ENERGY 2022

The Twelfth International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies

ISBN: 978-1-61208-967-6

May 22nd –26th, 2022

Venice, Italy

ENERGY 2022 Editors

Vivian Sultan, California State University Los Angeles, USA

Oana Dini, IARIA, USA

ENERGY 2022

Foreword

The Twelfth International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies (ENERGY 2022), held between May 22 – 26, 2022, continued the event considering Green approaches for Smart Grids and IT-aware technologies. It addressed fundamentals, technologies, hardware and software needed support, and applications and challenges.

There is a perceived need for a fundamental transformation in IP communications, energy-aware technologies and the way all energy sources are integrated. This is accelerated by the complexity of smart devices, the need for special interfaces for an easy and remote access, and the new achievements in energy production. Smart Grid technologies promote ways to enhance efficiency and reliability of the electric grid, while addressing increasing demand and incorporating more renewable and distributed electricity generation. The adoption of data centers, penetration of new energy resources, large dissemination of smart sensing and control devices, including smart home, and new vehicular energy approaches demand a new position for distributed communications, energy storage, and integration of various sources of energy.

We take here the opportunity to warmly thank all the members of the ENERGY 2022 Technical Program Committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to ENERGY 2022. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the ENERGY 2022 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that ENERGY 2022 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the fields of smart grids, green communications and IT energy-aware technologies.

We are convinced that the participants found the event useful and communications very open. We also hope that Venice provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

ENERGY 2022 Chairs:

ENERGY 2022 Steering Committee
Eric MSP Veith, OFFIS e.V. – Oldenburg, Germany
Dragan Obradovic, Siemens - Corporate Technology, Munich, Germany
Mark Apperley, University of Waikato, New Zealand
Michael Negnevitsky, University of Tasmania, Australia
Vivian Sultan, California State University Los Angeles, USA
Steffen Fries, Siemens, Germany

ENERGY 2022 Publicity Chairs
Hannah Russell, Universitat Politècnica de València (UPV), Spain
Mar Parra, Universitat Politecnica de Valencia, Spain

ENERGY 2022

Committee

ENERGY 2022 Steering Committee

Eric MSP Veith, OFFIS e.V. – Oldenburg, Germany
Dragan Obradovic, Siemens - Corporate Technology, Munich, Germany
Mark Apperley, University of Waikato, New Zealand
Michael Negnevitsky, University of Tasmania, Australia
Vivian Sultan, California State University Los Angeles, USA
Steffen Fries, Siemens, Germany

ENERGY 2022 Publicity Chairs

Hannah Russell, Universitat Politècnica de València (UPV), Spain
Mar Parra, Universitat Politecnica de Valencia, Spain

ENERGY 2022 Technical Program Committee

Santosh Aditham, University of South Florida, USA
Kodjo Agbossou, Université du Québec à Trois-Rivières, Canada
Miltos Alamaniotis, University of Texas at San Antonio, USA
Kamal Al-Haddad, École de technologie supérieure, Montreal, Canada
Magnus Almgren, Chalmers University of Technology, Sweden
Ahmed Al-Salaymeh, The University of Jordan, Amman, Jordan
Mark Apperley, University of Waikato, New Zealand
Paranietharan Arunagirinathan, Clemson University, USA
Ashrant Aryal, Texas A&M University, USA
Adela Bara, Bucharest University of Economic Studies, Department of Economic Informatics and Cybernetics, Bucharest, Romania
Navid Bayati, Aalborg University, Denmark
Andreas Berl, Technische Hochschule Deggendorf, Germany
Rico Berner, Humboldt-Universität zu Berlin | Institut für Physik, Germany
Lasse Berntzen, University of South-Eastern Norway, Norway
Vito Calderaro, University of Salerno, Italy
M. Girish Chandra, TCS Research & Innovation, India
Dana-Alexandra Ciupageanu, University Politehnica of Bucharest, Romania
Daniele Codetta, University of Piemonte Orientale, Italy
Luigi Costanzo, Università degli Studi della Campania Luigi Vanvitelli, Italy
Fabio D'Agostino, University of Genova, Italy
Thusitha Dayaratne, Monash University, Australia
Payman Dehghanian, Assistant Professor, The George Washington University, USA
Margot Deruyck, Universiteit Gent - IMEC - WAVES, Belgium
Giovanna Dondossola, RSE, Italy
Virgil Dumbrava, University POLITEHNICA of Bucharest, Romania

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

GIS Deep Learning For Power Assets' Detection and Identification

Vivian Sultan
California State University
Los Angeles, CA USA
email: vsultan3@calstatela.edu

Jose Ramirez
California State University
Los Angeles, CA USA
email: jramire3@calstatela.edu

Jordan Peabody
California State University
Los Angeles, CA USA
email: jpeabod@calstatela.edu

Madison Bautista
California State University
Los Angeles, CA USA
email: mbauti33@calstatela.edu

***Abstract*—In this study, ArcGIS Pro 2.8 identified power poles and towers from Light Detection And Ranging (LiDAR) point-cloud data. In previous research, machine learning has identified objects from such data. We sought to demonstrate a deep-learning model developed by the Environmental Systems Research Institute and a group based in Australia and whether deep learning is a viable solution for identifying power assets in three California areas. The deep-learning model was deployed in ArcGIS Pro using the Classify Point Cloud Using Trained Model geoprocessing tool. The model successfully identified some power poles in both rural and urban areas. A better training dataset might improve on this limited success, suggesting that deep learning can successfully classify point clouds. Those interested in using LiDAR point clouds with deep learning to classify power poles and towers should produce training data using accurately labeled data that accurately represents the objects of interest to ensure optimal results with a new model.**

Keywords—LiDAR; deep learning; point cloud; ArcGIS Pro; point classification.

I. INTRODUCTION

Light Detection And Ranging (LiDAR), a type of remote-sensing technology, uses pulsed lasers to measure variable distances, heights, or depths of objects and areas. LiDAR devices are generally mounted on Unmanned Aerial Vehicles (UAV). These UAVs are remotely operated to scan areas of interest. At a minimum, this process requires a two-person team to remotely operate the UAV and verify the data is correct [1]. This data can be input to software that can read the point-cloud data for further processing. UAVs and LiDAR data provide several benefits over sending people to physically inspect all assets of interest. For instance, a UAV can easily scan large areas without regard to the type of terrain (steep slopes, dense forests, etc.). Several studies have examined extracting objects from point-cloud data.

Based on the current and future potential of LiDAR data in assessing and managing forest structures, remote sensing and classification can identify specific trees in a cluster and more closely identify the species [2]. The article is relevant to our research question of whether LiDAR can be used to identify power poles and structures, which may be imbedded in forests or other rural areas.

3D LiDAR imaging could be used in conjunction with a Recurring Neural Networks (RNN) to identify different objects. With the progression of scanners, 3D LiDAR images provide enhanced measurement data [3]. The point spacing between objects could be leveraged to create a model to recognize objects [3]. The RNN model showed promise, and warrants further research, as does pursuing better 3D data.

Using Convolutional Neural Networks (CNNs), Maggiori et al. [4] created an end-to-end satellite-imagery classification framework, noting its ability to classify satellite images, identify objects, and produce quality images. Untrained models performed less well. From an existing model, they constructed a manually classified dataset with significant improvements. They propose a two-step approach with a small manually classified training dataset to classify a larger unclassified set.

Kudinov [5] used the Point Convolution Neural Network (PointCNN) framework to automatically identify power lines and poles. The group used artificial intelligence for the labor-intensive task of manually labeling the point cloud. Their study area was a city in Australia, and their dataset contained 540 million points. They trained their PointCNN model using four classes: other, wires, stray wires, and utility poles.

Studying the You Only Look Once (YOLO) deep-learning algorithm, Fan et al. [6] detect objects in point-cloud datasets for self-driving vehicles, which need real-time information to avoid collisions. Consequently, the researchers propose a computationally efficient algorithm, LS-R-YOLOv4, using color images and point-cloud data to precisely segment and detect objects. Borcs et al. [7] proposed a pipeline to quickly classify point clouds. One component of this pipeline is a CNN trained to classify objects.

LiDAR data can be used to accurately pinpoint large areas' micromorphology [8]. Comparing a Digital Elevation Model (DEM) generated from LiDAR data to the surveyed plots, Brubaker et al. [8] learned that their research was accurate to within 0.3–0.4 m of the actual survey, a single point in the point cloud. With their data, they generated the surface constraint of the surveyed area faster and from a greater distance compared to a traditional survey. The DEM is important as it allows LiDAR data to be accurately separated from ground, water, or any surface constraints based on elevation.

Azevedo et al. [9] showcased UAVs to replace helicopters. UAVs with LiDAR would help companies maintain their equipment more efficiently with a team of a few people for data vetting and to control the UAV. LiDAR data can be converted to point-cloud data and fed through an algorithm to help identify and sort items in the LiDAR data. They argue that, while the algorithm they used failed to correctly identify possible points, those points were classified as unidentified due to the difficulty of differentiating vegetation and other objects. They conclude that a more powerful algorithm may correctly identify points of interest and that Graphics Processing Units (GPUs) would process faster.

 ISBN: 978-1-61208-967-6

The CNN algorithm was able to transform, organize, and label data [10]. With orthophotos and LiDAR data, Nahhas et al. [10] created a digital surface model, DEM, shapes, and input other data through the model to detect buildings. From their findings and experiments, the CNN and machine-learning model accurately classified background and buildings and drew the geometry and shapes of the building from the LiDAR and orthophotos. They were able to transform low-level detail into highly detailed, classified features.

Section II describes the materials and methods used in the study. Section III presents the results and discusses the findings. Section IV concludes.

II. MATERIALS AND METHODS

In this section, we define the problem, outline the data selection and acquisition, and lay out the methodology.

A. Problem Definition

Utility companies must carefully manage their assets against various hazards, visually inspect their assets regularly. Generally, these assets require many hours to manage and are not easily accessible due to their remoteness. Dispatching teams of people to assess the status of power equipment is expensive and time consuming. LiDAR data serves as a cost-efficient alternative for surveying large areas of land and generating real-time images of objects.

The point-cloud data generated by scans can be analyzed to identify assets needing maintenance. Additionally, utility companies can lower labor and transportation costs as there is no need to send maintenance crews into the field.

The cost of LiDAR depends on the type of equipment and the range and scope of work [11]. LiDAR drones can potentially be cost effective for difficult-to-reach forested areas, rural towns, or high-elevation areas. LiDAR can also be used in high-density areas such as urban or suburban areas [12]. The high upfront cost leaves just maintenance of the equipment, future upgrades, and pilot licensing as needed [13]. These costs can be calculated in advance, while the ongoing costs of dispatching workers depends on the scope of work and may not be easily estimated [14]. In many cases, contractors may be required in difficult-to-reach areas and may not have the quality control utility companies need.

While manually assessing and inspecting equipment is beneficial as the information about them can be updated in real time, high-scale LiDAR data must be processed and analyzed to ensure error-free data [9][10].

LiDAR technology provides several benefits when surveying objects. Therefore, this study sought to answer the following question. Can utilities use LiDAR point-cloud data to accurately define asset locations (poles and towers)?

The literature suggests deep learning can be used to classify objects of interest. This study deployed a deep-learning model to classify points of interest (utility poles). In addition, we gauged the effectiveness of other ArcGIS Pro classification tools at classifying poles and towers. This study may be of interest to utility companies and individuals interested in using LiDAR to manage assets.

B. Data Selection and Acquisition

Using publicly available point-cloud data from the United States Geological Survey website, we explored datasets from June 19, 2018, for parts of Santa Cruz, West Hollywood, and North Long Beach (>30 million points and 1.5 GB each) to study a PointCNN deep-learning model's ability to correctly classify power poles. The deep-learning model employed came from ArcGIS Online.

The following steps describe the point-cloud download process. (1) Navigate to the national US Geological Survey map [15]. (2) Select "Show where Lidar is available" and "Show AOI [Area Of Interest] Results" in the left pane. The latter may only show after an area of interest is defined. (3) Select the area of interest by Ctrl-clicking and dragging. (4) A results pane will appear on the right. Click "Lidar within AOI" for the download files. Note that all steps are for Windows users. Mac steps are similar.

C. Methodology

This research explored ArcGIS geoprocessing tools, including a deep-learning model and additional tools. We classify the tools into three categories: (a) data conversion, (b) deep learning, and (c) .LAS conversion and discuss deploying a deep-learning model to classify point-cloud data.

1. Data Conversion and Projection

The data files come from USGS in the .LAZ format. To use these files in ArcGIS Pro, they must be decompressed. The first step is to convert them to .LAS using the open-source laszip.exe executed at the command prompt: (1) Download the conversion tool [16]. Place the downloaded file in a folder with the .LAZ files. (2) Bring up the Command Prompt (Windows), allowing file access as necessary. (3) Type cd /d. (4) Copy the location of the .LAZ files from the file manager, paste it on the command line, and press Enter. (5) Copy the command **laszip.exe *.laz** to the command prompt and press Enter. (Note: pressing Enter will start the tool converting the .LAZ files to .LAS files. The tool must be located in the same folder as the .LAZ files, otherwise it will not find them.) The converted files will appear in the folder with the .LAZ files.

ArcGIS Pro requires a .LAS dataset to manipulate the point-cloud data. In addition, the deep-learning model's documentation requires the point cloud's *x*, *y*, and *z* coordinates be in metric. The USGS .LAS datasets, by default, are in imperial units. ArcGIS's "Create LAS Dataset" can perform this conversion: (1) Navigate to the Geoprocessing Tools pane in ArcGIS Pro and find "Create LAS Dataset." Click on the folder icon under Input Files and locate the .LAS files decompressed earlier. Under "Create PRJ For LAS Files," select "All LAS Files." (2) Click on the globe icon under Coordinate System. (3) A separate window will open for setting coordinate systems. Click on Current XY and find NAD 1983 NSRS2007 California (Teale) Albers (Meters) or whatever is appropriate for your dataset. It is found by expanding Projected Coordinate System – State Systems. (4) Click Current Z and navigate to "NAD 1983 (NSRS2007)" found by expanding Vertical Coordinate

 ISBN: 978-1-61208-967-6

System – Ellipsoidal-based – North America. Once the *X, Y* and *Z* coordinates are correctly specified, click OK.

2. Deep-Learning Tools

Our main approach involved using a publicly available PointCNN deep-learning model to automatically classify power poles and towers. Due to resource and time constraints, this study instantiated an existing trained model to determine whether deep learning is an effective solution for extracting the desired objects from point-cloud data.

ArcGIS Pro provides three tools to classify data, train a model, and use a model: Prepare Point Cloud Training Data, Train Point Cloud Classification Model, and Classify Point Cloud Using Trained Model

This project employed the Classify Point Cloud Using Trained Model (CPCWTM) geoprocessing tool to run the trained model on the .LAS datasets. To run deep-learning models in ArcGIS, one must install the Deep Learning Framework for ArcGIS Pro 2.8. (1) Download and install required framework to run deep-learning models on ArcGIS [17]. At that link, click Deep Learning Libraries Installer for ArcGIS Pro 2.8 in the Download section. Unzip and run the executable. (2) Download the deep-learning model to classify power lines [18]. The tool is also on ArcGIS online through the ArcGIS Catalog Pane. You will need to store the deep-learning model in your project folder. (3) Open the CPCWTM geoprocessing tool. Enter your .LAS dataset under *Target Point Cloud*. Under Input Model Definition, locate the downloaded deep-learning model from the folder icon. Under *Existing Class Code Heading*, select Edit Selected Points. Under *Existing Class Codes*, select "1" to run the model on unclassified points. With all parameters entered, click Run. The required time to complete depends upon the size of the dataset and the computer's resources. The model may take several hours to classify the point-cloud data. (4) Filter the layer to display only power poles: Select the .LAS dataset map layer in the Contents plane. On the Appearance tab of the ribbon click "LAS Data Points." Turn off all but classification code 15—transmission tower—by unchecking the other checkboxes.

3. .LAS Classification Tools

We also evaluated whether ArcGIS's .LAS classification tools in Geoprocessing Tools support classifying power poles and towers: "Classify LAS Ground," "Classify LAS Building," "Classify LAS by Height," "Classify LAS Noise," and "Change LAS Classification Codes."

The point-cloud data was classified using the above tools prior to running the model without improvement. In addition, the "Classify LAS by Height" tool helped determine if poles could be classified by their height. The tool proved ineffective at classifying poles as it only considers the height of the point not its other attributes or its relation to neighboring points.

III. Results and Discussion of Findings

In this section, we present the results and discuss the findings.

A. Results

The CPCWTM geoprocessing tool used with the PointCNN deep-learning model successfully classified point-cloud data points as power poles and towers. The model was able to identify power poles and towers in the Santa Cruz Mountains, West Hollywood, and Long Beach (Figures 1–3).

While the tools achieved the objective of identifying power poles and towers, they could have performed better. They performed better in urban areas than in rural areas, but it had difficulty finding most of the power poles and towers within the datasets examined. In the Santa Cruz dataset, only seven power poles were identified. In the North Long Beach dataset, several poles were identified but not in their entirety, and some poles were not classified at all. The West Hollywood dataset gave similar results. Further, the deep-learning model did not identify any power poles or towers in the densely forested parts of the Santa Cruz Mountains.

Figure 1. Scotts Valley, CA (Santa Cruz County).

Figure 2. North Long Beach, CA.

Figure 3. West Hollywood, CA.

The remote-sensing classification tools did not increase model performance compared to unclassified data. The model failed to identify many power poles and towers.

B. Discussion of Findings

Our project produced several findings. Troubleshooting the .LAZ–LAS file conversion led us to discover a third-party tool for converting .LAZ files, discussed in an earlier section. Using the tool on the command prompt, our group converted multiple files at one time for greater efficiency.

The deep-learning model required the dataset's projected coordinate system in the metric system. We used the "Classify LAS Dataset" geoprocessing tool to change the coordinate system. Some group members experienced issues

 ISBN: 978-1-61208-967-6

with plotting the data. More than once, the point cloud was plotted in the middle of the Atlantic Ocean. Each time this issue surfaced, team members repeated the first steps of the workflow. Importing newly converted data solved the problem each time, and the data was plotted correctly.

Running the CPCWTM geoprocessing tool resulted in key findings on converting data from imperial to metric measures. Finding a method to convert both the XY coordinates and the Z coordinates took a combined 20 hours. ArcGIS contains many coordinate systems with limited explanations of each system.

Using the PointCNN trained deep-learning model led us to several key findings. When running the model more than once, we found that ArcGIS occasionally did not always accept the parameters we entered. In one case, ArcGIS would not recognize where the downloaded tool was kept. Closing the project and creating a new one fixed the issue.

Another issue we faced was the inability to run the deep-learning model. Although the correct path to the deep-learning model was input from both the local file directory and ArcGIS Online, the CPCWTM tool could not find it. After several hours of troubleshooting, the issue was resolved by reinstalling ArcGIS pro on all desktop profiles.

The different members' varying computer hardware and processing capabilities was challenging. Recommended hardware for using ArcGIS Pro is a four-core central processing unit, with two cores at a minimum and 10 cores optimally. Devices with lower specifications had some issues, such as freezing and becoming unresponsive. The datasets in the research ranged from 1.5 to 3.2 gigabytes, and the group noticed that ArcGIS Pro to function with fewer issues with less data processed. Additionally, some troubleshooting was required for some users to get the machine-learning model to function. For example, ArcGIS Pro must be installed for all users or there will be an issue with permissions.

IV. CONCLUSIONS

Utilities must manage assets for reliable service and to protect property and life. Asset inspections are expensive, labor intensive, and difficult in hard-to-reach locations. LiDAR can efficiently scan areas and generate point-cloud data that can be efficiently and cost-effectively processed and classified to visualize objects of interest and identify physical changes with such software as ArcGIS Pro.

Small teams can quickly scan large areas and analyze the data to determine which assets need attention, focusing efforts on assets needing maintenance. This study validates deep-learning methods to classify power poles and towers from a LiDAR point cloud.

Due to time constraints and lack of training data, we opted for a model trained with other data. Though the study's results are suboptimal, they do show deep learning is viable. According to the Environmental Systems Research Institute's deep-learning documentation, a model developed with other data may underperform because the project data differs from the data used to train the model [19].

ArcGIS Pro comes equipped with tools to interactively classify points. These tools can be used to label points of interest in the training and validation data that accurately represent the project data. These steps should produce a model that can automatically classify other point clouds [20].

REFERENCES

[1] National Ocean Service, "What is LiDAR," US Department of Commerce. [Online]. Available from: https://oceanservice.noaa.gov/facts/lidar.html

[2] M. Van Leeuwen and M. Nieuwenhuis, "Retrieval of forest structural parameters using LiDAR remote sensing." Eur J For Res, vol. 129, pp. 749–770, 2010. [Online]. Available from: https://link.springer.com/article/10.1007/s10342-010-0381-4.

[3] D. V. Prokhorov, "Object recognition in 3D LiDAR data with recurrent neural network," IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, vol. 2009, pp. 9–15, 2009.

[4] E. Maggiori, Y. Tarabaka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," IEEE Xplore. [Online]. Available from: https://ieeexplore.ieee.org/document/7592858

[5] D. Kudinov, "PointCNN: Replacing 50,000 man hours with AI," Medium. [Online]. Available from: https://medium.com/geoai/pointcnn-replacing-50-000-man-hours-with-ai-d7397c1e7ffe

[6] Y.-C. Fan, C. M. Yelamandala, T.-W. Chen, and C.-J. Huang, "Real-time object detection for LiDAR based on LS-R-YOLOv4 neural network," J Sensor, vol. 2021, Article 5576262, 2021. [Online]. Available from: https://doi.org/10.1155/2021/5576262).

[7] A. Borcs, N. Balazs, and C. Benedek, "Instant object detection in LiDAR point clouds. Digital Object Identifier System," [Online]. Available from: https://doi.org/10.1109/LGRS.2017.2674799

[8] K. M. Brubaker, W. L. Myers, P. J. Drohan, D. A. Miller, and E. W. Boyer, "The use of LiDAR terrain data in characterizing surface roughness and microtopography," Appl Environ Soil Sci, vol. 2013, Article 891534, 2013. [Online]. Available from: https://doi.org/10.1155/2013/891534.

[9] F. Azevedo et al., "LiDAR-based real-time detection and modeling of power lines for unmanned aerial vehicles," MDPI. [Online]. Available from: https://www.mdpi.com/1424-8220/19/8/1812/htm

[10] F. H. Nahhas, H. Z. M. Shafri, M. I. Sameen, B. Pradhan, and S. Mansor, "Deep learning approach for building detection using LiDAR–orthophoto fusion," J Sensors, vol. 2018, Article 7212307, 2018. [Online]. Available from: https://www.hindawi.com/journals/js/2018/7212307.

[11] J. Antunes, "Should you choose LiDAR or photogrammetry for aerial drone surveys?" Commercial UAV News. [Online]. Available from: https://www.commercialuavnews.com/construction/choose-lidar-photogrammetry-aerial-drone-surveys

[12] K. K. Singh, G. Chen, J. B. McCarter, and R. K. Meentemeyer, "Effects of LiDAR point density and landscape context on estimates of urban forest biomass," ISPRS J Photogramm Remote Sens, vol. 101, pp. 310–322, 2015. [Online]. Available from: https://doi.org/10.1016/j.isprsjprs.2014.12.021.

