a server written in python using the Flask library and appropriate Data Science libraries, such as pandas, pickle and others.

The server has several API's that retrieve data either from the cloud (real data from the invertor), from the trained models for prediction and from the weather API.

The following list represents the available APIs and their functionality:

- /weather?latitude=latitudelongitude=longitude - this endpoint activates a function that re- trieves data from the weather API about the weather for the first seven days ahead;
- /weekly_prediction - this endpoint calls a function that predicts the production of energy in the following 7 days and returns the values for each day;
- /hourly_prediction - this endpoint calls a function that forecasts the production of energy for every hour in the following 7 days;
- /last_7days_wd - this endpoint retrieves from the cloud the data regarding the daily energy production of the system for the last 7 days;
- /last_7days_wh - this endpoint retrieves from the cloud the data regarding the hourly energy production of the system for the last 7 days;
- /hourly_predict.csv - this endpoint downloads the .csv file with the forecast of energy production until the year of 2024.

These endpoints were used by the frontend to retrieve needed data for display and functionality.

3.6 Frontend

React framework was chosen as the tool for creating the graphical user interface. This choice was driven by the large and active community of React users, as well as several key features that make it one of the top options for deploying an artificial intelligence model, such as:

- Component-based architecture. It makes it easier to manage and maintain the code base.
- Efficient rendering. React updates only the necessary components in the program instead of the whole app when changes occur in the model or other parts of the UI. This improves the performance and makes the user experience a lot smoother.
- State management. Useful when working with an AI model which requires real-time updates.
- Integration with third-party libraries. Easy to integrate with other API's for data visualisation or for showing the weather data on screen.

ECharts is a powerful charting library which was used for data visualisation. It supports a wide range of chart types, including line charts, bar charts, pie charts, scatter plots, and more. In this project, both charts are of type Stacked Line Chart which is great for comparing actual data vs. predicted data. It alows to view the exact value of data at a selected moment, to show one or two lines at a time and to download the chart on the user's computer.

Figure 9: User Interface

In order to showcase the correlation between weather data and predictions, WeatherAPI was used as the API to transmit the weather data to the user. On the left part of our site the user cand see weather data, including the date, condition, an icon, temperature, humidity and predicted energy production.

CONCLUSION

The Solar Sense prediction system presented in this technical paper demonstrates a comprehensive solution for monitoring and predicting solar power production. By integrating various components and leveraging the capabilities of cloud, the system enables real-time data collection, secure communication, efficient storage, and insightful visualization.

The solution represented in the paper, which encompasses solar production monitoring, analysis, and prediction, can significantly enhance the efficiency of energy production from solar systems. By enabling real-time monitoring, performance analysis, and predictive analytics, the system empowers operators to take proactive measures in optimizing energy generation and system performance.

Through continuous monitoring, anomalies or issues that may hinder energy production can be detected early, minimizing production losses and maximizing overall system efficiency. The in-depth analysis of performance data helps identify areas for improvement, such as module degradation, shad- ing, or system malfunctions, allowing operators to implement targeted actions to optimize energy generation.

The predictive analytics aspect of the solution leverages historical data and advanced algorithms to forecast future energy production. This enables better planning, resource allocation, and integration with the grid. By considering factors like weather patterns and solar irradiance forecasts, operators can make informed decisions to maximize energy generation, utilize energy storage efficiently, and enhance grid integration.

Additionally, the solution facilitates proactive maintenance by identifying potential issues before they escalate. By scheduling maintenance activities in a timely manner, downtime can be minimized, maintenance costs can be reduced, and the overall performance of the solar energy system can be optimized.

Ultimately, the comprehensive monitoring, analysis, and prediction capabilities provided by the solution contribute to making energy production from solar systems more efficient. By leveraging data- driven insights and enabling proactive decision-making, the solution helps maximize energy output, improve system performance, and achieve higher levels of efficiency in solar energy production.

In conclusion, the Solar Sense prediction system presented in this paper offers a robust and scalable solution for prediction of solar power production. The system's architecture, based on the integration of various technologies and services, demonstrates the potential of leveraging IoT principles to create efficient, secure, and data-driven solutions for renewable energy monitoring and optimization.

REFERENCES

1. alexsoft. ”Non-functional Requirements: Examples, Types, How to Approach”. Accessed April 3, 2023. https://www.altexsoft.com/blog/non-functional-requirements/
2. Tech Target, Enterprise Ai. ”Definition machine learning”. Accessed May 27th, 2023. https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML
3. Python Package Index. ”kydavra 0.3.4”. Accessed May 28th, 2023. https://pypi.org/project/ kydavra/
4. Towards Data Science. ”A Quick and Dirty Guide to Random Forest Re- gression” Accessed May 29th, 2023. https://towardsdatascience.com/ a-quick-and-dirty-guide-to-random-forest-regression-52ca0af157f8
5. Facebook Research ”Kats python library” Accessed June 2nd, 2023.https:// facebookresearch.github.io/Kats/