[13] C. Van Tassel, "The true cost—Implementing LiDAR into your business," LiDAR News. [Online]. Available from: https://lidarnews.com/articles/the-true-cost-implementing-lidar-into-your-business

[14] T. E. Glavinich and A. L. Chichester, "Pricing service work for profit," Electrical Contractor Magazine. [Online]. Available from: https://www.ecmag.com/section/your-business/pricing-service-work-profit

[15] US Geological Survey, "3DEP LidarExplorer," [Online]. Available from: https://prd-tnm.s3.amazonaws.com/LidarExplorer/index.html

[16] Rapidlasso, "LAStools: Award winning software for rapid LiDAR processing," [Online]. Available from: http://lastools.org/download/laszip.exe

[17] GitHub, "Deep learning libraries installers for ArcGIS," [Online]. Available from: https://github.com/Esri/deep-learning-frameworks/blob/master/README.md?rmedium=links_esri_com_b_d&rsource=https%3A%2F%2Flinks.esri.com%2Fdeep-learning-framework-install

[18] Environmental Systems Research Institute, "Deep learning model to classify point clouds into distribution wires, poles, or background,"

[Online]. Available from: https://www.arcgis.com/home/item.html?id=6ce6dae2d62c4037afc3a3abd19afb11

[19] Environmental Systems Research Institute, “Classify a point cloud with deep learning,” [Online]. Available from: https://pro.arcgis.com/en/pro-app/latest/help/data/las-dataset/classify-a-point-clould-with-deep-learning.htm.

[20] Environmental Systems Research Institute, “Introduction to deep learning and point clouds,” [Online]. Available from: https://pro.arcgis.com/en/pro-app/latest/help/data/las-dataset/introduction-to-deep-learning-and-point-clouds.htm

 ISBN: 978-1-61208-967-6

Improved Data Preprocessing Approach to Short-Term Load Forecasting

Athanasios Ioannis Arvanitidis
Dept. of Electrical and Computer Engineering
University of Thessaly
Volos, Greece
atarvanitidis@uth.gr

Dimitrios Bargiotas
Dept. of Electrical and Computer Engineering
University of Thessaly
Volos, Greece
bargiotas@uth.gr

***Abstract*—One of the most critical aspects for the smooth operation of power systems is short-term load forecasting. Forecast accuracy has a significant impact on an electricity utility's economic viability and reliability. Thus, robust deep learning methods, such as artificial neural networks, are implemented in order to achieve higher accuracy load forecasting results. In this paper, a new preprocessing method of the input data of a neural network, which emphasizes on the importance of specific input data, that show a higher Pearson's correlation coefficient with the output result, is proposed. This work implements the proposed preprocessing technique and compares the results with those derived from the classical min-max scaling methods. Numerical results of next hour's load forecasting, based on a multi-layer perceptron with the implementation of the proposed data scaling approach, show higher precision than the typical scaling method, demonstrating the importance of our work.**

***Index Terms*—short-term load forecasting, data preprocessing, scaling techniques, multi-layer perceptron**

I. Introduction

One of the most critical parts of effective power system management is the ability to forecast electrical load consumption. The accuracy of predictions has a direct impact on the economic feasibility and dependability of electricity systems. Short-Term Load Forecasting (STLF) covers a time span of one hour to one week and it is utilized for day-to-day power system operations, such as economic dispatch, demand response, energy transaction scheduling, power flow analysis, and power system reliability and stability research [1]. Short-term load forecasting has traditionally been performed using approaches such as time series models, regression-based algorithms, and Kalman filtering [2]. Recently, methods based on artificial intelligence and deep learning algorithms have been widely employed for power system optimization, since they outperform conventional approaches in terms of generalization and prediction [3]. Their primary applications include optimum power system operation and management, load forecasting and energy price forecasting.

In recent years, approaches based on Artificial Neural Networks (ANNs), as well as other computational intelligence methodologies, have emerged as potentially robust methods for short-term load forecasting. The increased availability of data due mainly to the expanded installation of new power meters and the breakthrough in the computational capability of current computers have contributed significantly to the recent success of neural networks. STLF is mostly dependent on historical load data, such as load data from prior days, weeks or yeas, as well as temperature and humidity data. The availability of load data per minute utilized by different types of neural networks achieves impressive performance as it brings even greater accuracy in the forecast results [4]. However, the data entered into the neural networks are not used in raw format, but they undergo into various types of preprocessing, such as eliminating outliers, handling missing values and feature scaling, so that they can be used properly and increase the efficiency of the forecasting model. Min-Max, z-score, standard and max absolute normalization are the most reputable techniques for scaling input data. Despite their extensive use, these strategies have certain drawbacks [5], which provides an opportunity to develop novel scaling techniques that increase the predictive abilities of ANNs.

This paper presents a unique data preprocessing technique that differs from earlier work in that it highlights the significance of specific input data by using neural networks to forecast next hour's load. The proposed technique focuses on the importance of certain neural network's input variables in relation to output variables, resulting in improved prediction outputs than usual preprocessing methods. This approach is applied to data from the Greek Power System and is utilized by a Multi-Layer Perceptron (MLP) for short-term load forecasting.

Our paper is developed as follows. In Section 2, the prevalent and most widely used preprocessing techniques of neural network's input data are presented. Section 3 presents precisely the analysis of the enhanced scaling method we propose as well as the improvement in accuracy that results in the short-term load forecasting under consideration, while Section 4 concludes the paper.

II. Preprocessing Techniques of Neural Network's Input data

Data preprocessing aims at making the raw data at hand more amenable to neural networks. This includes vectorization, normalization, handling missing values, and feature extraction [6].

 ISBN: 978-1-61208-967-6

A. Feature Selection

Feature selection is one of the initial steps in studying and understanding the dataset in order to construct a robust prediction model. The selection of suitable variables as input data for the neural network, in order to boost the accuracy of the prediction outcomes, is referred to as feature selection. Pearson's r correlation coefficient between each pair of variables is used to select features as input data. The Pearson correlation coefficient is a metric for determining the strength of a linear relationship between two variables of the dataset taking numbers between -1 and 1. When it is near to one, it indicates a significant positive association. When the coefficient is near to -1, it indicates a significant negative association. Finally, coefficients close to 0 indicate that no linear association exists. The Pearson correlation coefficient ignores whether a variable is considered dependent or independent, evaluating all variables identically. Pearson's correlation coefficient is given by (1):

$$r_{\mathrm{X,Y}} = \frac{cov(X,Y)}{\sigma_X \cdot \sigma_Y} \tag{1}$$

where $cov(X,Y)$ is the covariance, σ_X is the standard deviation of X and σ_Y is the standard deviation of Y.

Therefore, feature extraction and identification are one of the most important steps in the field of energy forecasting, including short term load forecasting [7]. The variables in the dataset that exhibit the strongest linear correlation with the load should be utilized in order to achieve higher accuracy and reduce the complexity of load forecasting. It is necessary to identify which characteristics selected from dataset are containing the most relevant information helping to provide accurate predictions. This crucial step is also applicable in the field of Energy where artificial intelligence algorithms are widely used [8]. In our work, the features with the highest Pearson's correlation with the output variable of the proposed MLP are selected as neural network's inputs.

B. Data Scaling

Data scaling is one of the most critical operations that should performed on the input data. Machine Learning (ML) methods, with a few exceptions, do not perform well when the input numerical characteristics have extremely varied scales [9]. In general, neural networks do not accept relatively big values or input data that are heterogeneous, i.e., there are substantial differences in the order of magnitude. As a result, to boost the neural network's performance, input data should contain values inside a closed interval.

Differences in the magnitude of scaling across input variables may increase the difficulty of the problem being approached. A model with large scale values is frequently unstable, which means it may perform poorly during learning and be sensitive to input values, resulting in larger generalization error. A basic linear rescaling of the input variables is one of the most prevalent types of preprocessing. In [10], the authors highlight that input data normalization can enhance neural networks' overperformance by reducing effectively the estimation errors and the computational time needed.

In forecasting approaches based on time series data, the most common normalizing methods are min-max, decimal scaling, z-score, median, and sigmoid normalization. A comparative study of these standard normalization techniques on the time series forecasting is presented in [11]. The authors use deep recurrent neural networks to predict the Bombay and New York stock exchanges by normalizing the input data using the methods described above and analysing the outcomes to determine which methodology is preferred. Meanwhile, Ogasawara et al. [12] propose an adaptive normalization technique for normalizing non-stationary time series. This innovative approach is used along a feed-forward neural network in order to predict numerous economic factors, producing greater results than the traditional normalization methods. Furthermore, in [13], the authors study the effectiveness of batch normalization technique in different types of convolutional neural networks concluding that the implementation of a normalization approach to the input data is inevitable.

The issue of data scaling has also influenced researchers' efforts for STLF, as it applies directly to the various types of ANNs used in the literature. Specifically, Che et al. [14] examine various machine learning algorithms in the STLF issue. In their work, the authors propose a fusion load forecasting model based on Support Vector Machines (SVM), Random Forests (RF), Long Short-Term Memory (LSTM) neural networks along with the Ensemble Empirical Mode Decomposition algorithm for dealing with the abnormal data. Their approach was tested on 15-min interval data yielding Mean Absolute Percentage Error (MAPE) lower than 3%. Furthermore, Yi et al. [15] propose a Multi-Temporal-spatial-scale Convolutional Network (MTCN) in order to reduce the data noise error, improve the time series features and enhance the prediction accuracy. The input data used in this model have been normalized via the standard min-max normalization method. The model has been tested using load data from Chinese power system producing better results in compare to the traditional ANN models used in the STLF issue. In [16], Kwon et al. study the impact of minimum-maximum, z-score and decimal normalization approaches to the input data of a MLP for the prediction of the load for 24 hours on weekdays. Using load and temperature data of the past two days of the Korean power system, came to the conclusion that the conventional min-max scaling outperforms the other two methods as it produces MAPE of 1,97%.

Most papers in the existing literature suggest that datasets should be subjected to a global normalization technique. In [17], Passalis et al. present some global normalization methods for the STLF issue. In contrast with the existing literature, this paper proposes that only some of the input variables should be normalized based on their impact on the predicted results. The proposed data scaling is done by multiplying certain input data with importance coefficient in order to obtain an order of magnitude that appropriately determines their influence in the result of the forecast.

 ISBN: 978-1-61208-967-6

III. Analysis of the Proposed Scaling Method

Following a thorough review of the literature, an innovative data processing technique is suggested and applied to certain specific data depending on the Pearson's correlation coefficient. An MLP neural network, which is used to predict the value of the next hour's load using historical temperature and load data from previous days and the previous hour, is presented in detail in this section. The data used, containing hourly load values, derives from the Greek national power system for the years 2013-2017, from which 80% is chosen as training set, while the remaining 20% consists the test set. Our proposed MLP neural network consist of three layers; an input layer, a hidden layer, and an output layer, as depicted in Fig. 1. Historical load data, meteorological data such as temperature, wind speed and direction, and data relating to the seasonality of the load, such as hour, day, month, etc., are included in the dataset. In order to reduce the complexity of the suggested forecasting model, only data with a high Pearson correlation coefficient related to the load variable are chosen as input variables. The input variables used for next hour's load forecasting are the following:

- Hour: The time of day for which the load forecast is made.
- Week Day: A characteristic coding to denote the day of the week.
- Holiday: Binary values are used to indicate whether a day is a holiday, which includes Greek state holidays, religious holidays and the weekends, or a normal working day
- Temperature: The hourly value (in Celsius) of the temperature of the day for which the load is forecast.
- D-7 Load: The value of the load at the corresponding time on the same day of the previous week.
- D-1 Load: The load value of the day preceding the one for which prediction is made, at the corresponding time.
- H-1 Load: The value of the previous hour's load on which the forecast is based.

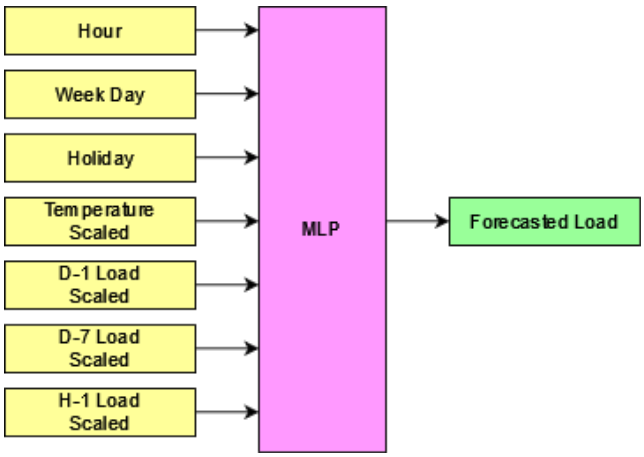


Fig. 1. Proposed MLP architecture for STLF.

Pearson's correlation coefficient of the input data is then calculated and compared to the neural network output, i.e., the load for the following hour. Data with a coefficient r near to +1 have a stronger impact on the outcome of the forecast and should thus be considered more important. The variables with higher r values compared to the output variable are $D\text{-}1Load$, $D\text{-}7Load$, $H\text{-}1Load$. As a result, in order to improve the predicting results, these variables with r approaching +1 in respect to the load variable are subjected to an improved scaling technique. The main benefit of this particular scaling for variables that have a strong correlation with the load variable is that they are given greater significance, allowing the neural network to use this knowledge and improve the forecasting accuracy. Fig. 2 summarizes the autocorrelation coefficient calculation results.

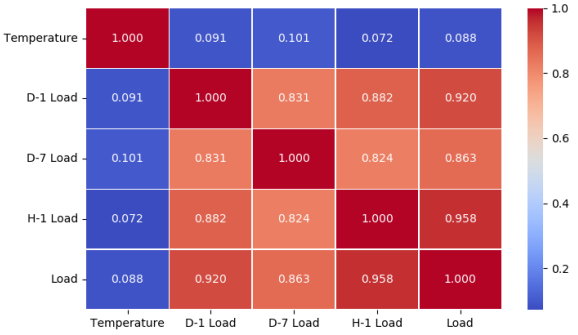


Fig. 2. Calculation of Pearson's correlation coefficient for input variables.

As previously stated, in order to produce more accurate prediction outcomes using neural networks, the input data should be scaled appropriately. Initially, the variables $Hour$, $WeekDay$ and $Holiday$ serve as labels for the day on which the prediction is created and are not susceptible to scaling. The $Temperature$ variable is subjected to the standard min-max scaling approach. Because of the large value of the coefficient r, the variables $D\text{-}1Load$, $D\text{-}7Load$ and $H\text{-}1Load$ are subject to both the standard min-max scaling approach and the modified min-max scaling method in (2). This paper proposes an enhanced min-max data preprocessing technique for STLF, that alters the order of magnitude of the variables $D\text{-}1Load$, $D\text{-}7Load$, $H\text{-}1Load$ giving them the appropriate weight, and compares the forecasing results with those obtained from the conventional implementation of the min-max method.

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \cdot ImpCoeff \tag{2}$$

where $ImpCoeff$ is an integer that appropriately identifies the significance of the data for the forecast result by allocating the input data within the closed interval [0, ImpCoeff].

A. Calculation of Importance Coefficient for the Enhanced Min-Max Scaling Method

The $ImpCoeff$ coefficient, which correctly attributes the significance of these variables, must be determined before

applying the suggested scaling method to the input data. $ImpCoeff$ is defined by the accuracy of the neural network prediction by calculating the resultant MAPE value through a trial-and-error procedure. At first, the coefficient accepts integer values in the range [1,100]. It is underlined that when the coefficient equals 1, the suggested method is associated with the traditional min-max scaling methodology. Fig. 3 depicts the MAPE values obtained by implementing the suggested MLP for the STLF at various $ImpCoeff$ values.

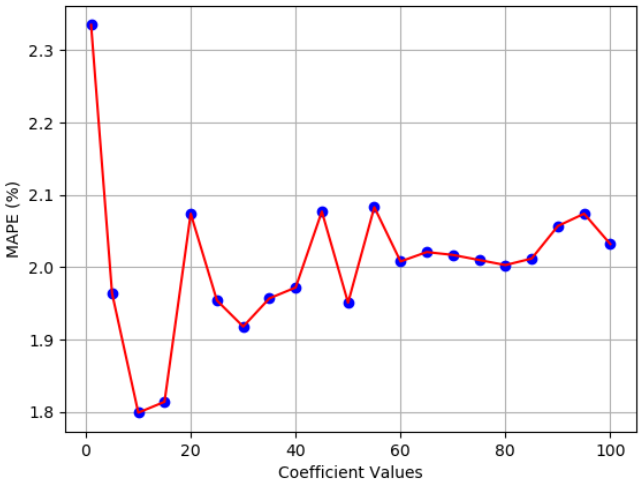


Fig. 3. MAPE calculation for the various ImpCoeff values.

Then it is discovered that for coefficient values in the interval [7,12], MAPE obtains the lowest values. Fig. 4 depicts the thorough computation of $ImpCoeff$ in this interval, stressing that when $ImpCoeff$ equals 10, MAPE yields the smallest feasible value. As a result, with $ImpCoeff$ equal to 10, our suggested scaling approach is obtained from (2).

B. Numerical Results

Before entering the proposed MLP neural network to estimate the following day's load, input data with the highest coefficient r value is exposed to both scaling strategies. Table I summarizes and compares the outcomes of both procedures. As predicted, the technique with the lowest MAPE is deemed to be more efficient.

TABLE I
MAPE CALCULATION FOR THE TWO SCALING TECHNIQUES OF INPUT DATA

Scaling Method	MAPE
Classic Min-Max Scaling	2.34%
Enhanced Min-Max Scaling	1.80%

It turns out that our enhanced Min-Max Scaling technique yields a lower MAPE value in the forecast. Despite its simplicity, this technique appropriately emphasizes the weight and importance of the input variables D-1$Load$, D-7$Load$ and

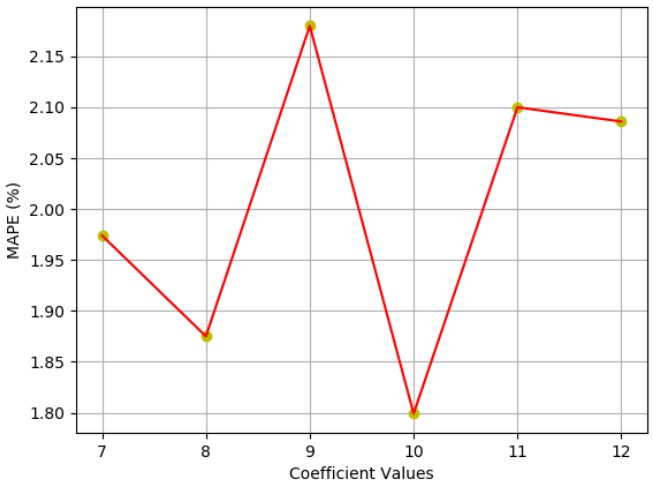


Fig. 4. Computation of optimum value for $ImpCoeff$.

H-1$Load$ in terms of MLP performance. When the proposed preprocessing technique is applied to the input data of a MLP neural network, the resulting value of MAPE decreases below to 2%, resulting in one of the lowest prediction value in the literature, based on data of the Greek interconnected power system. Fig. 5 and Fig. 6 provide a graphical comparison of prediction outcomes in 2017 using the proposed MLP for estimating next hour's load. In comparison to the usual scaling strategy, it is clear that the suggested method's outputs closely match the real load curve.

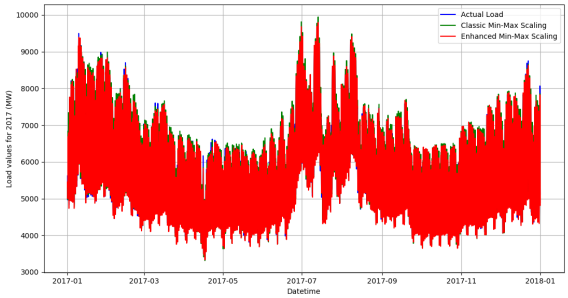


Fig. 5. Load curves for the year 2017.

IV. Conclusion

The increased use of neural networks in short-term load forecasting necessitates the development of novel data preparation approaches to increase the forecasting model's accuracy. In this paper, an enhanced preprocessing technique is presented that is applied to the input data of an MLP neural network to predict the value of the load in the following hour. This approach is based on the precise determination of a coefficient that assigns the proper importance to particular input data that demonstrate a high degree of correlation with

 ISBN: 978-1-61208-967-6

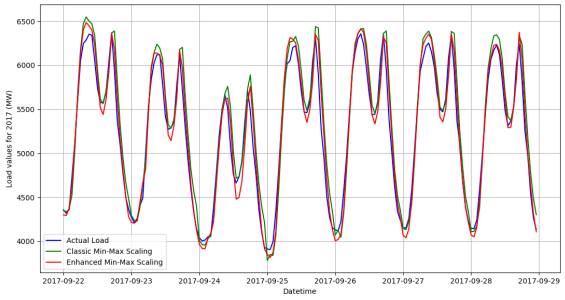


Fig. 6. Comparison of load prediction.

the proposed MLP's forecast output. Despite its simplicity, the findings of short-term load forecasting are more accurate when compared to other results in the literature that use data from the Greek interconnected system, as indicated by the low MAPE value, which is around 1.80%.

REFERENCES

[1] E. Kyriakides and M. Polycarpou, *Short Term Electric Load Forecasting: A Tutorial*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 391–418.

[2] M. Alamaniotis, D. Bargiotas, and L. Tsoukalas, "Towards smart energy systems: Application of kernel machine regression for medium term electricity load forecasting," *SpringerPlus*, vol. 5, 12 2016.

[3] A. I. Arvanitidis, D. Bargiotas, A. Daskalopulu, V. M. Laitsos, and L. H. Tsoukalas, "Enhanced short-term load forecasting using artificial neural networks," *Energies*, vol. 14, no. 22, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/22/7788

[4] D. Kontogiannis, D. Bargiotas, and A. Daskalopulu, "Minutely active power forecasting models using neural networks," *Sustainability*, vol. 12, no. 8, 2020. [Online]. Available: https://www.mdpi.com/2071-1050/12/8/3177

[5] P. Koprinkova and M. Petrova, "Data-scaling problems in neural-network training," *Engineering Applications of Artificial Intelligence*, vol. 12, no. 3, pp. 281–296, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197699000081

[6] D. Kontogiannis, D. Bargiotas, A. Daskalopulu, and L. H. Tsoukalas, "A meta-modeling power consumption forecasting approach combining client similarity and causality," *Energies*, vol. 14, no. 19, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/19/6088

[7] I. Jebli, F.-Z. Belouadha, M. I. Kabbaj, and A. Tilioua, "Prediction of solar energy guided by pearson correlation using machine learning," *Energy*, vol. 224, p. 120109, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544221003583

[8] D. Rochman, E. Baugé, A. L. Vasiliev, H. Ferroukhi, S. Pelloni, A. J. Koning, and J.-C. Sublet, "Monte carlo nuclear data adjustment via integral information," *The European Physical Journal Plus*, vol. 133, pp. 1–23, 2018.

[9] A. Gron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 1st ed. O'Reilly Media, Inc., 2017.

[10] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *IEEE Transactions on Nuclear Science*, vol. 44, no. 3, pp. 1464–1468, 1997.

[11] S. Bhanja and A. Das, "Impact of data normalization on deep neural network for time series forecasting," *ArXiv*, vol. abs/1812.05519, 2018.

[12] E. Ogasawara, L. C. Martinez, D. de Oliveira, G. Zimbrão, G. L. Pappa, and M. Mattoso, "Adaptive normalization: A novel data normalization approach for non-stationary time series," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1–8.

[13] V. Thakkar, S. Tewary, and C. Chakraborty, "Batch normalization in convolutional neural networks — a comparative study with cifar-10 data," in *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, 2018, pp. 1–5.

[14] W. Guo, L. Che, M. Shahidehpour, and X. Wan, "Machine-learning based methods in short-term load forecasting," *The Electricity Journal*, vol. 34, p. 106884, 01 2021.

[15] L. Yin and J. Xie, "Multi-temporal-spatial-scale temporal convolution network for short-term load forecasting of power systems," *Applied Energy*, vol. 283, p. 116328, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261920317128

[16] B.-S. Kwon, R.-J. Park, S.-W. Jo, and K.-B. Song, "Analysis of short-term load forecasting using artificial neural network algorithm according to normalization and selection of input data on weekdays," in *2018 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, 2018, pp. 280–283.

[17] N. Passalis and A. Tefas, "Global adaptive input normalization for short-term electric load forecasting," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2020, pp. 1–8.

 ISBN: 978-1-61208-967-6

Attention-guided Temporal Convolutional Network for Non-intrusive Load Monitoring

1st Huamin Ren
School of Economics, Innovation and Technology
Kristiania University College
Oslo, Norway
huamin.ren@kristiania.no

2nd Xiaomeng Su
Dept. of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
xiaomeng.su@ntnu.no

3rd Robert Jenssen
Dept. of Physics and Technology
UiT The Arctic University of Norway
Tromsø, Norway
robert.jenssen@uit.no

4th Jingyue Li
Dept. of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
jingyue.li@ntnu.no

5th Stian Normann Anfinsen
NORCE Norwegian Research Centre
UiT The Arctic University of Norway
Tromsø, Norway
stian.normann.anfinsen@uit.no

***Abstract*—With the prevalence of smart meter infrastructure, data analysis on consumer side becomes more and more important in smart grid systems. One of the fundamental tasks is to disaggregate users' total consumption into appliance-wise values. It has been well noted that encoding of temporal dependency is a key issue for successful modelling of the relations between the total consumption and its decomposed consumption on an appliance historically, and therefore has been implemented in many state-of-the-art models. However, how to encode the varied long-term and short-term dependency coming from different appliances is yet an open and under-addressed question. In this paper, we propose an Attention-guided Temporal Convolutional Network (ATCN), which generates different temporal residual blocks and provides an attention mechanism to indicate the importance of those blocks with respect to the appliance. Ultimately, we aim to address these two questions: i) How to employ both long-term and short-term temporal dependency to better disaggregate future loads while maintaining an affordable memory cost? ii) How to employ attention during the training of an appliance to obtain a better representation of the consumption pattern? We have demonstrated the effectiveness of our approach through comprehensive experiments and show that our proposed ATCN model achieves state-of-the-art performance, particularly on multi-status appliances that are normally hard to cope with regarding disaggregation accuracy and generalization capability.**

***Index Terms*—energy disaggregation, non-intrusive load monitoring, deep learning, temporal convolutional network, attention model**

I. Introduction

Non-Intrusive Load Monitoring (NILM), also referred to as energy disaggregation, aims to disaggregate the power consumption of a customer as a whole into detailed appliance-level consumption [1]. It has become one of the key tools to make effective use of the emerging smart meter infrastructure for the benefit of energy customers and producers, with great potential in applications such as energy awareness, energy conservation, and identification of controllable loads [2].

NILM has been framed historically both as classification and regression problems. Our paper is treating NILM as a regression problem, i.e, to estimate the consumption of individual appliances from the mains signal. In order to capture all distinct consumption patterns from all types of appliances, NILM algorithms tend to adopt a training dataset with a long time span (as long as memory permits) and attempt to learn temporal dependencies for each appliance. The trend is that recent work tends to utilize a range of deep neural network architectures, such as encoder-decoder networks, long short-term memory (LSTM) networks, bi-directional, sequence-to-sequence, and sequence-to-point [3] [4] [5] based prediction algorithms and their variants, including the very recent Bitcn-NILM algorithm [6], which combines sequence-to-point with bidirectional dilated convolution network. The key challenges of the prediction strategy are these: if the time window is too small, essential dependencies cannot be learned, e.g. if an appliance has a cyclic consumption pattern and the time window does not cover a full period. However, if it is too large, the efficiency of the scheme can significantly degrade, since loading long historical data burdens the memory requirement. Additionally, it also requires a much longer prediction time, and therefore cannot meet the needs of real applications.

Remark that different appliances exhibit vastly different temporal dependencies. As an example, we present in Figure 1 the daily power consumption of a household from the REDD dataset (shown in black color) along with its corresponding appliance-level consumption. Two important findings can be seen: 1) both local neighbors and far-away neighbors in disjoint time windows can together help with the disaggregation of appliance-level consumption. 2) for some appliances, local neighbors are most important for prediction of future consumption (e.g., fridge in Fig.1); for others, far-away points play a more important role (e.g., microwave and washer dryer in Fig.1). In other words, the relevant dependency ranges are specific to each appliance and should be adapted accordingly.

 ISBN: 978-1-61208-967-6

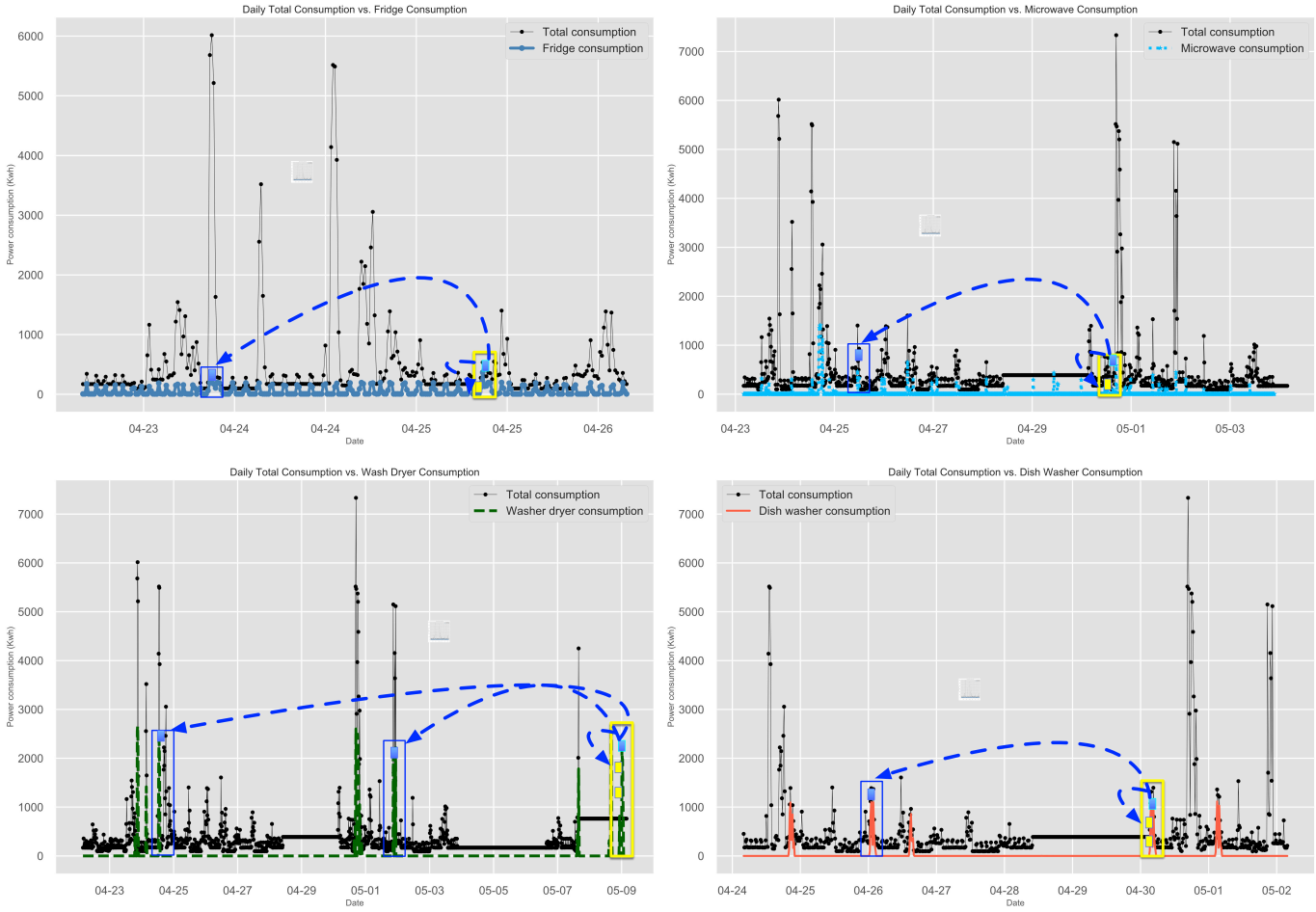


Fig. 1. Different appliances demonstrate various dependencies on short-term, mid-term and long-term neighbors. Appliance-wise consumption patterns in REDD dataset [7] showing temporal dependencies over mixed time scales: The power consumption marked with cyan indicates that it depends on its historical neighbors in a local time window (the yellow rectangle); A potential strong connection with far-apart neighbors (marked in blue points within a blue rectangle window) is also indicated. Depending upon the appliance, such a dependency can range one day ahead (fridge), or even six days ahead (microwave).

Therefore, the ability to use both long-term and short-term dependencies, while varying the attention on them according to the appliance, is crucial in NILM methodologies. How to accomplish this, therefore, is the key question in our paper.

II. Proposed Algorithm and Results

To the best of our knowledge, the adaptive attention is still missing in the current literature and the question of how to put varied attention on different appliances, i.e., short-term or long-term dependency, has not been fully addressed. We therefore propose an Attention-guided Temporal Convolutional Network (ATCN) to encode such dependencies.

Our algorithm sequentially acquires the input and actively attends relevant pieces of temporal information to refine the target consumption estimate at each time step. The key components are the *casual dilation* nature of the model and the *attention mechanisms*, both of which we empirically show the contribution to the appliance-wise consumption prediction. The overall architecture is show in Fig 3.

The comparative results with state-of-the-art algorithms on RMSE metric are shown in Table. I. Our ATCN algorithm has a competitive performance on multi-stage appliances, such as microwave and dish washer (on dish washer, our algorithm can achieve the second best place, just following after Seq2Point algorithm); it particularly achieves the best performance on the most difficult multi-stage appliance: washer dryer, owning to the combination of short-term and long-term dependency and attention mechanism.

In addition to RMSE metric evaluation, we also have provided the comparative results on EA metric, where estimated accuracy (EA) has also been employed as in [8] and [9]. Our approach has proven competitive performance on multi-stage appliances, including wash dryer and dish washer. Be noted that different evaluation metrics does not always have consensus result on different algorithms, which have been observed through the experiments. Moreover, failure to penalize the false detection during the evaluation procedure also has been noticed, and therefore how to propose a reliable evaluation criteria for NILM algorithms would be one of our

 ISBN: 978-1-61208-967-6

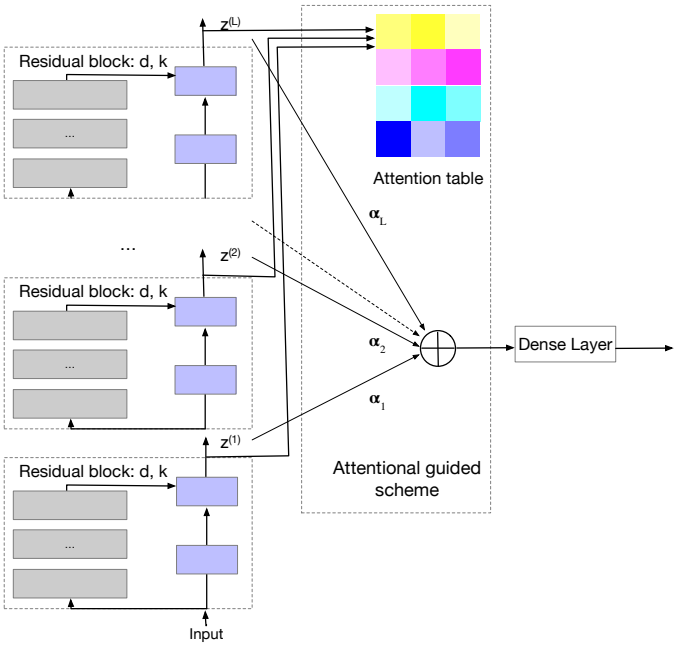


Fig. 2. Attention guided temporal convolutional networks.

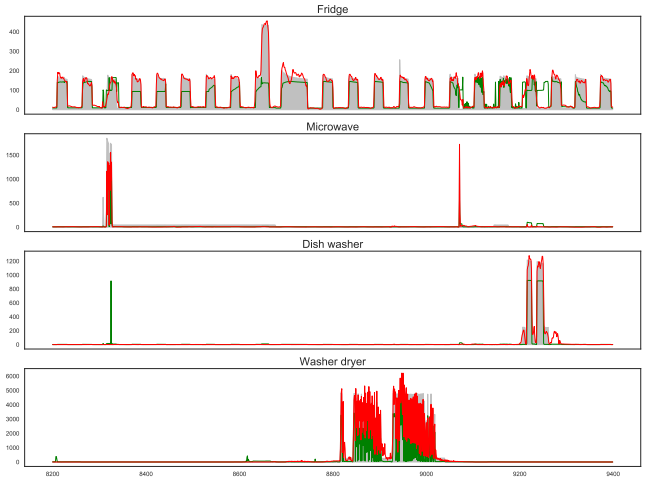


Fig. 3. Comparison of ATCN (red) with TCN (green) algorithm per each appliance on the REDD dataset (replace the last appliance.

research work in future.

III. Conclusion

As an important problem in smart home management, NILM still remains a challenge with great potential for further exploration and improvement. We propose a residual block concatenation strategy and apply an attention mechanism based on such residuals instead of dilated layers to improve NILM performance. The essential dilation and temporal convolution structure helps capture the long-term as well as short-term dependencies in the consumption signatures, while attention residuals ensure that the model's emphasis on relevant time scales is adapted to the appliance. Our proposed

TABLE I
Comparison of RMSE for various models.

Models	RMSE			
	Fridge	Microwave	Dish Washer	Wash Dryer
RNN	16.18	30.58	9.01	316.85
DAE	66.77	52.86	14.83	293.20
Seq2Seq	78.72	33.91	10.65	267.87
Seq2Point	72.52	28.56	4.70	196.10
TCN_NILM	34.71	37.75	40.16	221.82
ATCN_NILM	73.24	44.00	6.59	182.30

TABLE II
Comparison of EA for various models.

Models	EA			
	Fridge	Microwave	Dish Washer	Wash Dryer
RNN	0.97	0.73	0.86	0.50
DAE	0.78	0.39	0.77	0.55
Seq2Seq	0.74	0.69	0.86	0.83
Seq2Point	0.76	0.74	0.94	0.82
TCN_NILM	0.91	0.59	0.29	0.80
ATCN_NILM	0.75	0.55	0.90	0.85

ATCN algorithm outperforms state-of-the-art methodologies on multi-status appliances, especially those with short usage time, and has demonstrated excellent generalization capability.

Acknowledgment

This project was partly supported by Equinor under the Academia Agreement with UiT The Arctic University of Norway, and partly by Kristiania University College under interdisciplinary research fund under the project number: 10096 ForskM 2022- MLG-DA-500T.

References

[1] J. Z. Kolter and T. Jaakkola. "Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation," Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, La Palma, Canary Islands, pp. 1472–1482, 2012.

[2] W. Kong et al. "A Hierarchical Hidden Markov Model Framework for Home Appliance Modeling," in IEEE Transactions on Smart Grid, vol. 9, no. 4, pp. 3079-3090, July 2018.

[3] J. Kelly, and W. Knottenbelt. "Neural NILM: Deep Neural Networks Applied to Energy Disaggregation," In Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments (BuildSys '15). Association for Computing Machinery, New York, NY, USA, pp. 55–64, 2015.

[4] M. Kaselimi, N. Doulamis, A. Voulodimos, E. Protopapadakis, and A. Doulamis, "Context Aware Energy Disaggregation Using Adaptive Bidirectional LSTM Models," in IEEE Transactions on Smart Grid, vol. 11, no. 4, pp. 3054-3067, July 2020.

[5] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton. "Sequence-to-Point Learning With Neural Networks for Non-Intrusive Load Monitoring," AAAI Press, 2019.

[6] Z. Jia, L. Yang, Z. Zhang, H. Liu, and F. Kong, "Sequence to point learning based on bidirectional dilated residual network for non-intrusive load monitoring," International Journal of Electrical Power & Energy Systems, Volume 129, 2021.

[7] J. Z. Kolter, M. J. Johnson, "Redd: A public data set for energy disag- gregation research," in: IN SUSTKDD, 2011.

 ISBN: 978-1-61208-967-6

[8] H. Ren, F. Bianchi, J. Li, R. Olsen, R. Jense, S. Anfinsen, Towards applicability: A comparative study on non-intrusive load monitoring algorithms, in: ICCE, IEEE, United States, 2021.

[9] C. Klemenjak, S. Makonin, W. Elmenreich, Towards comparability in non-intrusive load monitoring: On data and performance evaluation, in: ISGT, USA, IEEE, 2020, pp. 1–5.

 ISBN: 978-1-61208-967-6

Constraint-based Modeling of Smart Grid Services in ICT-Reliant Power Systems

Frauke Oest
Department of Computer Science
University of Oldenburg, Germany
Email: frauke.oest@uni-oldenburg.de

Sebastian Lehnhoff
Department of Computing Science
University of Oldenburg, Germany
Email: sebastian.lehnhoff@uni-oldenburg.de

***Abstract*—The future Information and Communication Technology (ICT) infrastructure in distribution grids requires a significant network and computational resources for potentially running all so-called Smart Grid Services (SGS). The insufficient infrastructure may create network and computational congestions and resource shortages, which can lead to e.g., delayed critical messages in the power system and thus affect the power system stability. This paper presents a model for the configuration of SGSs with consideration of the underlying ICT infrastructure as a constraint satisfaction problem. This model is studied in a nominal and overvoltage scenario. The resulting over-constrained problem in the second scenario is relaxed by SGS data-rate reduction, SGS migration or SGS distribution. We show that an over-constrained problem can be relaxed with our proposed strategies.**

Keywords*—*smart grids*; *information and communication technology*; *quality of service*; *constraint satisfaction problem

I. Introduction

The operation of modern energy systems is based on a number of measurement, control and automation tasks for monitoring and operating extensively distributed resources, e.g., voltage stability monitoring or state estimation, for which different Quality of Service (QoS) requirements must be guaranteed. They serve as the basis for higher optimization functions that realize reliable, efficient and forward-looking overall system operation. In transmission systems, a dedicated and high-performance real-time communication infrastructure guarantees timing behavior and allows parallel execution of these communication-intensive functions and services with very heterogeneous and homogeneous latency requirements and communication demands. Such requirements range from, e.g., 10 ms up to several minutes maximum latency [1]. Due to the expansion of renewable energies at lower voltage level distribution networks and the shift of system responsibility to (operators of) these systems, similar functions and services must also be implemented in distribution systems – so-called *Smart Grid Services* (SGSs) with similar QoS requirements and guarantees. In addition, the future smart distribution system will be supplemented by additional SGSs comprising market and user-based applications, which may use the same communication infrastructure to enable a synergetic use [2]. This would require, similarly to transmission systems, a dedicated, over-provisioned communication infrastructure to run planned and future SGSs. Due to economical reasons, such infrastructure is not likely to be available soon (if ever) for smart distribution grids. Other solutions like wireless-based dedicated infrastructures or the usage of public networks do not provide sufficient and reliable resources to ensure QoS requirements of all potentially running SGSs. Additionally, the behavior and criticality of SGSs may change in different states of the power grid and thus QoS requirements may vary. For instance, in case of sudden changes of fluctuating renewable feed-in, the power grid may enter a critical state in which SGSs (e.g., feed-in management, voltage control, or congestion control) need to stabilize the system. In such critical situations, a reconfiguration might be necessary to meet new QoS requirements, e.g., due to higher sampling rates for increased measurement precision [3]. A solution to this problem could be the reconfiguration of SGS in the ICT-system with regards to the power system. Such reconfiguration can include a controlled reduction of data rates, migration of SGS to another server or a change in the overlay topology. This implies, that besides network QoS, also computational demands of SGSs must be considered to avoid congestion on servers. These SGS requirements and the limitations of the physical ICT infrastructure (i.e., computational resources on server, maximum line bitrates in the communication infrastructure) can be denoted as constraints and hence, this may be modeled as constraint satisfaction problem.

A. Related Work

Several aspects of flexibilization of communication for smart grids can be found in the literature. Virtualization concepts like Software-Defined Networking (SDN) may build the foundation to enable such flexibilization. In SDN, the network control (running on the SDN controller) is separated from the data flow (running on SDN devices). The infrastructure components (SDN devices) will be more quickly and adaptively configurable through abstract SDN applications in the SDN controller. One of the central features of SDN is the rapid adaptability of flows and packet routes and the implementation of QoS mechanisms [4]. In smart grids, the use of SDN has enhanced the reliability of field device communication through fast migration of functionality from a failed device towards a redundant device [5] and enabled a power system-dependent prioritization of SGS communication triggered by one SGS [6]. Another approach to virtualization in the communication system is called Network Function Virtualization (NFV), in which the network functions (e.g., routing, firewall, load balancer) are virtualized as an entire function and can thus be flexibly moved and multiplied [7]. This has been applied in smart grid communication where it demonstrates how this flexibility in communication may increase the dependability of

 ISBN: 978-1-61208-967-6

metering communication [8] or functionality may be moved towards the edge of the network near field devices in order to reduce traffic and communication delays [9]. Similar to NFV, the Grid Function Virtualization (GFV) concept enables the migration of SGSs. This concept has been studied in a simulation in case of server failures with a running SGS for voltage control. The affected SGS could be migrated after the failure, improving the voltage quality much faster compared to non-GFV simulations [10]. In another study, GFV was used to decrease the data rates of a non-prioritized SGS to favor QoS requirements of the prioritized SGS [11].

B. Contribution

This paper presents a constrained-based model for SGS configurations considering QoS and computational requirements for the underlying ICT infrastructure, which can be solved by using constraint satisfaction programming. This model is studied in a use case with two power system-based scenarios in which one scenario leads to non-satisfiability of constraints. With this, the usage of SGS flexibilities, such as data-rate reduction, migration and distribution, is motivated as a better controllable form of problem relaxation.

The remainder of the paper is structured as follows: The constraint-based model is defined in Section II. In Section III this model is integrated into an experimentation environment and scenarios with exemplary SGSs are defined. Furthermore, flexibilities of SGSs are introduced to resolve the constraint satisfaction problem if the scenario is over-constrained (e.g., due to changes in QoS of one SGS). Section IV presents the results for the nominal scenario and different over-constrained scenarios with and without using SGSs flexibilities showing the selected solutions and the computational performance of the process. Finally, Section V summarizes the paper, draws a conclusion, and presents ideas for future research.

II. Constraint-based Model

A Smart Grid Service (SGS) is an application serving the operation of the power grid. It contains field devices, such as sensors and actors in the power grid, and may contain a central server for processing field device data. SGSs impose requirements on the ICT infrastructure, such as maximum latency or minimum bitrate requirement, and minimum computational resources. The allocation of SGSs require a feasible path configuration of the single SGS in a given communication infrastructure and with regards to QoS requirements of SGSs. Therefore, this can be categorized as a combinatorial problem (such as other resource allocation problems) and may be formalized as a Constraint Satisfaction Problem (CSP). Our problem formalization is based on the definition of CSPs as described in Bartak et al. [12]. For this, the i-th SGS from a set of n SGSs will be associated with a decision variable x_i. In the following description, we continue to denote i for the SGS i specifics. The domain of each decision variable comprises a set of tuples $d_{i,j} \in D_i$ and represents a set of feasible solution candidates for SGS i, in which a single solution candidates $d_{i,j}$ can be selected by x_i. The solution candidate $d_{i,j}$ can be denoted as follows:

$$d_{i,j} = (p_{h_s}, m_{h_s}, s_{h_s}, a_{\max}, B) \text{ with } d_{i,j} \in D_i. \quad (1)$$

The constraints $C = \{c_1, c_2, \ldots, c_p\}$ may be defined by n-ary functions, which limit the values from the domain that can be assigned to the decision variable. The domain is determined by the parameterization of the SGS i and therefore, regards the field devices H_{f_i} and server h_{s_i} belonging to this SGS, as well as network (latency α_i, bitrate β_i) and computational QoS requirements (CPU p_i, memory m_i, storage s_i) and the configured overlay topology (centralized or distributed). In a centralized topology, field devices are connected to the central server in a star topology, whereas in the distributed topology, field devices are directly connected, creating a fully-meshed topology. The computational properties are defined as CPU demand p_{h_s}, memory demand m_{h_s} and storage demand s_{h_s}, which are applied to the server node $h_s \in H_{s_i}$. The network properties are determined by maximum path latency $a_{\max}$ and by the bitrate demands for all edges B which are determined by the paths representing the routing or data flow in the underlying ICT infrastructure. The ICT infrastructure is represented by the physical graph $G(V, E)$ defined by vertices V and edges E. A weighted edge is defined as $e = (v', v'', a, b)$, with v' as a source node connected to v'' with the edge weight properties a as line propagation latency and b as maximum line bitrates. The set of vertices V includes infrastructure nodes (e.g., router) R and hosts, such as server nodes H_s and field nodes H_f. To construct the network properties of a solution candidate, subgraphs of the physical graphs are constructed, which contain one feasible simple path for each end-to-end connection. This end-to-end connection is defined by the overlay topology and the host nodes of the SGS i. To create such subgraphs, the paths have to be determined first (e.g., by depth-first search). A path from the source h_1 to the target h_2 is determined by

$$p(h_1, h_2) = (h_1, r_1, \ldots, r_q, h_2) \quad (2)$$

where $r \in R$ and $h_1, h_2 \in H_s \cup H_f$. The path $p(h_1, h_2)$ is part of the set $P(h_1, h_2)$ comprising all potential paths from h_1 to h_2, which occur if there are cycles in the physical graph. In a preliminary step, a set of path combinations Z_i is constructed comprising one feasible path per end-to-end connection, i.e., one path combination z_i contains the topological information to construct subgraphs connecting all host nodes $h \in H_i$ of SGS i. For this, all end-to-end connections per SGS i are defined by the set of tuples $Q_i = \{(h_1, h'_1), \ldots, (h_n, h'_n)\}$ with $h, h' \in H_i$. The set of path combinations can then be constructed as a cartesian product by

$$Z_i = \prod_{q \in Q} P(q). \quad (3)$$

Based on this, the subgraph $f_{i,j}$ can be built by adopting the edge latency of the physical graph $G(V, E)$ if the vertices $(v', v'') \in z_{i,j} \in Z_i$ are also $v', v'' \in V(G)$. The edge bitrates $e(b)$ are determined for the edge $(v', v'') \in z_{i,j}$ by the bitrate

 ISBN: 978-1-61208-967-6

QoS requirement β_i of the SGS i for the solution candidate j, such that

$$f_{i,j}(e(b)) = e(b) + \beta_i. \tag{4}$$

This implies, that an edge in this subgraph is used by multiple simple paths, the bitrate weight is adjusted by summing the bitrate weight with the number of paths containing this edge. The bitrate resource B of the solution candidate $d_{i,j}$ can be derived from the subgraph f_i as an adjacency matrix B_{adj}. The end-to-end latency a_{max} of the solution candidates $d_{i,j}$ is determined by

$$a_{\text{max}} = \max_{h_1,h_2 \in H_i} \left(\sum_{e \in E(P_{h_1,h_2})} e(a) \right). \tag{5}$$

The amount of SGSs running on a server $h \in H_s$ is limited by its server resources for cpu h_p, memory h_m and storage h_s. As x_i chooses a solution candidate $d_{i,j}$, the information can be accessed by the function $p_{h_s}(x_i)$ for using the CPU usage on server h_s of this solution candidate. This also applies to $m_{h_s}(x_i)$ and $s_{h_s}(x_i)$. Therefore, the server resource constraints are defined by the constraints c_p, c_m, and c_s

$$c_p := \sum_{x_i \in X} p_{h_s}(x_i) \leq h_p \tag{6}$$

$$c_m := \sum_{x_i \in X} m_{h_s}(x_i) \leq h_m \tag{7}$$

$$c_s := \sum_{x_i \in X} s_{h_s}(x_i) \leq h_s \tag{8}$$

The network constraints are defined by the latency constraint c_a and the bitrate constraint c_b, which should not exceed the maximum physical bitrates of the weighted adjacency matrics E_{adj} of G with bitrates as weights.

$$c_a := a_{\text{max}}(x_i) \leq \alpha_i \tag{9}$$

$$c_b := \sum_{x_i \in X} B_{\text{adj}}(x_i) \leq E_{\text{adj}} \tag{10}$$

In this constraint model, constraints can be relaxed by manually adjusting the SGS parameterization, such as the overlay topology, network or computational QoS requirements.

III. Methodology

The aforementioned constrained-based model is integrated into a constraint satisfaction program and then exemplary scenarios are defined based on a physical graph and five SGSs with network and computational requirements. These requirements considered in the scenarios depend on the state of the power system (e.g., nominal or overvoltage). The overvoltage scenario leads to a non-solvability of the problem. Therefore, relaxation strategies are presented at the end of this section.

A. Experimentation Setup

The constrained-based model is implemented using NetworkX [13] for solution candidate generation and Minizinc [14] for constraint satisfaction programming and solving. For this, the physical graph is modeled in NetworkX comprising all nodes and edges with latency and bitrate properties. SGSs can be defined by the QoS requirement parameters (latency, bitrate, CPU, memory, storage), the assigned field devices and server, and the overlay topology (centralized or distributed). NetworkX is used to determine the simple paths in order to create subgraphs containing all end-to-end connections to all host nodes (i.e., server and field devices) of an SGS. These subgraphs can be transformed to adjacency matrices for easier bitrate calculation in Minizinc. Thus, NetworkX is used to determine the adjacency matrix for bitrate calculations and the corresponding end-to-end path latency as part of the solution candidate. The SGSs and physical constraints are implemented as a Minizinc model, the solution candidates are integrated via a python interface to Minzinc as a model instance. Gecode [15] is used as a solver in Minizinc. The output of Minizinc consists of a feasible solution candidate for each SGS, which can be visualized in NetworkX.

The model of the physical communication network in Figure 1 is based on the modeling of field devices in the CigreMV power grid [11]. The inner communication infrastructure is adjusted with regards to bitrates and latency. Routers are connected in three subnetworks and one core network linking the subnetworks, which creates a hierarchical infrastructure [16] [17]. The core network is defined by the routers R10, R20 and R30; the subnetworks comprise the routers R11 - R12, R21 - R23 and R31 - 32. The edge of the network is modeled by the 17 field devices (F11 - F16, F21 - F26, F31 - F35) and the two servers S1 and S2. The modeled infrastructure is the most powerful in its core network, followed by the subnetworks and the edge network. Therefore, the core network is modeled with 200 kByte / s bitrate resource and a propagation delay of 20 ms on each edge. The subnetwork is defined with 100 kByte / s bitrate and a latency of 25 ms and the edge network is denoted with a bitrate and latency property of 25 kByte / s and 30 ms. The server S1 contains 6 CPU cores, 16 GB memory and 50 GB storage. The server S2 is modeled with 4 CPU cores, 8 GB memory and 20 GB storage capacity.

B. Scenarios

We test our model from Section II with two scenarios representing two states of the power system. In both scenarios, we consider the SGSs Adaptive Relaying (AR), Coordinated Voltage Control (CVC), Line Monitoring (LM), State Estimation (SE) and Virtual Power Plant management (VPP). The first scenario describes the nominal state of the power system, in which the power system properties (voltage, active and reactive power, temperature of the operating equipment) are within the ideal range, so that grid critical services only need the minimum computational and network resources. The bitrate and latency requirements for SGS AR and LM are based on [18] and the requirements for the SGSs CVC and

 ISBN: 978-1-61208-967-6

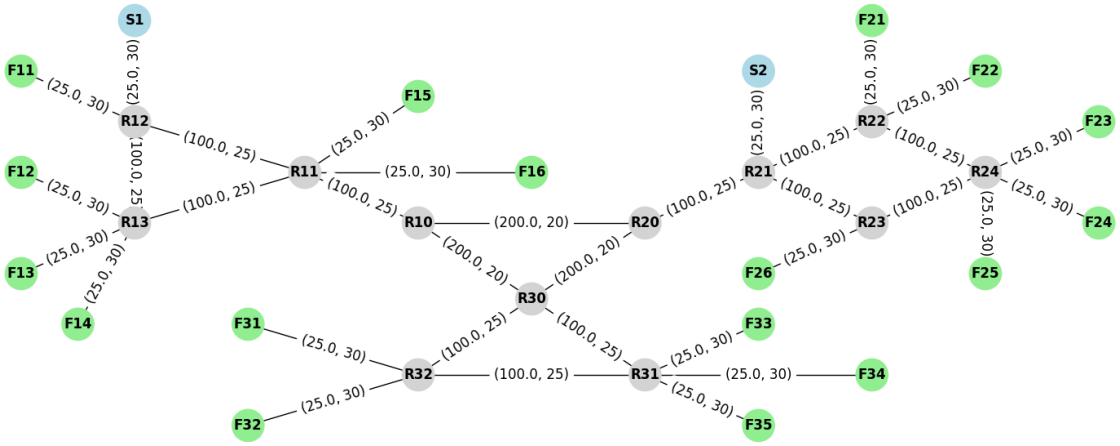


Figure 1. Physical graph with bitrates and latency on the edges.

SE are based on [19]. The bitrate requirements are derived from the sampling rates. For this, we estimate the size of a TCP / IP packet including payload to 100 Byte per sampling value. The QoS requirements for the SGS VPP are derived from the assumptions and results of [20].

The computational demands of SGSs are very dependent on the implementation of the service. Therefore, we try to categorize the demands based on a likely implementation. For this, we determine the computational demands into the categories low, medium and high for each computational demand characteristic. In Table I, these categories are translated into concrete requirements. The CPU resources are modeled as proportions (as millicpu) of one hyper thread on a bare-metal Intel processor. These computational demands are idealized abstractions but may work as a rough estimate.

TABLE I. COMPUTATIONAL RESOURCE DEMANDS FOR SGSS.

Resource demand	*low*	*medium*	*high*
CPU	500m	1.0	4.0
Memory	200MB	2GB	8GB
Storage	500MB	4 GB	16GB

The SGS demands for computational resources are estimated based on some implementation characteristics, e.g., whether historical data is needed, if the SGS can be parallelized (by, e.g., a multi-agent system or some machine learning methods) or the amount of measurement data. Therefore, SGS demands are determined based on the following assumptions: The purpose of the SGS LM is the evaluation of (a limited amount of) measurement data against a threshold [21]. The computational resource demands should be low in each category. The SGS SE for distribution grids is often based on a weighted least-squares approach and thus, needs pseudo-measurements to approximate missing measurements. Those can be achieved by a trained machine learning model [22]. Hence, we estimate the CPU demand as high, the memory demand as medium, and the storage demand as low. For the SGS AR, knowledge about topology, load flow, and characteristics of DGs are needed as an input for, e.g., a machine learning or linear programming-based approach [23]. Therefore, we assume high CPU demand, and medium memory and storage demands. The SGS CVC is threshold-based but needs short-term load and generation forecasts to avoid unnecessary control and switching signals, which may be caused by weather-dependent generators [24] [25]. For this, we assume that weather forecasts are included in the voltage calculation. Therefore, the resource demands are medium in each category. For the SGS VPP, we assume it to be implemented as a multi-agent system that can be run either centralized on a server or distributed by each agent representing a distributed generator. To find an optimal combined operational schedule, each agent may choose a schedule from a pre-defined set of feasible schedules for each generator participating in this VPP. [20]. The resources demands are assumed to be high for CPU, memory and storage. The resulting parameterization for SGSs can be taken from the following Table II.

TABLE II. PARAMETERIZATION OF SGSS FOR THE NOMINAL SCENARIO.

SGS	β *Byte / s*	α *ms*	*c cores*	*m GB*	*s GB*	*field devices*	*server*
AR	2.5	100	1	2	4	F36	S2
CVC	3	500	1	2	4	F16, F22, F35	S1
LM	1	1	0.5	0.2	0.5	F12, F24, F32	S2
SE	0.5	1000	1	2	0.5	F11, F15, F21, F23, F33, F34	S1
VPP	2	800	4	8	4	F13, F14, F25, F31	S1

The second scenario is describing an overvoltage situation in the power grid. Therefore, the QoS and computational requirements of CVC increase leading to a bitrate requirement of at least 6 kByte/s, a maximum end-to-end latency of 200 ms and a CPU demand of 4 cores. The CSP cannot be solved anymore, as this introduces a bitrate shortage from R12 to S1, all paths $P(F35, S1)$ exceed the latency requirement and server S1 cannot supply enough computational resources. We propose the following SGS flexibilities to relax this overconstrained problem:

 ISBN: 978-1-61208-967-6

Reduction: Reduction of QoS requirements of a non-prioritized SGS. In this scenario, the bitrate requirement of SE is reduced to β=1 kBit / s

Virtualization: Migration of an SGS to another server. In this scenario, the SGS CVC is migrated from S1 to S2.

Distribution: Change of overlay topology of an SGS. In this scenario, the SGS VPP switches from centralized to distributed operation, which does not need communication to any server.

We have conducted one experiment for the nominal scenario and six for the overvoltage scenario testing each constraint violation with and without relaxation.

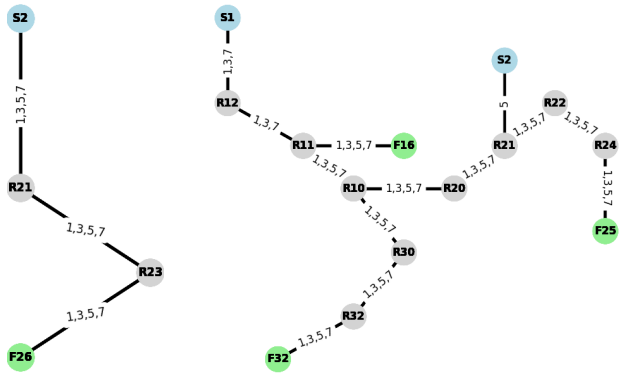
IV. Results

We have conducted seven experiments and present the resulting solutions (each as a possible SGS configuration) and the accompanied solution process data, such as the number of solution candidates per SGS and the calculation time of each step. The experiments are enumerated as follows:

1) Nominal case.
2) Overvoltage case, bitrate demand of CVC is increased, no problem relaxation.
3) Overvoltage case, bitrate demand of CVC is increased, SE communicates with reduced bitrate requirements.
4) Overvoltage case, latency demand of CVC is increased, no problem relaxation.
5) Overvoltage case, latency demand of CVC is increased, CVC is migrated to another server.
6) Overvoltage case, CPU demand of CVC is increased, no problem relaxation.
7) Overvoltage case, CPU demand of CVC is increased, VPP operates in the distributed mode.

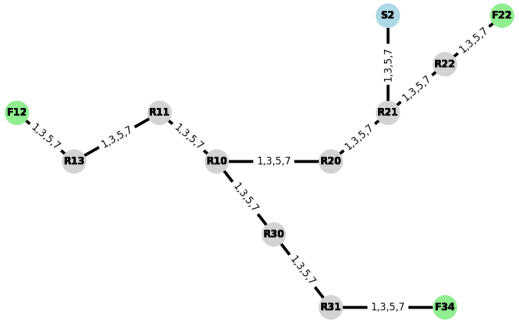
A resulting feasible configuration is displayed in Figure 2 for each SGS. The chosen topologies for the SGSs AR (2a, CVC (2b), LM (2c, SE (2d), and VPP(2e) are displayd by the experiment number on the edge of the graphs. Following the idea of a CSP, the first valid solution for the experiment is used. Therefore, often the same first valid solution candidate is chosen. Experiment 1 shows a feasible solution for the nominal scenario in which all constraints can be fulfilled and hence, no relaxation is needed. This experiment is used to illustrate topological changes in the following experiments. In case of overvoltage without a problem relaxation with SGS flexibilization, no solutions can be found. For this reason, experiments 2, 4, 6 cannot be visualized in Figure 2. The successful problem relaxations are visible in experiments 3, 5 and 7. The bitrate reduction of SGS SE conducted in experiment 3 lead to the same topology as in the nominal scenario. In experiment 5, SGS CVC shifted the end-to-end connectivity from the field devices from server S1 to server S2, visible by the edge from R21 to S2. In experiment 7, SGS VPP Management does not have a connection from R12 to S1.

Table III shows the number of generated solution candidates per SGS, the accompanied time to generate solution candidates

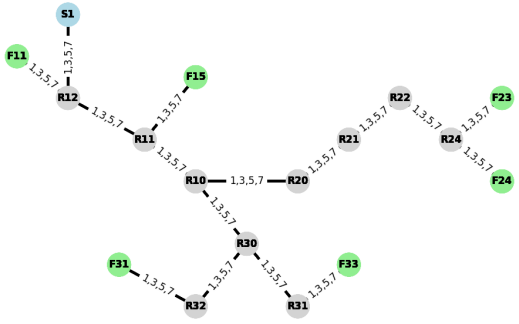


(a) AR

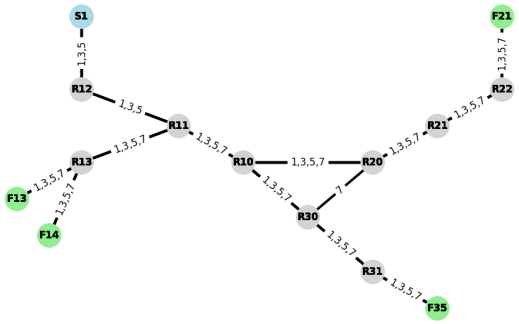
(b) CVC



(c) LM



(d) SE



(e) VPP

Figure 2. The resulting SGSs configurations as solutions of the CSP.

TABLE III. SIZE OF THE SET OF SOLUTION CANDIDATES AND TIME MEASURES OF THE SOLUTION FINDING PROCESS.

exp.	*number of solution candidates*					*generation time of solution candidates (s)*					*solving duration (s)*	
ID	*AR*	*CVC*	*LM*	*SE*	*VPP*	*AR*	*CVC*	*LM*	*SE*	*VPP*	*mean*	*std.*
1	2	128	32	8192	256	0.0011	0.0355	0.0101	3.3635	0.0857	18.2850	0.4118
2	2	128	32	8192	256	0.0011	0.0348	0.0100	3.5954	0.0884	18.1686	0.3248
3	2	128	32	8192	256	0.0011	0.0346	0.0099	3.5270	0.1037	18.0044	0.2305
4	2	128	32	8192	256	0.0011	0.0355	0.0101	3.5528	0.0775	10.4416	0.2333
5	2	16	32	8192	256	0.0011	0.0061	0.0223	3.5493	0.0821	18.0519	0.3075
6	2	128	32	8192	256	0.0011	0.0347	0.0101	3.5908	0.0740	16.7446	0.4263
7	2	128	32	8192	32768	0.0011	0.0355	0.0099	3.4880	14.5952	84.9177	1.0081

and the mean time and standard deviation to find a solution for the problem created in the experiment. It shows the change of generated solution candidates for SGS CVC and VPP in experiments 5 and 7, where a topology change was conducted by migration or the switch to a distributed topology. In experiment 5, only 16 CVC solution candidates were generated, whereas in experiment 7, 32768 VPP candidates were created, which is 128 times the size of the centralized VPP topology. These differences are caused by the placement of field devices and the server in this example physical graph. In experiment 5, the size decreases due to the decrease of hops per end-to-end connection. The fully-connected topology in experiment 7 leads to six end-to-end connections, which may include 3 cycles in the physical graph. As each traversed cycle creates 2 possible paths, this leads to 8 paths for 5 of 6 end-2-end connections (there is only one possible path for $P(F13, F14)$. This results in a total of 32768 solution candidates.

The time measurements for the generation of solution candidates and the solving are conducted on an Intel Xeon CPU E5-2680 with 4×2.40 GHz and 24 GB RAM. The solving process was performed 25 times and all time measures are in seconds. The time to generate solution candidates is similar in each experiment apart from experiment 5 and 7, where the duration for CVC decreases and the duration for VPP increases. Also, the time to solve the CSP is similar in experiments 1, 2, 3, 5 and 6. The decrease of solving time in experiment 4 is caused by the definition of the latency constraint, as this violation is easy to detect. The increase in solving time in experiment 7 is determined by the size of potential solution candidates.

V. CONCLUSION AND FUTURE WORK

In this paper, we have presented how to flexibilize SGSs and we have shown the potential of this approach by creating a sporadically occurring high ICT-demand situation on a small communication network with five SGSs. We have formalized ICT demands as a constraint satisfaction problem considering communication network requirements (latency and bitrates), and computational requirements (CPU, memory and storage usage). For this, we have studied 7 experiments based on two scenarios: nominal operation and overvoltage. The first scenario serves as a baseline experiment. The latter is defined by an increase of several requirements of the service CVC, leading to an over-constrained problem. We have conducted experiments with each individual adjusted requirement and a corresponding problem relaxation strategy using the flexibilities of SGSs, such as reduction of requirements, migration to other servers or switching to a distributed communication topology. By using these flexibilities, the new requirements of CVC could be fulfilled.

These findings show the potential of such SGS flexibilities to operate the power system resiliently under QoS considerations without the need for a strong over-provisioned dedicated communication infrastructure. SGS flexibilities may improve the maximization of the number of SGS requirements fulfilled by migration or distribution of some SGSs and a controlled degradation of requirements of low prioritized SGSs. The proposed approach needs further extensions in future work. So far, we only distinguish between the two categories prioritized and non-prioritized SGS. Furthermore, there is no optimized selection of the most suitable relaxation strategy. The approach can be enhanced by integrating a fine-grained order of SGSs based on pre-defined power system states and refining our SGS model by defining an order of SGS flexibilities with regard to performance degradations. For larger scale experiments based on larger physical networks and an increased amount of SGSs, we consider refining our constraint-based model and search space modeling (e.g., by pre-filtering the solution candidates) to improve the solving time. We will integrate those ideas to an adaptive process, which will be evaluated in a simulation.

ACKNOWLEDGMENT

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation).

REFERENCES

[1] D. Griffith, M. Souryal, and N. Golmie, "Wireless networks for smart grid applications," *Smart Grid Communications and Networking*, vol. 9781107014, pp. 234–262, 2010.

[2] B. Sörries, S. Lucidi, L. Nett, and M. Wissner, "Gutachten: Digitalisierung der Energiewende. Topthema 3: TK-Netzinfrastruktur und TK-Regulierung [Report: Digitization of the energy transition. Topic 3: Telecommunications network infrastructure and telecommunications regulation]," Wissenschaftliches Institut für Infrastruktur und Kommunikationsdienste, Tech. Rep., 2019.

[3] P. Das, A. Narayan, D. Babazadeh, P. Baboli, and S. Lehnhoff, "Real-time context-aware operation of digitalized power systems by reporting rate control of pmus," in *2021 IEEE Madrid PowerTech*, 2021, pp. 1–6.

[4] M. H. Rehmani, A. Davy, B. Jennings, and C. Assi, "Software defined networks-based smart grid communication: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2637–2670, 2019.

[5] F. von Tüllenburg, P. Dorfinger, A. Veichtlbauer, U. Pache, O. Langthaler, H. Kapoun, C. Bischof, and F. Kupzog, "Virtualising redundancy of power equipment controllers using software-defined networking," *Energy Informatics*, vol. 2, pp. 1–20, 2019.

 ISBN: 978-1-61208-967-6

[6] N. Dorsch, F. Kurtz, S. Dalhues, L. Robitzky, U. Häger, and C. Wietfeld, "Intertwined: Software-defined communication networks for multi-agent system-based smart grid control," in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2016, pp. 254–259.

[7] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys and Tutorials*, pp. 236–262, 2016.

[8] M. Niedermeier and H. de Meer, "Constructing Dependable Smart Grid Networks using Network Functions Virtualization," *Journal of Network and Systems Management*, p. 449–469, 2016.

[9] F. Kurtz, N. Dorsch, and C. Wietfeld, "Empirical comparison of virtualized and bare-metal switching for sdn-based 5g communication in critical infrastructures," *IEEE NETSOFT 2016 - 2016 IEEE NetSoft Conference and Workshops: Software-Defined Infrastructure for Networks, Clouds, IoT and Services*, pp. 453–458, 2016.

[10] C. Krüger, A. Narayan, F. Castro, B. Hage Hassan, S. Attarha, D. Babazadeh, and S. Lehnhoff, "Real-time test platform for enabling grid service virtualisation in cyber physical energy system," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2020, pp. 109–116.

[11] S. Attarha, A. Narayan, B. Hage Hassan, C. Krüger, F. Castro, D. Babazadeh, and S. Lehnhoff, "Virtualization management concept for flexible and fault-tolerant smart grid service provision," *Energies*, vol. 13, no. 9, 2020.

[12] R. Barták, M. A. Salido, and F. Rossi, "Constraint satisfaction techniques in planning and scheduling," *Journal of Intelligent Manufacturing*, p. 5–15, 2010.

[13] "NetworkX - network analysis in python," https://networkx.org/, accessed: 2022.04.06.

[14] "Minizinc," https://www.minizinc.org/, accessed: 2022.04.06.

[15] "Gecode," https://www.gecode.org, accessed: 2022.04.06.

[16] J. F. Kurose, *Computer networking : a top-down approach*. Addison-Wesley, 2017, ch. Network Layer, p. 335.

[17] G. Bedürftig, J. Kapanen, E. M. Metsälä, and J. T. Salmelin ha, *LTE backhaul: planning and optimization*. John Wiley and Sons, 2016, ch. LTE Backhaul, pp. 3–45.

[18] C. Samitier, *Field Device to Central Platform Applications*. Cham: Springer International Publishing, 2017, pp. 29–36.

[19] L. E. Miller, A. Silverstein, D. Anand, A. Goldstein, Y. Makarov, F. Tuffner, and K. Jones, "PMU Data Quality: A Framework for the Attributes of PMU Data Quality and a Methodology for Examining Data Quality Impacts to Synchrophasor Applications," U.S. Department of Energy, Tech. Rep. 3, 2017.

[20] F. Oest, M. Radtke, M. Blank-Babazadeh, S. Holly, and S. Lehnhoff, "Evaluation of communication infrastructures for distributed optimization of virtual power plant schedules," *Energies*, vol. 14, no. 5, 2021.

[21] D. Douglass and et al., "Ieee standard for calculating the current-temperature relationship of bare overhead conductors (ieee std 738™-2012)," IEEE Power and Energy Society, Tech. Rep., 2013.

[22] K. Dehghanpour, Z. Wang, J. Wang, Y. Yuan, and F. Bu, "A survey on state estimation techniques and challenges in smart distribution systems," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 2312–2322, 2019.

[23] C. Chandraratne, T. Logenthiran, R. T. Naayagi, and W. L. Woo, "Overview of adaptive protection system for modern power systems," in *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, 2018, pp. 1239–1244.

[24] K. E. Antoniadou-Plytaria, I. N. Kouveliotis-Lysikatos, P. S. Georgilakis, and N. D. Hatziargyriou, "Distributed and decentralized voltage control of smart distribution networks: Models, methods, and future research," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2999–3008, 2017.

[25] G. C. Kryonidis, E. O. Kontis, A. I. Chrysochos, C. S. Demoulias, and G. K. Papagiannis, "A coordinated droop control strategy for overvoltage mitigation in active distribution networks," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5260–5270, 2018.

 ISBN: 978-1-61208-967-6

Training an Energy Management Simulation with Multi-Agent Reinforcement Learning

Alexander Haemmerle *, Kapil Deshpande *, Philipp Moehl *, Georg Weichhart *
*Robotics and Automation Systems
PROFACTOR GmbH, Steyr, Austria
https://www.profactor.at/
emails: {alexander.haemmerle,kapil.deshpande,philipp.moehl,georg.weichhart}@profactor.at

***Abstract*—In this paper, we report on the application of multi-agent reinforcement learning to the development of a microgrid energy management simulation. The simulation is made up of energy producers and consumers as well as storage devices. We regard these components as agents that are trained in a shared environment with reinforcement learning. A significant share of energy production in the microgrid is provided by renewable energy sources with stochastic characteristics, e.g., photo-voltaic installations. The stochastic nature of such producers, as well as of consumers, is captured in energy consumption/production profiles that are used for training the respective agents. For our results, the agents have been trained with an actor-critic algorithm, using real-world energy profile data for photo-voltaic installations and industrial consumers in Austria. A centralised critic addresses the multi-agent nature of the energy management problem. Running what-if analyses is an application scenario for the trained simulation. In such analyses the effects of different microgrid configurations on energy management performance can be investigated. The presented work has been conducted in the context of the projects RESINET and Zer0p.**

***Index Terms*—Energy Management; Multi-Agent Reinforcement Learning; Photo-Voltaic; Battery Storage; Microgrid.**

I. Introduction

Green Industry for a sustainable and economically prosperous future is becoming a reality [1]. One of the pillars for Green Industry is the introduction of renewable energy sources into the energy supply for industrial companies. In many cases, the installment of a renewable energy source will be local, e.g., photo-voltaic (PV) panels on a plant roof. Because of the stochastic nature of prominent renewable energy sources like PV panels and wind turbines, such installments have to be accompanied with adequate storage systems to absorb excess renewable energy, and to provide energy in times of low energy production. However, companies that are willing to invest into local renewable energy sources are facing a decision problem, as they have to decide upon the right size of the installation, i.e., storage capacity and power output. The "right size" does not only depend on the company's energy consumption profiles, but also on local (meteorological) conditions that impact renewable energy production. An energy consumption/production profile is a time series of power values. In our work, we use real-world data from Austria, containing industrial consumption profiles as well as energy production profiles from PV installations.

The above decision problem motivated the following research question: Is it possible to develop an energy management simulation for a microgrid, leveraging historic data for renewable energy production profiles and consumption profiles? Such a simulation would then allow to run what-if analyses with different sizes of local renewable energy installations.

The research question contains two important aspects: leverage data, and energy management, which is a sequential decision-making problem in nature. These aspects motivated the usage of Deep Reinforcement Learning (DRL) to tackle the research question. Moreover, an energy management system contains multiple agents with potentially selfish agendas. However, in order to meet system-wide objectives (e.g., load balancing) the individual agents have to cooperate. To also meet the multi-agent character of energy management, we employ Multi-Agent Reinforcement Learning (MARL).

The remainder of the paper is structured as follows. In section II we present related work and argue our contribution, the proposed approach is described in section III, results are reported in section IV, and finally section V presents conclusions and further research steps.

II. Related Work

In recent years, the application of Reinforcement Learning (RL) to Energy Management (EM) problems in microgrids attracted some attention. The literature can be classified into single-agent and multi-agent approaches, which is reflected in the following subsections.

A. Single-Agent Reinforcement Learning

The work of Qin et al. [2] proposes a novel privacy preserving load control scheme for a residential microgrid, where a central operator controls a number of smart homes. In fact, preservation of privacy could be achieved with a multi-agent RL approach, where each smart home is represented by an agent. The observation of such an agent would then include privacy information, which is not visible in other agents' observations. However, the authors choose a single-agent reinforcement learning approach, where the microgrid operator is represented as an agent, and the agent is trained with an actor-critic algorithm. The authors argue that the integration

 ISBN: 978-1-61208-967-6

of a recurrent neural network with the deep learning model solves the privacy issue.

Muriithi and Chowdhury [3] deal with EM in a microgrid consisting of a PV installation, a Battery Energy Storage System (BESS) and local loads. The microgrid can exchange energy with the main grid. The EM objective function is the minimisation of energy cost, where a battery degradation cost model is taken into account. Single-agent RL with Q-learning is applied to solve the EM problem, where the actions to be learned by the agent are discrete charging/discharging actions for the BESS.

The microgrid architecture in Ji et al. [4] consists of distributed generators, a BESS, renewable energy production from a PV installation and a wind turbine, and some local consumption loads. The microgrid is connected to the main grid. A deep Q-network algorithm is used to train the microgrid controller agent. The EM objective is to find cost-efficient energy generation schedules for the distributed generators, thus the action space for the agent does not contain demand response actions.

B. Multi-Agent Reinforcement Learning

In Samadi et al. [5], they deal with a microgrid composed of renewable energy sources (wind and PV), an electrical energy storage system, and combined heat and power producers as well as a diesel generator as controllable energy producers. Additionally there are several electrical and thermal energy consumers, respectively. The microgrid is connected to the main grid. The paper proposes a multi-agent, market-based approach to EM, i.e., each energy producer/consumer is represented as an agent that sells/buys energy in the microgrid EM market. The EM goal is then to minimise energy cost for consumers in the microgrid, and Q-learning is used to train the agents towards the EM goal. Agents representing renewable energy sources actually do have no action choices, they just submit the produced energy per time step to the energy market, and thus these agents are not part of training.

The microgrid in Foruzan et al. [6] is composed of energy sellers (PV and wind generators as well as diesel generators) and energy buyers, a storage system and a connection to the main grid. In an auction-based market approach each microgrid component is represented as an agent, and the EM goal of an agent is to maximise its profit. Q-learning is applied to learn optimal agent policies.

In Fang et al. [7], a residential microgrid is modeled as an auction-based marketplace for renewable energy production agents, an agent repesenting a set of residential loads, and an agent representing a fleet of electric vehicles that can serve as storage system for the microgrid. With Q-learning the agents learn to maximise their individual profits, and the overall EM goal is to reach a Nash-equilibrium for the microgrid.

The work of Fang et al. [8] considers a regional microgrid with PV installations, wind turbines and micro turbines as producers, distributed batteries as storage, and industrial and residential loads as consumers. The microgrid is connected with the main grid. The EM problem is modeled as a double auction market, with seller agents (producers), buyer agents (consumers) and management agents being responsible for inter-agent communication and auction clearing. A deep Q-network algorithm is applied to train the non-management agents, where each agent has an individual Q-network, learning Q values for the agent's individual observation and the agents' joint actions as input. An equilibrium selection process in the training process ensures convergence towards a Nash equilibrium with respect to the agents' Q values.

The work of Xu et al. [9] targets a residential microgrid connected to the main grid, with a PV installation and various residential loads, e.g., electric vehicle and air conditioning. The multi-agent EM system consists of four agents, corresponding to non-shiftable loads, time-shiftable loads, power-shiftable loads and electric vehicle load, thus only demand side actions are considered in the EM system. For agent training, a Q-learning algorithm is integrated with a neural network model predicting energy price and PV generation. The reward scheme considers energy cost and so-called dissatisfaction terms penalising load shifting.

C. Discussion

The related work shows that algorithms based on Q-learning are predominantly used when it comes to solve EM problems with RL. The application of policy gradient methods, and here especially actor-critic methods, deserves more attention, as these methods show excellent performance in applications like mastering the games of Go [10] and Dota 2 [11], and solving Rubik's cube with a robot hand [12].

In many cases in the literature the reward schemes are rather simple, as they are used in market-based models for EM with cost minimisation as EM objective. In multi-objective EM the reward scheme has to reflect these objectives, and the impact of such complex reward schemes on RL peformance has to be investigated.

For training an energy management simulation including components with stochastic energy production/consumption characteristics it is essential that the simulation model is trained with a rich variety in data patterns, captured in energy production/consumption profiles. In the majority of contributions in the related work, energy profiles or predictors for renewable energy production and/or consumption are used. However, these profiles/predictors do rather show a limited variety in data patterns. On the one hand the time resolution is rather coarse (1h time steps), and on the other hand the profiles do look smoothened out, not exhibiting the stochastic variations found in raw energy profiles. Based on the above findings, our contribution can be summarised as follows.

1) A multidimensional reward scheme encodes the following EM objectives: a) follow a given energy profile as close as possible (for profile-driven agents), b) load balancing, i.e., matching energy production with consumption, c) minimise energy production from non-renewable sources, and d) correct charging behaviour of BESS: charge if excess renewable energy is available, discharge otherwise.

 ISBN: 978-1-61208-967-6

2) For agent training, we use an extension of Reinforcement Learning library (RLlib)'s implementation of a Proximal Policy Optimisation (PPO) algorithm, which is an actor-critic RL algorithm. The extension consists of a centralised critic approach, where a critic model (implemented as a deep neural network) processes all agents' observations and actions, and the agents share this critic model.
3) For training profile-driven agents, we use energy production and consumption profiles with a 15 min resolution, and due to the real-world character of these profiles, they show a considerable variety in data patterns being observed by the agents.

III. Proposed Approach

In this section, we describe the energy management system to be simulated, followed by a brief introduction into DRL and MARL. The main part of this section is then dedicated to the description of the training environment and the agents that make up the energy management system.

A. Energy Management in a Microgrid

Our work is based on a microgrid with five components: 1) PV energy producer, 2) fully controllable energy producer (e.g., diesel generator), 3) profile-driven energy consumer, 4) BESS, and 5) free acting energy consumer. In contrast to 3), the free consumer is not profile-driven, but it can freely absorb excess production that cannot be absorbed by 3) and 4). Energy management in the described microgrid is a sequential decision-making process. At any time step, the components adjust their loads in pursuing the following goals: a) load balancing, i.e., energy production should match consumption at any time, b) profile following, i.e., components 1) and 3) should follow given energy profiles as close as possible, c) the BESS component should charge if there is more renewable energy available than the profile-driven consumer needs, and BESS should discharge when not enough renewable energy is available to satisfy the profile-driven consumer's demand, d) the fully controllable producer should only produce energy if the profile-driven consumer's demand can not be satisfied by components 1) and 4). The components are configured with max/min State Of Charge (SOC) (for the BESS), max/min load (for all consumers and producers) and max increase/decrease in load from one time step to the next one (for all components).

B. Deep Reinforcement Learning

RL is learning to make decisions from interactions with an environment. Interactions are episodic, leading to a sequential decision-making process. The environment defines an observation space $\mathcal{S}$ and an action space $\mathcal{A}$. In every time step t, the RL agent receives an observation s_t and a reward r_t from the environment and chooses an action a_t, following a policy function $\pi(a_t|s_t)$. The learning goal is to maximize the expected cumulative reward,

$$R_t = \sum_{k=1}^{\infty} \gamma^k r_{t+k+1}, \quad \gamma \in (0, 1]. \tag{1}$$

For a state s, the value of a policy π is defined as: $v_\pi(s) = \mathbb{E}_\pi(R_t|s_t = s)$. Maximizing for the value function also leads to a maximization in the goal sense. We call this policy optimal. Taking action a in a state s leads to the action-value function of a policy π: $q_\pi(s, a) = \mathbb{E}_\pi(R_t|s_t = s, a_t = a)$.

With the latest achievements in Deep Learning (DL), new possibilities in many areas of machine learning arose. Especially the combination of DL and RL, DRL, achieved new impressive results in various fields, e.g., superhuman performance in video games. In DRL deep neural networks are used as function approximators for value and policy functions. Function approximation is crucial for larger spaces of states and/or actions, where a tabular representation is not feasible. It also enables the policy to be optimised directly, by searching in the policy space $\{\pi_\theta(a_t|s_t), \theta\}$ for optimal parameters θ of such a function approximation.

For neural networks, θ are the weights and biases and we can use the gradient ascent method (Baird and Moore [13]) to optimise, leading to a class of algorithms called policy gradient methods. The gradient of an objective function is representing an estimate, to update the parameters. A commonly used objective function for policy gradient methods is (cf. Schulman et al. [14]):

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t[\log \pi_\theta(a_t|s_t)\hat{A}_t] \tag{2}$$

where $\hat{A}_t$ is an estimator of the advantage function, describing the extra reward that could be obtained by taking action a_t.

Combining policy gradient methods with action-value functions leads to actor-critic methods. The actor approximates the policy, and the critic approximates the action-value function, thus criticising the actions taken by the policy.

C. Multi-Agent Reinforcement Learning

A generalisation of RL into multi-agent systems is MARL, where we study how multiple agents learn within a shared environment. A key challenge in MARL is the fact that other agents are part of the training environment, and they are modifying the environment with their actions. The observation that an agent receives does not only reflect the agent's action, but also the actions taken by other agents. In other words: in MARL multiple agents are interacting indirectly through their actions in the training environment. In the energy management case, load balancing requires coordinated actions from all agents. To be able to train coordinated actions, we used RLlib's implementation of a PPO algorithm and extended it with a centralised critic. The usage of a centralised critic approach is inspired by Yu et al. [15].

PPO is a new family of actor-critic methods, proposed by Schulman et al. [14]. With an adaptation of (2), an idea to stabilise training was introduced. The new objective constrains large policy changes, leading to smaller steps and enabling for multiple epochs of mini-batch updates. With the ratio between new and old policy $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{old}}(a_t|s_t)$, the new objective is defined as:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t] \tag{3}$$

 ISBN: 978-1-61208-967-6

where $clip(r_t(\theta), 1-\epsilon, 1+\epsilon)$ clips the ratio to the interval $[1-\epsilon, 1+\epsilon]$.

D. Training Environment and Agents

This subsection describes the development of a multi-agent compatible RL training environment, where the five components described in section III-A are represented as agents. The training objective is then to learn optimal policies to achieve the energy management goals described in section III-A.

1) Training Environment: For the development of the training environment RLlib's [16] multi-agent environment has been bootstrapped, which makes it compatible with OpenAI gym environments. The environment uses Box observation spaces and discrete action spaces for the agents. The decision to use discrete action spaces with PPO algorithm has been inspired by Tang and Agrawal [17], who state, "the discrete policy achieves significant performance gains with state-of-the-art on-policy optimization algorithms PPO". Tang and Agrawal [17] also give an optimum number of discrete sampling of a continuous action space which is (7-15) and as per our experiments 11 discrete actions gave the best results.

2) Agents: In the following agent configuration characteristics are described. For a profile-driven agent, the energy profile and the profile-corridor, introducing some tolerance for deviating from the profile, are important characteristics, while for BESS their initial SOC and minimum SOC play an important role. For all the agents, their max-load-diff denotes the maximum load difference between consecutive time steps, thus max-load-diff determines the agent's maximum speed of reaction. Another important configuration parameter for all agents is the load-balancing tolerance. If the absolute difference between total production and total consumption is smaller than the tolerance, load-balancing is achieved. The observation of an agent is composed of four time series with five time steps each, see Figure 1.

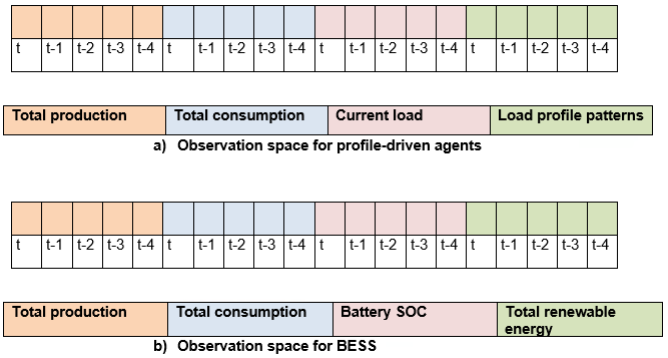


Fig. 1: Observation space for agents.

In the following the agents are specified in more detail.

- PV producer agent:
  This agent is a profile-driven agent that follows the energy production profiles of a PV panel, with a small tolerance with which the agent is allowed to deviate from its specific profile. The agent has a 20-dimensional observation space represented by Figure 1 (a), and a discrete action space of 11 non-negative numbers, with (0-4): decrease production load, 5: do nothing and (6-10): increase production load. The effective increase/decrease of production load is then (0.2, 0.4, 0.6, 0.8, 1.0) * max-load-diff.
- Profile-driven consumer agent:
  This agent is a profile-driven agent that follows power consumption profiles. The profile following tolerance, observation space and action space are the same as for the PV agent.
- BESS agent:
  This agent simulates the behaviour of a battery storage, where the main aim is to charge/discharge appropriately. It has a 20-dimensional observation space represented by Figure 1 (b), an action space of 11 non-negative numbers, with (0-4): battery discharges, 5: battery does nothing and (6-10): battery charges. In the BESS agent's context, max-load-diff is the maximum charging/discharging rate of a battery. The effective charging/discharging rate is called the battery magnitude and is calculated as (0.2, 0.4, 0.6, 0.8, 1.0) * max-load-diff. The BESS agent is further specified with initial battery SOC, minimum SOC and maximum SOC. The setting of maximum SOC value is decided based upon total renewable energy available after consumption has been satisfied. The setting of initial SOC is based upon the energy required for consumption at the initial part of the episode. Minimum SOC is chosen randomly, but it can be chosen considering safe operation of the battery.
- Fully controllable producer agent, free acting consumer agent:
  The observation space and the action space are the same as for the PV producer agent. An important configuration parameter for the fully controllable producer is the maximum power output that can be delivered into the microgrid. Profile-following is not of concern for these agents.

3) Deep Reinforcement Learning Model: The current implementation adapts the PPO implementation of RLlib, such that the agents share a centralised critic model. Figure 2 shows the DRL model used for each agent. The actor model has three layers, with the action logits in the final layer. The centralised critic model for each agent has three input layers. To understand the input layers let us assume that we have n agents. The first input layer corresponds to the agent's own observation with shape $(, 20)$, the second input layer processes the opponent agents' observations with shape $(, 20 * (n-1))$, and the third layer processes the opponent agents' actions with shape $(, 11 * (n-1))$. The three input layers are concatenated, followed by two dense hidden layers. The final layer outputs a single value, indicating how good the input is in terms of cumulative rewards over an episode. For all layers the first dimension in $(, \text{size})$ is not specified, as it depends on the mini-batch size used by the PPO algorithm.

 ISBN: 978-1-61208-967-6

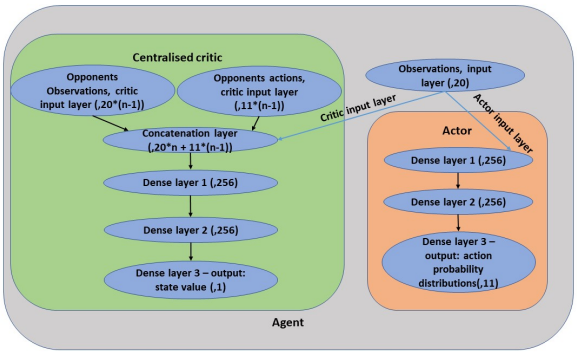


Fig. 2: DRL model.

4) Reward Scheme for Agents: The reward scheme for the agents in our energy management simulation is as follows:

- Profile-driven producer and consumer:
  For profile-driven agents there are only two penalties or rewards:
  - Profile deviation penalty

    L Current Agent Load
    PL Current Agent Profile Load
    $CORR$ profile corridor tolerance
    if abs(L - PL) <CORR **then**
    No Penalty
    else
    Penalty = abs(L - PL) * const-penalty
    end if
  - Load balancing reward

    TP Total Production
    TC Total Consumption
    if abs(TP -TC) <balance-tolerance **then**
    Reward = constant
    else
    No Reward
    end if
- Fully controllable producer and free acting consumer:
  For the fully controllable producer and free acting consumer agents there are three penalties and rewards and they are as follows:
  - Load balancing reward
  - Excess-energy penalty and appropriate-energy reward

    R Renewable energy
    C Consumption, only profile-driven consumer
    BM Battery Magnitude charging/discharging
    BM >0.0 {charging}, BM <0.0 {discharging}

    FCP Load of fully controllable producer
    FCC Load of free acting consumer
    EE = R - C - BM Remainder energy after battery charges/discharges
    if R - C >0.0 **then** {Excess renewable energy is available}
    if FCP >0.0 **then** {FCP should not get greater than zero}
    penalty $\propto$ FCP
    end if
    if abs(FCC - EE) >balance-tolerance **then** {FCC should equate to remainder energy EE}
    penalty = constant
    else
    reward = constant
    end if
    else {No excess renewable energy is available}
    if FCC >0.0 **then** {FCC should not get greater than zero}
    penalty $\propto$ FCC
    end if
    if abs(FCP -EE) >balance-tolerance **then** {FCP production equates to remainder energy EE}
    penalty = constant
    else
    reward = constant
    end if
    end if
- BESS:
  For the BESS agent there are three penalties and rewards and they are as follows:
  - Load balancing reward
  - Correct charging/discharging behaviour reward and incorrect behaviour penalty
    This reward scheme, allows to select a specific action from the action sets (0,10). The battery magnitude for discharging actions (0,4) is (-0.2,-0.4,-0.6,-0.8,-1.0) * max-load-diff, and for charging actions (6,10) it is (0.2, 0.4, 0.6, 0.8, 1.0) * max-load-diff, at action 5 the battery does nothing.
    if consumption >PV production **then**
    action(0,4) is rewarded. The reward favours the action with the best effect on load balancing.
    else if PV production >consumption **then**
    action (6,10) is rewarded. The reward favours the action with the best effect on load balancing.
    else
    action 5 is rewarded
    end if

IV. Results

This section describes results for training the EM simulation with the proposed reward scheme. The results have been achieved in the context of the projects RESINET and Zer0p.

 ISBN: 978-1-61208-967-6

A. Experiment Setup

The experimental setup is as follows. The PV producer agent and the profile-driven consumer agent are trained each with a single profile. The profiles are taken from real-world data for large-scale PV installations and industrial consumers in Austria, with maximum consumption loads at 1700 kW, and maximum production loads at 600 kW. For our experiments, we scaled down the power values of the producer by 10^5 and consumption by $6 * 10^5$. The decision to scale down the consumption further by a factor of 6 was taken to create a scenario where charging of the battery is feasible. Each profile has data of 96 time steps covering one day, which is then the episode length for RL training. The profile-corridor parameter for the profile-driven agents has been kept really small (0.0001), so that the agents are forced to follow their respective profile as close as possible. The load-balancing tolerance is set up as 0.5. For the BESS agent, the maximum SOC is 60, and both the initial battery SOC and the minimum battery SOC are at 2% of the maximum value. The max-load-diff parameters are set as follows. BESS: 2.5, PV producer: 0.4, profile-driven consumer: 0.25, fully controllable producer and free acting consumer: 0.5. With the above configuration, the agents are trained for 20,000 episodes. Figure 3 illustrates the training progress with different reward curves, x-axis units are time steps.

- rew_charging_mean: reward curve for correct charging/discharging behaviour of the BESS agent
- rew_dev_profile_mean: reward curve for profile following
- rew_total_load_mean and rew_total_out_mean : reward curve for load balancing
- rew_exp_energy_mean: reward curve for fully controllable producer
- rew_episode_reward_mean: total reward curve

B. Evaluation

Figures 4a and 4b illustrate the performance of the trained energy management simulation, x-axis units are time steps, y-axis units are arbitrary power units resulting from scaling down the real-world data. Figure 4b shows that load balancing is achieved quite accurately, plotting total energy production vs. total energy consumption. In Figure 4a the load curves of the individual agents are plotted, as well as the profiles for the profile-driven agents. The figure shows that profile-following is achieved quite accurately, both for the profile-driven consumer and the PV producer. The BESS agent shows proper charging behaviour: the agent does nothing in the initial part of the episode when there is no renewable power available, in the middle part it is charging when renewable power is available, and it discharges in the last part when there is only very little output from the PV producer. The fully controllable producer produces sufficient energy in the initial and last part of the episode to fulfill consumption demand, and it outputs near zero energy when renewable energy is available. The free acting consumer only consumes energy in the middle of the episode, where there is remainder energy from the PV producer that cannot be consumed by the profile-driven consumer and the BESS.

C. Scalability

In multi-agent systems scalability issues are of interest, where it is investigated how the number of agents impacts system performance. So far, our experiments with respect to scalability have been limited. For these experiments we used three agent types: PV producer, profile-driven consumer and fully controllable producer. We investigated different system configurations (I, J, K) where I denotes the number of PV producer agents, J denotes the number of profile-driven consumers and K denotes the number of fully controllable producers. We performed experiments with the following configurations: (1,1,1), (2,2,2), (5,5,3) and (5,5,5). We found that all configurations resulted in proper profile following and load balancing. Table I provides an overview of the conducted scalability experiments in terms of number of episodes used for training and training time on a state-of-the-art desktop PC.

TABLE I: SCALABILITY RESULTS

Configuration	Number of agents	Number of episodes	Training time
(1,1,1)	3	8000	40mins
(2,2,2)	6	16000	1hr 20mins
(5,5,3)	13	60000	4hr 30mins
(5,5,5)	15	60000	5hr 15mins

V. Conclusion

The results show that it is possible to train an energy management simulation for a microgrid, leveraging data for renewable energy production profiles and consumption profiles in the training process. A crucial point was the development of an appropriate reward scheme, enabling to learn the key desired agent behaviours ”profile following” and ”load balancing” in an energy management case with five agent types: profile-driven producer/consumer, fully controllable producer, free acting consumer and storage. In the following we outline our further research steps.

1) Quantitative analysis of simulation performance: for evaluation of the trained simulation we currently rely on a visual analysis of the results of simulation runs. For a quantitative analysis a metric has to be developed, measuring the performance of the multi-agent system when running a simulation. We envisage the application of simple statistical methods such as *Mean Absolute Error* and *Root Mean Square Error*, calculated over all time steps of a simulation run with respect to the desired behaviours ”profile following” and ”load balancing”.
2) Multi-profile training: so far, we have only considered single-profile training, i.e. the renewable energy producer and the profile-driven consumer, respectively, have been trained with one profile each. With multi-profile training, the goal is to achieve flexible agent behaviour, i.e., the trained behaviours should be able to cope with various situations. A situation is characterised by a

 ISBN: 978-1-61208-967-6

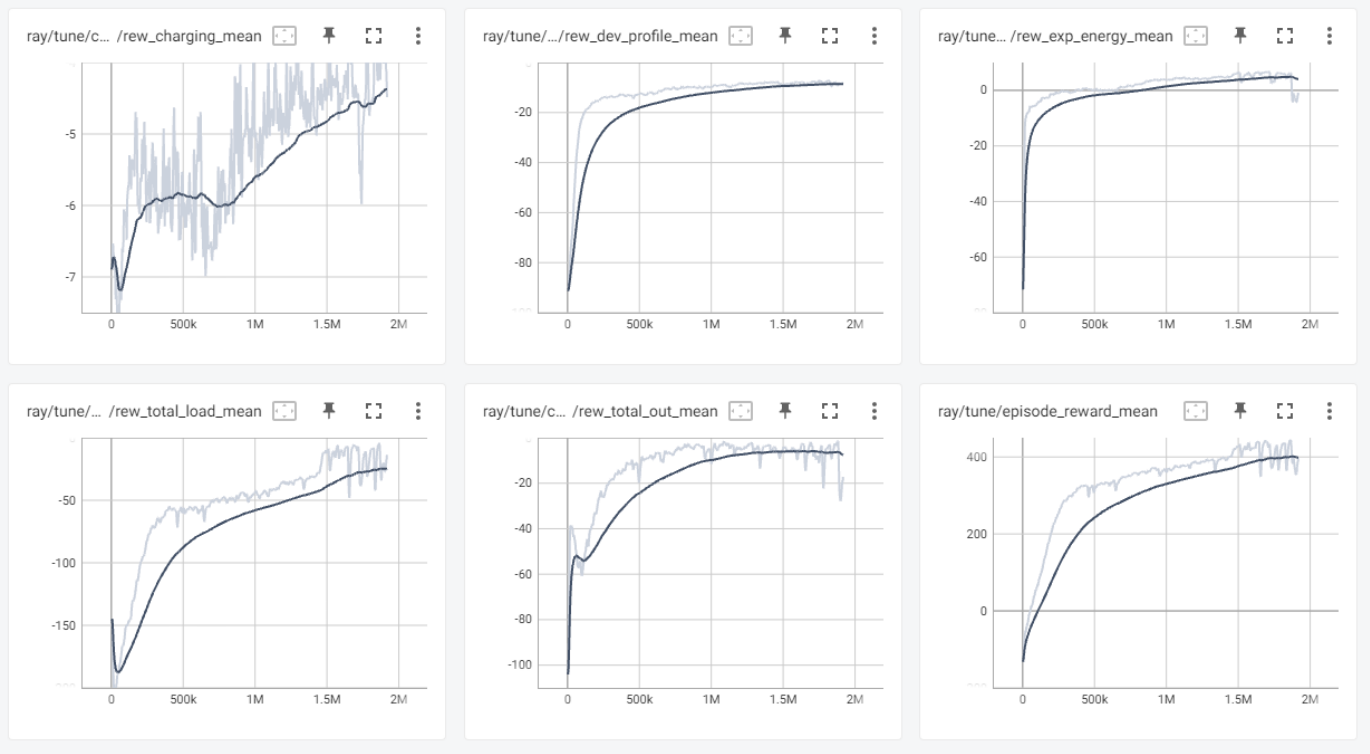
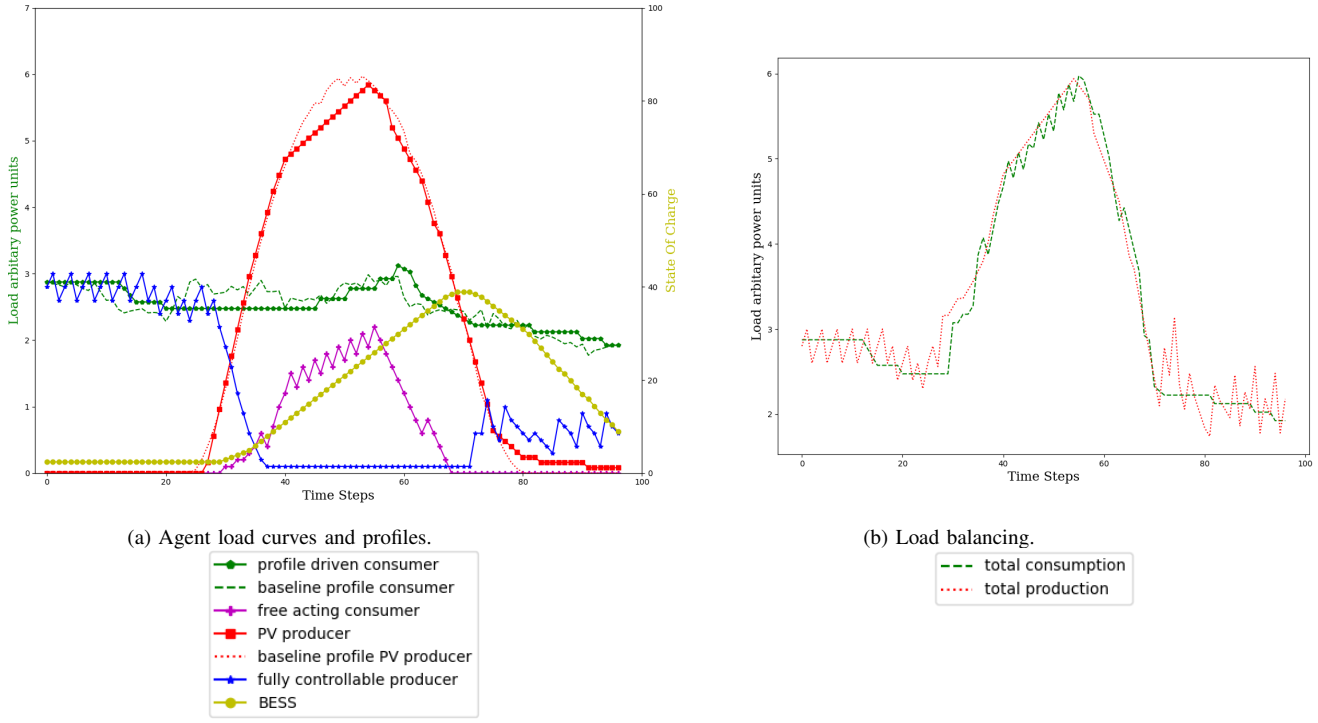


Fig. 3: Training progress.



(a) Agent load curves and profiles.

(b) Load balancing.

Fig. 4: Simulation results.

concrete pair of energy profiles, one for the renewable energy producer, the other one for the profile-driven consumer. To what extent such flexibility can be achieved is a future research question.

3) Up-scaling the number of agents: in future experiments we will intensify our research with respect to up-scaling the number of agents in the energy management system, thus increasing system size. We will investigate the question, how system size does influence training and simulation performance.

ACKNOWLEDGMENT

The research has been funded by the European Commission and the Government of Upper Austria through the program EFRE / IWB 2020, the priority REACT-EU and the project RESINET (RESIlienzsteigerung In energieNETzen – RESilience increase In energy NETworks; Nr.:WI-2020-701900/12). This work also received funding from the Government of Upper Austria through the program "Programm des Landes Oö zur Stimulierung der Erschließung/Erweiterung von zukunftsweisenden Forschungsfeldern bei den Oö. außeruniversitären Forschungseinrichtungen im Zeitraum 2022-2029" and the project Zer0p (Zero Defect Manufacturing für nachhaltige Produktion; Nr.:WI-2021-299762/12-Au).

REFERENCES

[1] UNIDO, "Green Industry." [Online]. Available: https://www.unido.org/our-focus/cross-cutting-services/green-industry

[2] Z. Qin, D. Liu, H. Hua, and J. Cao, "Privacy Preserving Load Control of Residential Microgrid via Deep Reinforcement Learning," *IEEE Transactions on Smart Grid*, vol. 12, no. 5, pp. 4079–4089, Sep. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9451164/

[3] G. Muriithi and S. Chowdhury, "Optimal Energy Management of a Grid-Tied Solar PV-Battery Microgrid: A Reinforcement Learning Approach," *Energies*, vol. 14, no. 9, p. 2700, May 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/9/2700

[4] Y. Ji, J. Wang, J. Xu, X. Fang, and H. Zhang, "Real-Time Energy Management of a Microgrid Using Deep Reinforcement Learning," *Energies*, vol. 12, no. 12, p. 2291, Jun. 2019. [Online]. Available: https://www.mdpi.com/1996-1073/12/12/2291

[5] E. Samadi, A. Badri, and R. Ebrahimpour, "Decentralized multi-agent based energy management of microgrid using reinforcement learning," *International Journal of Electrical Power & Energy Systems*, vol. 122, p. 106211, Nov. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0142061520304877

[6] E. Foruzan, L.-K. Soh, and S. Asgarpoor, "Reinforcement Learning Approach for Optimal Distributed Energy Management in a Microgrid," *IEEE Transactions on Power Systems*, vol. 33, no. 5, pp. 5749–5758, Sep. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8331897/

[7] X. Fang, J. Wang, G. Song, Y. Han, Q. Zhao, and Z. Cao, "Multi-Agent Reinforcement Learning Approach for Residential Microgrid Energy Scheduling," *Energies*, vol. 13, no. 1, p. 123, Dec. 2019. [Online]. Available: https://www.mdpi.com/1996-1073/13/1/123

[8] X. Fang, Q. Zhao, J. Wang, Y. Han, and Y. Li, "Multi-agent Deep Reinforcement Learning for Distributed Energy Management and Strategy Optimization of Microgrid Market," *Sustainable Cities and Society*, vol. 74, p. 103163, Nov. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2210670721004443

[9] X. Xu, Y. Jia, Y. Xu, Z. Xu, S. Chai, and C. S. Lai, "A Multi-Agent Reinforcement Learning-Based Data-Driven Method for Home Energy Management," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3201–3211, Jul. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8981876/

[10] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[11] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse *et al.*, "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

[12] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas *et al.*, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.

[13] L. Baird and A. Moore, "Gradient Descent for General Reinforcement Learning," in *Advances in Neural Information Processing Systems*, M. Kearns, S. Solla, and D. Cohn, Eds., vol. 11. MIT Press, 1998. [Online]. Available: https://proceedings.neurips.cc/paper/1998/file/af5afd7f7c807171981d443ad4f4f648-Paper.pdf

[14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *CoRR*, vol. abs/1707.06347, 2017, arXiv: 1707.06347. [Online]. Available: http://arxiv.org/abs/1707.06347

[15] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu, "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games," *arXiv:2103.01955 [cs]*, Jul. 2021, arXiv: 2103.01955. [Online]. Available: http://arxiv.org/abs/2103.01955

[16] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg, J. Gonzalez, M. Jordan, and I. Stoica, "RLlib: Abstractions for Distributed Reinforcement Learning," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, Jul. 2018, pp. 3053–3062. [Online]. Available: https://proceedings.mlr.press/v80/liang18b.html

[17] Y. Tang and S. Agrawal, "Discretizing Continuous Action Space for On-Policy Optimization," *arXiv:1901.10500 [cs]*, Mar. 2020, arXiv: 1901.10500. [Online]. Available: http://arxiv.org/abs/1901.10500

 ISBN: 978-1-61208-967-6

Performance of Linear Programming in Optimizing the Energy Schedule of a Grid-connected Hybrid System Compared to Particle Swarm Optimization

Hoda Elaouni
LUSAC Laboratory, University of Caen Normandy
Orange Innovation
Lannion, France
email: hoda.elaouni@unicaen.fr

Hussein Obeid
LUSAC Laboratory, University of Caen Normandy
Cherbourg, France
email: hussein.obeid@unicaen.fr

Stéphane LE Masson
Orange Innovation
Lannion, France
email: stephane.lemasson@orange.com

Hamid Gualous
LUSAC Laboratory, University of Caen Normandy
Cherbourg, France
email: hamid.gualous@unicaen.fr

Olivier Foucault
Orange Innovation
Lannion, France
email: olivier.foucault@orange.com

***Abstract*—This study aims to find a rapid and efficient method for managing the energy of a Grid-connected hybrid system. Thus, two optimization strategies, the Linear Programming (LP) and the Particle Swarm Optimization (PSO), have been suggested to minimize the operating cost of the hybrid system while respecting the constraints of all the system components. Then, a comparative study has been made between these two methods (i.e., LP and PSO). Consequently, the operating cost obtained using PSO algorithm is close to the one provided by the LP algorithm. However, the PSO algorithm is slower than the LP algorithm and requires different parameters to be chosen. Finally, the impact of the battery initial state of charge on the operating cost is studied.**

Keywords*—Linear Programming; Particle Swarm Optimization; Optimization; Grid-connected hybrid system.

I. Introduction

The energy consumption of access networks represents a principal part of telecommunications operators' energy bills. Several works [1] [2] have been initiated on sources, energy storage, and their management to reduce this consumption and the carbon footprint. Green production is a promising way to overcome this fossil energy issue [3] [4]. Besides, it is necessary to develop acceptable management methods and technical tools guaranteeing network reliability [5]. In this perspective, the notion of microgrid has appeared to resolve part of this management problem. Indeed, it is an intelligent system composed of green and local production as well as a storage system to ensure the reliability of the system. An energy management system provides an optimal configuration and sizing with the economic management of exchanged energy within. The maximization of economic efficiency and reliability is undoubtedly the top of all research targets [6].
Different studies have been made to achieve this purpose. For instance, in [6] a multi-objective optimization problem of optimizing the schedule of sources, as well as the import/export power with the grid, has been solved using an optimization-based approach called Branch and Bound method. In addition, optimal energy management of microgrid, which constitutes of a PV system and a storage system with minimum of cash flow using dynamic programming technique has been suggested in [7] [8]. A comparative study has been presented in [9] to illustrate the efficiency of Linear Programming (LP) compared to PSO and adaptive dynamic programming for an intelligent home energy resources scheduling in the presence of uncertain data. Hossain et al. [10] present a particle swarm optimization for real-time application energy management to find optimal battery control of a community microgrid. In [11], a fuzzy logic-based energy management system for a residential grid-connected system including renewable energy sources and storage capability is suggested. The difference between this study and the studies cited above occurs in the problem formulation and the constraints to be respected. To optimize the energy scheduling/management in a connected microgrid, there are two types of methods: an exact optimization methods that guarantee finding an optimal solution (e.g., LP) and heuristic optimization methods that don't guarantee that the solution founded is optimal (e.g., PSO). In this study, both algorithms LP and PSO are applied to find the optimal energy scheduling of a grid-connected hybrid system. Furthermore, three different scenarios are considered to provide a comparison between these two algorithms. It is shown that the LP algorithm is faster and does not require parameters to be tuned which is not the case for PSO algorithm. On the other hand, the impact of the initial state of charge of the battery on the operating cost is studied for the LP algorithm.

The work is organized as shown: system description and energy models of the architecture components are introduced in Section II. These models will be used to calculate the required parameters for optimization approaches. Section III

 ISBN: 978-1-61208-967-6

suggests two different methodologies for optimizing energy planning with a minimum operating cost. In this section, we introduce the objective function and constraints adopted in each method. The results obtained by these approaches are compared in Section IV. In addition, some assessment has been established to show the robustness of the proposed strategy.

II. System Description

The architecture studied is a grid-connected hybrid system composed of renewable energy sources, i.e., Photovoltaic and Wind Turbine, batteries, and DC load as shown in Figure 1. To reduce the energy loss, we assume that the battery should be charged only by the remaining energy, this means when the energy produced by renewable sources is greater than the load demand. Otherwise, the energy left will be exchanged with another local site. On the other hand, if the consumption exceeds the production of the renewable source, the battery will discharge to meet the remaining energy.

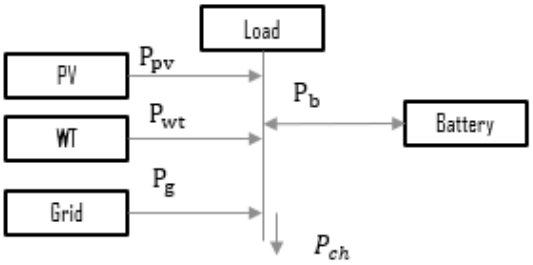


Figure 1. Grid-connected hybrid system architecture

A. Photovoltaic model

The mathematical model for estimating the output power of a PV module is a linear function of the solar radiation and the ambient temperature [12]. It can be calculated as follows:

$$P_{pv} = P_p f_c G(\frac{1+\beta(T_c - T_{ref})}{G_r}) \quad (1)$$

$$T_c = T_a + G(\frac{(NOCT-20)}{800}) \quad (2)$$

where P_p is the rated power under standard test conditions (kW), f_c is PV derating factor (88%), G is solar radiation (W/m^2), β is the temperature coefficient of efficiency (-0.41%/°C), G_r is the standard amount radiation (1000 W/m^2), T_{ref} is standard test temperature (25°C), T_c is the cell temperature (°C), T_a is the ambient temperature (°C) and NOCT is the nominal operating cell temperature (45°C).

B. Wind Turbine model

To model the wind turbine, the mechanical power, which is directly extracted from it, can be given by [13]

$$P_{wt} = \begin{cases} 0.5 C_p S \varphi V^3, & \text{if } V_i \le V \le V_n \\ P_r, & \text{if} V_n \le V \le V_o \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where V_n, V_i and V_o are the rated (11m/s), the cut-in (3.5m/s), the cut-out (25m/s) wind speeds respectively, C_p is the power coefficient, S is the turbine blades swept area (10.87m^2), φ is the air density (1.225kg/m^3), V is the wind speed at hub height H and P_r is the rated power. For the purpose of adjusting the wind profile according to the height, the following equation can be used [14]

$$V = V_0(\frac{H}{H_0})^{\alpha} \quad (4)$$

where V_0 is the wind speed measured at the reference height H_0. α is the power law exponent depends on the nature of terrain (0.14).

C. Battery model

Batteries are used to store excess power in the microgrid and operate when the system has deficit power. At any hour, the battery stored energy is related to the previous one and the energy production and consumption situation of the system during the time from $t-1$ and t as used in [10].

- Charging mode

$$W_b(t) = W_b(t-1) + (W_{pv} + W_{wt} - W_l) \quad (5)$$

- Discharging mode

$$W_b(t) = W_b(t-1) - (W_l - W_{wt} - W_{pv}) \quad (6)$$

where W_l is the energy consumption (kWh). W_{pv}, W_{wt} represent the energy production by Photovoltaic module and Wind turbine (kWh) respectively.

III. Optimization algorithm

The main goal of this paper is to minimize the operating cost of the energy exchanged with the grid to obtain an optimal energy schedule of the grid-connected hybrid system. The optimization algorithm should ensure that the discharging of the battery will be done during high demand, while the charging will be done during high production, moreover, the state of charge should be within upper and lower limits. In addition, the battery must return to its initial state of charge at the end of the optimization horizon to ensure that the system has a stabilized energy balance for one cycle. To solve this issue, we presume two methods of optimization explained below:

A. Linear programming

This approach is based on the linear programming paradigm that consists in minimizing or maximizing a given function according to the following constrained scheme [9]:

$$\begin{cases} \max f(x) = c^T x \text{ or } \min f(x) = c^T x \\ \text{subject to: } \quad Ax \le b \; or \; Ax \ge b \; or A_{eq} x = b_{eq} \\ \text{Where: } \quad x \ge 0, x \in \mathbb{R}^{n\times 1}, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^{m\times 1}, \\ c \in \mathbb{R}^{n\times 1}, A_{eq} \in \mathbb{R}^{p\times n}, b_{eq} \in \mathbb{R}^{p\times 1} \end{cases} \quad (7)$$

In this approach, the decision variables for the economic dispatch problem are as follows:

- The battery energy of charging and discharging (W_c and W_d).

- The exchanged energy with the grid utility (W_g).
- The exchanged energy with another site (W_{exch}).
- The battery state of charge (SOC).

This means, in our case the vector x is chosen as follows: $[W_g, \ W_c, \ W_d, \ SOC, \ W_{exch}]^T$. The vector c is formed using (8)-(9). The right part of (10)-(12) represents the elements of the matrix A_{eq}, and the left part of these equations are used to form the vector b_{eq}. Matrix A contains the coefficients multiply the decision variables in (13)-(18). The vector b includes the upper and the lower limits of each variable.

The objective function in (8) aims to minimize the cost of the energy purchased from the grid W_g when the consumption is greater than the production as well as make profit by exchanging the remaining energy with other site W_{exch}. And, C_g in (8) represents the purchased energy price (0.2€/kWh).

$$\min C = \sum_{i=1}^{T} W_g(i) C_g k(i) \tag{8}$$

Where T is the energy management system period (e.g., one day) and i is time interval (e.g., 1h). The parameter k is a binary variable for the charging state of the battery. In fact, k represents two constraints to be respected which are the battery should not be discharged when the system has an excess energy and vice versa. k is calculated in terms of the load demand and renewable energy production, as described in (9), where W_l, W_{pv} and W_{wt} are the load, solar PV and Wind Turbine energy (kWh)

$$k(i) = \begin{cases} 1, & \text{if } d(i) \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

$$d(i) = W_l(i) - W_{pv}(i) - W_{wt}(i) \tag{10}$$

Equation (11) guarantees that the battery will be charged mostly by the renewable energy.

$$d(i) = k(i)(W_g(i) - W_d(i)) + (k(i) - 1)(W_c(i) - W_{exch}(i)) \tag{11}$$

Equation (12) calculates the battery state of charge SOC in each slot time i to maintain its values within the given limitations in (13).

$$SOC(i+1) = SOC(i) + \frac{(1 - k(i))W_c + k(i)W_d}{E_c} \tag{12}$$

$$SOC_{min} \leq SOC(i) \leq SOC_{max} \tag{13}$$

Where E_c is the nominal energy of the battery[kWh]. The inequalities (14)-(17) indicate the lower and upper bounds that should be respected for the exchanged energy with the grid, the external consumers, and the battery, respectively.

$$0 \leq W_g(i) \leq W_{gmax} \tag{14}$$

$$W_{exch}(i) \geq 0 \tag{15}$$

$$0 \leq P_c(i) \leq P_{cmax} \tag{16}$$

$$P_{dmax} \leq P_d(i) \leq 0 \tag{17}$$

The ε refers to an admitted tolerance in the constraints concern the charging of the battery in the end of the period of the optimization T.

$$|SOC(T) - SOC(1)| \leq \varepsilon \tag{18}$$

B. Particle swarm optimization

Its concept is based on the behavior of birds to compute global optimization functions [10]. In PSO, each possible solution is modeled as a particle that moves through the input hyperspace, which can have numerous dimensions [15]. First and foremost, each solution takes a random position with a random velocity in the search space. At each iteration, the particles move towards their best position, and therefore that of their neighborhood, which corresponds to the optimum position, by updating their velocity [16]. In this section, the objective is similar, whereas the objective function has presented differently. The first term in (19) refers to a penalty applied in the case of the battery charged with the grid. The decision variable in this methodology is only the exchanged energy with the battery. For this strategy, the cost function is described by (19) to reduce the electricity bill by minimizing the exchanged energy with the grid. Indeed, according to the difference between the load demand and the renewable energy production, as well as the sign of the decision variable, the penalty is applied to avoid the following scenarios:

- Charging or discharging the battery when the system is in a steady state.
- Discharging the battery even if there is an excess of energy.
- Energy left is not sufficient to charge the battery.

$$\min C = P + \sum_{i=1}^{T} W_g(i) C_g \tag{19}$$

The grid will meet the load when the consumption is greater than the production. On the other hand, if the suppliers transcend the load demand, the energy left will be exchanged with other consumers.

$$d(i) = W_l(i) - W_{pv}(i) - W_{wt}(i) \tag{20}$$

$$W_g = \begin{cases} W_l(i) - d(i) + W_b(i), & \text{if } d(i) \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

$$W_{exch} = \begin{cases} -d(i) - W_b(i), & \text{if } d(i) \leq 0 \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

At each point of time, the program ensures that the solution respects the constraints presented as follows:

$$P_{bmin} \leq P_b(i) \leq P_{bmax} \tag{23}$$

$$SOC_{min} \leq SOC(i) \leq SOC_{max} \tag{24}$$

$$0 \leq W_g(i) \leq W_{gmax} \tag{25}$$

$$W_{exch}(i) \geq 0 \tag{26}$$

$$|SOC(T) - SOC(1)| \leq \varepsilon \tag{27}$$

IV. Simulation and discussion

The considered system includes a DC load with a constant rated power of 5 kW, a PV with an installed peak power of 69 kW, a Wind turbine with a rated power of 16 kW, and a battery with a rated energy of 74 kWh.

 ISBN: 978-1-61208-967-6

TABLE I. PSO PARAMETERS

PSO	Number of variables	24
	Number of iterations	300
	population size	1000
	Inertia coefficient	1
	Damping ratio of inertia coefficient	0.99
	Personal acceleration coefficient	2
	Social acceleration coefficient	2

A. Comparative study

In this subsection, the performance of both algorithms to provide the energy management for this aforementioned system is compared through simulation results using Matlab. Indeed, three scenarios have been tested (see Figures 2-4) to find which approach is more efficient regarding some parameters such as the operating cost, the computational time, and the energy exchanged with other site. These scenarios present the meteorological data at Lannion for three different months July, May and October. In the simulations which follow, the period is fixed as 24 hours, the initial state of charge is chosen as $SOC(1) = 80\%$, and the error between the final and the initial values of SOC is taken as 3%, i.e., $\varepsilon = 3\%$ in Eqs.(18) and (27). Moreover, the values of $SOC(min)$ and $SOC(max)$ have been selected as SOC(min)=30% and SOC(max)=100%. The optimization parameters for PSO algorithm are given in Table I. In Figures 2-4, it can be observed that the energy dispatch proposed by both methods for the three scenarios, are globally similar. Indeed, when the PV and Wind turbine production is more important than the load demand, the two approaches suggest to charge the battery and transfer the remaining energy to other consumers. But, the difference occurs when the production is less than the demand. In this case, PSO algorithm proposes to use the grid and the battery to meet the load demand. Conversely, linear programming suggests meeting the load by discharging only the battery. Besides, the two strategies respect obviously the constraints about the final value of the battery state of charge in all cases. Regarding the computational time, the linear programming can find the optimal solution within one minute. However, the PSO algorithm takes more than one hour to find it, since its convergence depends on the number of iterations and population size which have been chosen big enough. Here, it should be mentioned that if the number of iterations and population size have not been adequately chosen, the convergence of PSO algorithm cannot be ensured. The operating cost proposed by LP is less expensive than the one proposed by PSO algorithm for all scenarios, as it can be shown in TableII. On the other hand, the PSO algorithm offers to exchange more energy compared to other one since the battery discharges less than the first technique as shown in the Figures 2-4.

To sum up, the LP finds quickly and efficiently the optimal schedule of the considered system compared to the PSO algorithm. Furthermore, as soon as the numbers of the decision variables increase, the use of the PSO becomes avoidable. That is due to the reason that, the PSO algorithm requires a lot of parameters to be tuned.

TABLE II. OPERATING COST AND EXCHANGED ENERGY PROPOSED BY TWO APPROACHES.

Scenario	**PSO**		**LP**	
	C (€)	W_{exch} (kWh)	**C** (€)	W_{exch} (kWh)
Case 1	1.85	32	0.68	24
Case 2	2.61	27	1.18	18
Case 3	2.38	18.5	1.2	10

Where case 1, case 2, and case 3 represent the meteorological data in July, May, and October, respectively.

B. Sensitivity analysis

In this part, some parameters will be analyzed using linear programming. It treats the impact of the initial state of charge on the operational cost and the energy sold to other sites over 24h.The values of the SOC_{min}, SOC_{max}, and ε are similar to those used in the subsection A. However, the data that will be used in this subsection are presented in Figure 2. Table III shows that the minimum operational cost has been obtained in the case where the battery initial level of energy is 70%. Indeed, the purchased energy from the grid is zero as well as the 70% of energy stored is sufficient to meet the load and respect the constraint about the final value of the SOC. Consequently, the exchanged energy with the other consumers is the minimum because the most excess energy is used to charge the battery. At the beginning of the optimization, if the battery is fully charged or discharged, the operational cost and the exchanged energy for the solution obtained would be greater than the other scenarios. In other words, the battery is less used in these cases in order to respect the constraint about the final SOC. Figure 5, it represents the energy schedule of the system aforementioned considering the optimal value of the $SOC(t_0)$. As can be shown in the Figure 5, the battery has completely discharged in the state of deficit when the renewable sources production is insufficient. Besides, it is remarkable that the excess energy has been sufficient to charge completely the battery. For that reason, the system has respected the constraints without using the grid energy to meet the load like the other cases. To conclude, the economic scenario to adopt is with SOC(t_0) equals to 70% since the energy exchanged with the grid in this case is zero.

TABLE III. OPERATING COST AND EXCHANGED ENERGY FOR DIFFERENT VALUES OF THE INITIAL BATTERY STATE OF CHARGE USING LP.

SOC_{int} (%)	**100**	**90**	**80**	**70**	**60**	**50**	**40**
C (€)	2.61	1.64	0.68	0	0.46	1.42	2.39
W_{exch} (kWh)	39	31	24	19	22	30	37

 ISBN: 978-1-61208-967-6

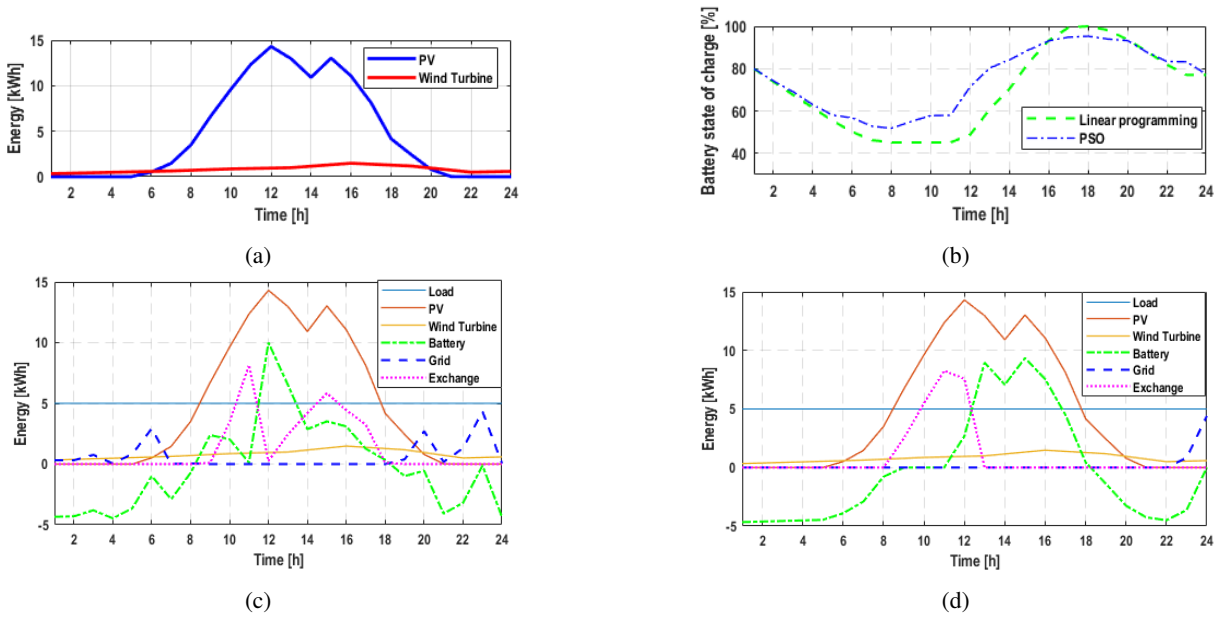


Figure 2. (a) The PV and Wind turbine energy profile on 1st of July in Lannion. (b) The variation of the battery state of charge proposed by PSO and LP. The energy scheduling on 1st of July using: (c) PSO and (d) LP.

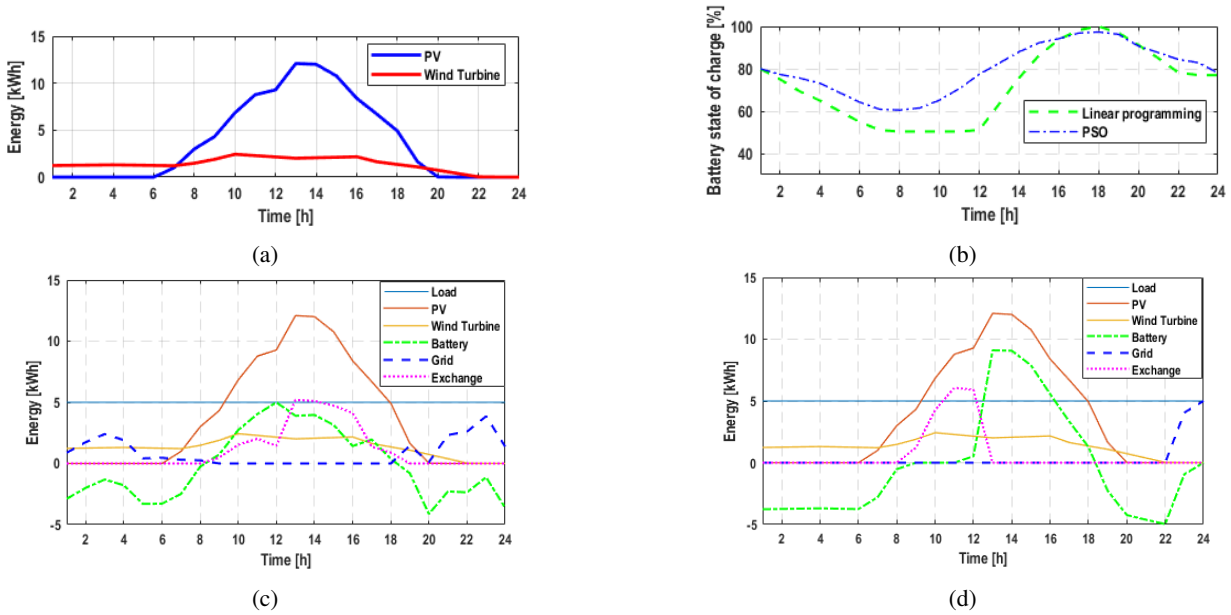


Figure 3. (a) The PV and Wind turbine energy profile on 1st of May in Lannion.(b) The variation of the battery state of charge proposed by PSO and LP. The energy scheduling on 1st of May using: (c) PSO and (d) LP.

V. Conclusion

This paper has applied two approaches for the optimal energy scheduling of a Grid-connected hybrid system which are the LP and the PSO algorithms. Then, a comparison has been made to confirm the effectiveness and the rapidity of the LP in front of the PSO algorithm in terms of computational time and operational cost. Moreover, it is shown that the PSO algorithm requires some parameters to be tuned to achieve the convergence which is not the case for the LP.

On the other hand, a sensitivity analysis for the LP has been studied also. The obtained results confirm that the LP will be more effective if the battery starts with an initial state of charge equal to 70%. As future works, a comparative study between two exact optimization methods (i.e., LP and Mixed-Integer linear programming (MILP)) will be studied. Furthermore, the LP-based energy management will be combined with a sizing algorithm to optimize the configuration of a grid-connected hybrid system. Moreover, a

 ISBN: 978-1-61208-967-6

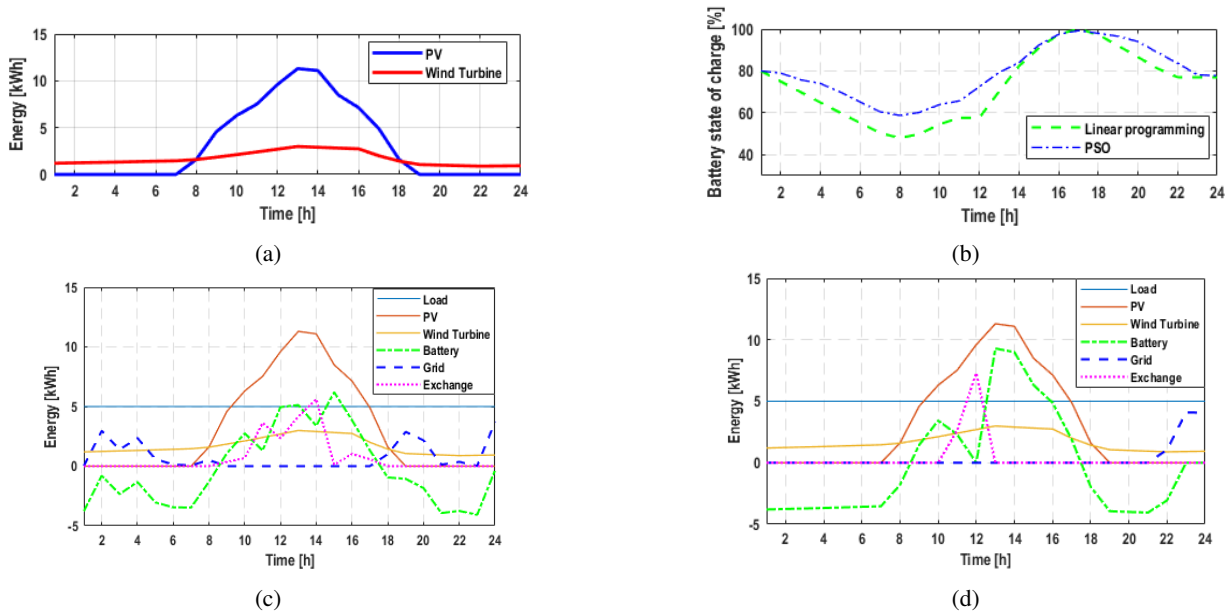


Figure 4. (a) The PV and Wind turbine energy profile during on 1st of October in Lannion. (b) The variation of the battery state of charge using PSO and LP. The energy scheduling on 1st of October using: (c) PSO and (d) LP.

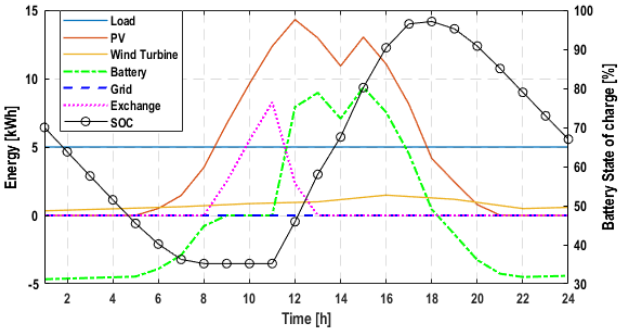


Figure 5. The energy scheduling of the system studied on 1st of July in Lannion

comparative study with the existing sizing algorithms will be performed.

REFERENCES

[1] D. Emad, M. El-Hameed, M. Yousef, and A. El-Fergany, "Computational methods for optimal planning of hybrid renewable microgrids: a comprehensive review and challenges," *Archives of Computational Methods in Engineering*, vol. 27, no. 4, pp. 1297–1319, 2020.

[2] R. Palma-Behnke, C. Benavides, F. Lanas, B. Severino, L. Reyes, J. Llanos, and D. Sáez, "A microgrid energy management system based on the rolling horizon strategy," *IEEE Transactions on smart grid*, vol. 4, no. 2, pp. 996–1006, 2013.

[3] D. Morin, Y. Stevenin, C. Grolleau, and P. Brault, "Evaluation of performance improvement by model predictive control in a renewable energy system with hydrogen storage," *International Journal of Hydrogen Energy*, vol. 43, no. 45, pp. 21017–21029, 2018.

[4] C. Bordin, H. O. Anuta, A. Crossland, I. L. Gutierrez, C. J. Dent, and D. Vigo, "A linear programming approach for battery degradation analysis and optimization in offgrid power systems with solar energy integration," *Renewable Energy*, vol. 101, pp. 417–430, 2017.

[5] I. El Kafazi and R. Bannari, "Multiobjective scheduling-based energy management system considering renewable energy and energy storage systems: A case study and experimental result," *Journal of Control, Automation and Electrical Systems*, vol. 30, no. 6, pp. 1030–1040, 2019.

[6] L. N. An, T. T. M. Dung, and T. Quoc-Tuan, "Optimal energy management for an on-grid microgrid by using branch and bound method," in *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, pp. 1–5, IEEE, 2018.

[7] L. N. An and T. Quoc-Tuan, "Optimal energy management for grid connected microgrid by using dynamic programming method," in *2015 IEEE Power & Energy Society General Meeting*, pp. 1–5, IEEE, 2015.

[8] M.-H. Laraki, B. Brahmi, C. Z. El-Bayeh, and M. H. Rahman, "Energy management system for a stand-alone wind/diesel/bess/fuel-cell using dynamic programming," in *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*, pp. 1258–1263, IEEE, 2021.

[9] S. Squartini, M. Boaro, F. De Angelis, D. Fuselli, and F. Piazza, "Optimization algorithms for home energy resource scheduling in presence of data uncertainty," in *2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP)*, pp. 323–328, IEEE, 2013.

[10] M. A. Hossain, H. R. Pota, S. Squartini, F. Zaman, and J. M. Guerrero, "Energy scheduling of community microgrid with battery cost using particle swarm optimisation," *Applied Energy*, vol. 254, p. 113723, 2019.

[11] D. Arcos-Aviles, J. Pascual, L. Marroyo, P. Sanchis, and F. Guinjoan, "Fuzzy logic-based energy management system design for residential grid-connected microgrids," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 530–543, 2016.

[12] H. Lan, S. Wen, Y.-Y. Hong, C. Y. David, and L. Zhang, "Optimal sizing of hybrid pv/diesel/battery in ship power system," *Applied energy*, vol. 158, pp. 26–34, 2015.

[13] S. Diaf, D. Diaf, M. Belhamel, M. Haddadi, and A. Louche, "A methodology for optimal sizing of autonomous hybrid pv/wind system," *Energy policy*, vol. 35, no. 11, pp. 5708–5718, 2007.

[14] H. Borhanazad, S. Mekhilef, V. G. Ganapathy, M. Modiri-Delshad, and A. Mirtaheri, "Optimization of micro-grid system using mopso," *Renewable Energy*, vol. 71, pp. 295–306, 2014.

[15] A. Moses, A. Landeros, and M. F. Abdel-Fattah, "Particle swarm optimization for sizing hybrid power systems incorporating demand response," in *2018 IEEE 59th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)*, pp. 1–7, IEEE, 2018.

[16] N. Lazaar, E. Fakhri, M. Barakat, H. Gualous, and J. Sabor, "Optimal sizing of marine current energy based hybrid microgrid," in *Conference on renewable energies and power quality*, vol. 8, 2020.

 ISBN: 978-1-61208-967-6

Photovoltaic Generation Forecasting – A Case Study

Sinan Wannous, Isabel Praça, Rui Andrade, and Sergio Ramos
Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development (GECAD)
School of Engineering, Polytechnic of Porto (ISEP/IPP)
Porto, Portugal
e-mail: {sinai, icp, faa, scr}@isep.ipp.pt

***Abstract*— The increasing demand for renewable energy sources has empowered their integration into existing power networks. This initiated an interest in investigating the capabilities of these clean sources and how can then be efficiently utilized to support the balance of energy markets. In this regard, forecasting energy generation has become an essential research problem to improve the reliability of energy systems. It is of key importance to meet the energy demand, as well as to bridge the gap between energy consumption and production in energy markets. In this research, we present a case study to investigate the performance of ensemble learning models for forecasting the energy generation of photovoltaic (PV) modules. For this purpose, we utilize a dynamic energy forecasting tool to perform various experiments with different combinations of input data fields. Primarily, the performance of 3 ensemble learning models (Adaboost, Random Forest, and Gradient Boosting Regressor) has been investigated and then compared to the predictions of two previously undertaken neural network-based methods. The results indicated higher accuracy of the ensemble approaches in almost all experiments. Which was also better than the accuracy of the neural networks-based methods.**

Keywords-Energy Prediction; Energy Forecasting Tools; Prediction Models; Machine Learning.

I. INTRODUCTION

The rapid shift from traditional fossil fuel-based energy towards renewable energy sources is one of the core strategies in developing sustainable future energy systems [1]. As a constant source of energy, sunlight is used to meet the ever-increasing energy needs, and solar energy becomes a suitable substitute for fossil fuels [2]. The forecasting of wind and solar energy is getting much attention over the last two decades. Primarily, due to the increasing amount of energy generated from these renewable sources. That said, special emphasis is given to predicting wind and solar energy records because of their variability and limited predictability, as well as instantaneous response to weather phenomena [3].

On the other hand, the prediction of solar power using photovoltaics is crucial to mitigate the random fluctuations in the incoming values. Many approaches have been utilized to predict the generated energy from Photovoltaic Panels (PV). Most of them make use of traditional statistical methods and Machine Learning (ML) approaches. Furthermore, historical data sets used to make predictions usually combine a variety of weather characteristics, cloud motion tracking, solar radiation, and many others. Nevertheless, time-horizon and climate have the most noticeable impact on the performance of solar energy forecasting [4].

In this article, we contribute to the current efforts by assessing the performance of 3 ensemble learning methods in predicting energy generation from PV panels. We also compare the results to the output of two other Artificial Neural Networks (ANN) and Deep Neural Networks (DNN) based models. The goal of this case study is to explore better forecasting circumstances by manipulating various prediction models and different input fields. The rest of this research is organized as follows. Section II summarizes a state of the art of currently used models to forecast solar energy production. Section III describes the conducted case study, including the used energy forecasting tool, data sets, and methods. In this section, we also present, discuss, and compare the results. Finally, Section IV highlights the conclusion and future perspectives of this research.

II. STATE OF THE ART

Energy forecasting is crucial in energy markets, it basically aims to build accurate forecasting models to inspect future generation/consumption scenarios. Forecasting of energy production has been widely covered in the literature to balance the supply and demand in energy systems. Attempts from workers in various fields have been made to obtain as accurate prediction models as possible. The accuracy of the forecasting models has significantly increased in the last decade. Various methods have been utilized to undergo short-term prediction experiments for energy generation obtained from photovoltaic panels. Namely, statistical methods and machine learning based methods.

A state-of-the-art of the accuracy of solar energy forecasting is conducted by Blaga, R., et al. [4]. The compared forecasting models cover various classes: persistence, classical statistics, machine learning, cloud-motion tracking, numerical weather prediction, and hybrid models. As a result, machine learning and hybrid models have the best performance for intra-hour predictions in all climates. However, according to Tato, J.H. and Brito, M.C. [5], using meteorological and historical data is not enough to produce accurate solar energy forecasts. Instead, the authors integrate a Smart Persistence prediction algorithm with Random Forests to analyze the data of six solar PV modules. The results showed a great improvement in the accuracy of short-term forecasts.

 ISBN: 978-1-61208-967-6

Furthermore, due to the dependency of PV panels on solar radiation, Global Horizontal Solar Irradiance (GHI) has a strong influence on PV production. An ANN based model was proposed to predict the next-day produced power from PV panels [6]. The model makes use of real-time solar irradiance to provide a set of decision rules for a proper prediction system. The research shows that machine learning algorithms hold some promise in this regard. Another research based on a non-linear autoregressive neural network was presented in [7]. It aims at forecasting global horizontal solar irradiance as input to a photovoltaic simulator presented in another study [8]. This system estimates the energy generation profiles of PV systems in real-sky conditions. The goal of this process is to predict energy production in short-term time periods. In a similar manner, authors of Cannizzaro, D., et al. [9] present a methodology to forecast GHI from the next 15 min and up to the next 24 hours. The proposed approach implements ML techniques including Variational Mode Decomposition (VMD), Convolutional Neural Networks (CNN), Random Forest (RF), and Long Short-Term Memory (LSTM).

On the other hand, authors of Gellert, A., et al. [10] propose a technique to predict the electricity production and consumption in a household with photovoltaics and storage systems. They analyze statistical models based on previous values aiming at increasing the self-consumption and reducing the dependency on the power grid. However, the study lacks considering environmental-specific input parameters, such as weather characteristics and contextual details. In a later study [11], the authors evaluate two statistical prediction methods: ARIMA and TBATS, and compare them to other models: Markov model, Multi-Layer Perceptron (MLP), Gated Recurrent Unit (GRU), Bayesian Regression Structural Time Series (BRSTS), and LSTM. The evaluation results showed a better mean absolute error for TBATS over what was obtained by the other models.

Deep learning methods have been also approached to tackle energy forecasting in solar systems. Three deep learning-based forecasting models were introduced for the continuous prediction of energy generated by concentrated solar power plants in Spain [12]. The proposed models are Naïve cloud-cover, ANN, and LSTM based approaches. The authors used as inputs the irradiance values and weather conditions forecasts. Another deep learning approach established on LSTM was introduced in [13]. It aimed at forecasting one hour-ahead energy production from a solar-PV plant. In this study, two other data-driven methods were also applied, and the results revealed that the LSTM model gave the best results.

Moreover, a method for detailed PV energy yield forecasting is presented in [14]. This study utilizes a local sky-imager and neural networks for horizons up to 15 min. The proposed approach eliminates the usual models, from irradiation forecast to energy yield estimation, and reduces the propagated errors. Another approach was presented to predict local PV power output based on short-term solar forecasting using ground-based cameras [15]. The research also analyzes the benefits of the forecasts to the power system. Furthermore, daily energy production forecasting methods for photovoltaic solar panels were presented using mathematical methods and fuzzy logic models [16]. The studies showed that the best model is a two-input Takagi-Sugeno system with nonlinear membership functions. In their study, authors also present a prototype software implementing the best-performing models.

Our approach brings novelty in many aspects: first, instead of considering radiation values and cloud tracking, we employ historical weather information, time contextual fields, and previous energy values as input to train our ML models. In addition, to obtain the most accurate results, we focus our efforts on conducting various experiments considering different combinations of the available input fields. Finally, we successfully utilize the tuned model to predict the energy production for a whole week instead of a couple of upcoming hours.

III. CASE STUDY

A. Overview

The aim of this case study is to investigate the performance of three ensemble learning methods in forecasting energy generation. In this context, we use a dynamic forecasting tool to perform and compare various experiments in different conditions. We also use a historical data set that combines the instant generation of 3 PV panels. Primarily, results of the considered models are presented, discussed, and then compared to previously undertaken predictions using two neural network-based models. The forecasting of PV generation data will be used and integrated by an Energy Resource Management System (ERMS), in a collective residential building, to support the management of all building resources aiming to minimize the electricity consumption costs.

B. Energy Forecasting Tool

In this case study, we used a dynamic energy forecasting tool that was developed by the GECAD research group [17]. The tool is a web-based application that extends a set of machine learning models to provide dynamic energy forecasting services. It provides interactive user interfaces to predict energy generation/consumption, build forecasting models, compare predictions, and fine-tune prediction models. The used tool utilizes five supervised machine learning estimators which include: Adaboost.R2 (Ada.) [18], Random Forest Regressor (RF) [19], Gradient Boosting Regressor (GBR) [20], Support Vector Regression (SVR) [21], and Linear Regression (LR) [22]. Furthermore, the tool maintains two common validation mechanisms: Train Test Split and Cross-validation. Services provided by this tool cover a set of training, predicting, and tuning features with various input and output capabilities. In Figure 1, we show a sample of the tool's training interface, in which the user controls the configurations of 3 prediction models. This tool has been used previously in conjunction with other aspects that might benefit from the energy forecasting services. Mainly, in building trust models for Local Energy Markets [23] [24].

 ISBN: 978-1-61208-967-6

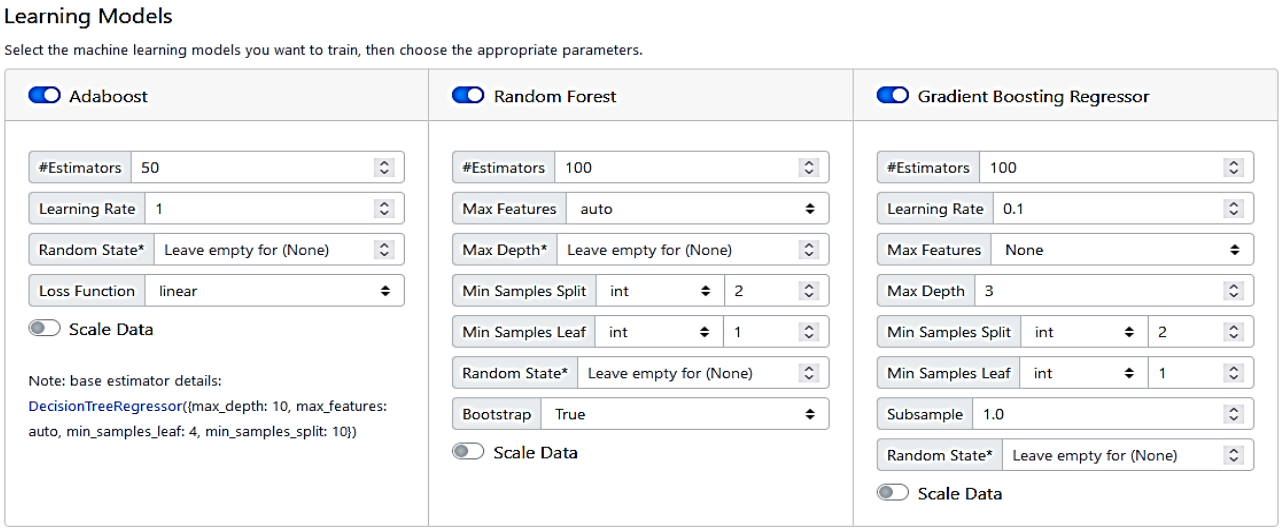


Figure 1. Energy Forecasting Tool: An example of the model training interface, it shows 3 models with their default parameters.

C. *Data Sets*

The historical data used in this case study combines the energy generation of three different photovoltaic solar modules installed and operated in Porto. The PV generation system under analysis is installed on the roof of a residential building consisting of 15 apartments of different typologies. There are 28 PV panels installed, each with a power of 400 Wp, for a total installed capacity of 11.2 kW. This total PV power is distributed into three sets of producers, each with a 3.68 kW installed PV power [25]. The data set represents the generated energy values in kW for each solar panel as well as the total generated values. It was internally collected and registered in a timestamp interval of 15 min and covers the whole year of 2019.

Furthermore, to enrich the input data fields, we managed to retrieve detailed weather values of the exact location where the panels are installed. For this sake, we used a global weather API provided by World Weather Online [26]. Collected weather values include but are not limited to temperature, wind speed, direction, precipitation, humidity, visibility, pressure, cloud cover, etc.

For the sake of transparently comparing the output predictions, the data set also includes forecasting results of two other prediction models. ANN and DNN were previously trained and used to forecast energy generation for the first week of September of the same year (2019) [27]. In this study, four forecasting performance metrics (Mean Absolute Error, Symmetric Mean Absolute Percentage Error, Weighted Absolute Percentage Error, and Normalized Root Mean Square Error) were used to evaluate the accuracy of both forecasting algorithms. The obtained forecasting results showed that both techniques had similar prediction behavior, however, and based on the obtained forecasting evaluation errors, the ANN presented a slightly better prediction performance in comparison with DNN.

D. *Methods*

We utilized the energy forecasting tool to perform multiple experiments. Primarily, we made use of 3 main services: a) model training with default parameters, to train models using a historical data set and generate downloadable trained models, b) bulk prediction, to use trained models to predict multiple future records, and c) model tuning, to find the best parameters for each model considering specific input data fields. Moreover, as per a case study undertaken using the same tool [17], ensemble learning methods had proven the most accurate results in energy forecasting in similar conditions. Consequently, we used the three ensemble learning methods: Ada., RF, and GBR. To validate trained models, we preferred to opt for the cross-validation mechanism over the train test split. Although this validation method is more expensive in terms of computational cost, it brings better and more reliable accuracy values.

As input, the tool accepts three categories of data fields: contextual fields, weather attributes, and preceding consumption/generation values. In this case study, we consider all available contextual fields, weather fields that might affect the solar reflection on panels and, thus, affect the generated energy, and up to 10 previous generation values (see Table I).

TABLE I. INPUT DATA FIELDS CONSIDERED IN THE CASE STUDY

Input Categories	Input Fields
Contextual values	Minute (min), Hour (h), Day of the week (dw), Day of the month (dm), Month (m), Year (y).
Weather attributes	Temperature (temp), Wind speed (ws), Cloud cover (cc), Visibility (vis), Precipitation (p).
Previous values	Up to 10 previous generations, where (v_{t-1}) refers to the 1st previously generated value, (v_{t-2}) refers to the 2nd previously generated value, and (v_{t-n}) refers to the nth previously generated value.

 ISBN: 978-1-61208-967-6

For this case study, we used the tool to perform 14 training experiments for each model (Ada., RF, and GBR). Each experiment examines a different combination of input fields (see Table II). To obtain the most reliable results, we developed our case study as the following: first, we trained the three models using only contextual fields (Exp^1). Then, we combined both contextual and weather fields to check the expected influence of weather conditions (Exp^2). After that, to adjust the best number of previous values to be incorporated, we combined contextual fields with 1, 2, 3, and up to 10 previous values (Exp^{3-12}). Afterward, we combined the contextual fields, weather data, and the best number of previous values for each model (Exp^{13}). Finally, we used the tuning module to fine-tune the resulted models and perform the final experiment (Exp^{14}).

That said, for each experiment, we trained the three ensemble models using the total generated energy records, from January until August 2019. Then, we validated the trained models using the cross-validation technique and registered the averaged prediction accuracy for each model/experiment. Finally, to compare results with the previously conducted ANN and DNN methods, we used the bulk prediction service to predict energy generation during the 1st week of September of the same year. Nevertheless, although all models can be generalized to cover multiple years energy records, we had to consider 8 months to compare results with ANN and DNN predictions.

TABLE II. TRAINING EXPERIMENTS, INCLUDING INPUT DATA FIELDS FOR EACH ONE

Exp.	Input Fields
Exp^1	Contextual (min, h, dw, dm, m, y)
Exp^2	Contextual (min, h, dw, dm, m, y), Weather (temp, ws, cc, vis, p)
Exp^3	Contextual (min, h, dw, dm, m, y), Previous Values (v_{t-1})
Exp^4	Contextual (min, h, dw, dm, m, y), Previous Values (v_{t-1}, v_{t-2})
Exp^5	Contextual (min, h, dw, dm, m, y), Previous Values (v_{t-1}, v_{t-2}, v_{t-3})
Exp^6	Contextual (min, h, dw, dm, m, y), Previous Values (v_{t-1}, v_{t-2}, v_{t-3}, v_{t-4})
Exp^7	Contextual (min, h, dw, dm, m, y), Previous Values (v_{t-1}, v_{t-2}, v_{t-3}, v_{t-4}, v_{t-5})
Exp^8	Contextual (min, h, dw, dm, m, y), Previous Values (v_{t-1}, v_{t-2}, v_{t-3}, v_{t-4}, v_{t-5}, v_{t-6})
Exp^9	Contextual (min, h, dw, dm, m, y), Previous Values (v_{t-1}, v_{t-2}, v_{t-3}, v_{t-4}, v_{t-5}, v_{t-6}, v_{t-7})
Exp^{10}	Contextual (min, h, dw, dm, m, y), Previous Values (v_{t-1}, v_{t-2}, v_{t-3}, v_{t-4}, v_{t-5}, v_{t-6}, v_{t-7}, v_{t-8})
Exp^{11}	Contextual (min, h, dw, dm, m, y), Previous Values (v_{t-1}, v_{t-2}, v_{t-3}, v_{t-4}, v_{t-5}, v_{t-6}, v_{t-7}, v_{t-8}, v_{t-9})
Exp^{12}	Contextual (min, h, dw, dm, m, y), Previous Values (v_{t-1}, v_{t-2}, v_{t-3}, v_{t-4}, v_{t-5}, v_{t-6}, v_{t-7}, v_{t-8}, v_{t-9}, v_{t-10})
Exp^{13}	Contextual (min, h, dw, dm, m, y), Weather (temp, ws, cc, vis, p), Previous Values (Best of Exp^{3-12} for each model)
Exp^{14}	Same as Exp^{13}

E. Results and Discussion

In Table III, we summarize the prediction accuracy R^2 (1) achieved by each model in all performed experiments.

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}, where\ \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad (1)$$

TABLE III. PREDICTION ACCURACY FOR EACH MODEL IN ALL EXPERIMENTS

Exp	Description	Ada. R^2	FR R^2	GBR R^2
Exp^1	Only contextual fields	93.6 %	95.2 %	83.7 %
Exp^2	Contextual + Weather	94.5 %	95.8 %	86.5 %
Exp^3	Contextual + 1 previous value	95.9 %	96 %	96.2 %
Exp^4	Contextual + 2 previous values	96 %	96.1 %	96.3 %
Exp^5	Contextual + 3 previous values	96 %	96.2 %	96.3 %
Exp^6	Contextual + 4 previous values	96 %	96.3 %	96.4 %
Exp^7	Contextual + 5 previous values	96.1 %	96.3 %	96.3 %
Exp^8	Contextual + 6 previous values	96.1 %	96.3 %	96.3 %
Exp^9	Contextual + 7 previous values	96.1 %	96.3 %	96.3 %
Exp^{10}	Contextual + 8 previous values	96.2 %	96.3 %	96.3 %
Exp^{11}	Contextual + 9 previous values	96.1 %	96.3 %	96.3 %
Exp^{12}	Contextual + 10 previous values	96.2 %	96.3 %	96.3 %
Exp^{13}	Contextual + Best previous values* + Weather	96.2 %	96.3 %	96.4 %
Exp^{14}	Exp^{13} TUNED	96.3 %	96.4 %	96.4 %

* For Ada.: 8 (or 10) previous values, For RF: 4 (or 5-10) previous values, For GBR: 4 previous values

Looking into the detailed results, all models could obtain high accuracy in almost all experiments. This might be explained in terms of the consistent generation of the considered solar panels. Even when using only contextual fields (Exp^1), we get high accuracy with a minimum of 83.7% for the GBR model. Such results clearly indicate the significant influence of contextual fields on energy predictions. Furthermore, we also notice the enhancement that weather fields achieved when combined with contextual data (Exp^2). Nevertheless, as weather conditions highly affect energy generation using solar panels, we still expect greater impacts of weather fields in different circumstances. For example, with a lower base contextual accuracy when dealing with less or inconsistent generation values.

On the other hand, the results also show an increasing accuracy upon considering preceding values (Exp^{3-12}), especially in the very early stage when we started to combine the latest generation values (Exp^{3-6}). These experiments also indicate that the more previous values to consider do not necessarily mean higher prediction accuracy. As we can notice a phase of fluctuation for each model after reaching a specific number of previous values. Nevertheless, with a 15-min interval data log, 10 previous values cover 2:30 hours. Consequently, we also expect a downgrade in prediction accuracies when combining a longer prior period which might involve much divergence in the actual generated energy. Finally, as expected, combining all input fields (Exp^{13}) as well as utilizing tuned models (Exp^{14}) were eventually able to bring the best accurate results.

 ISBN: 978-1-61208-967-6

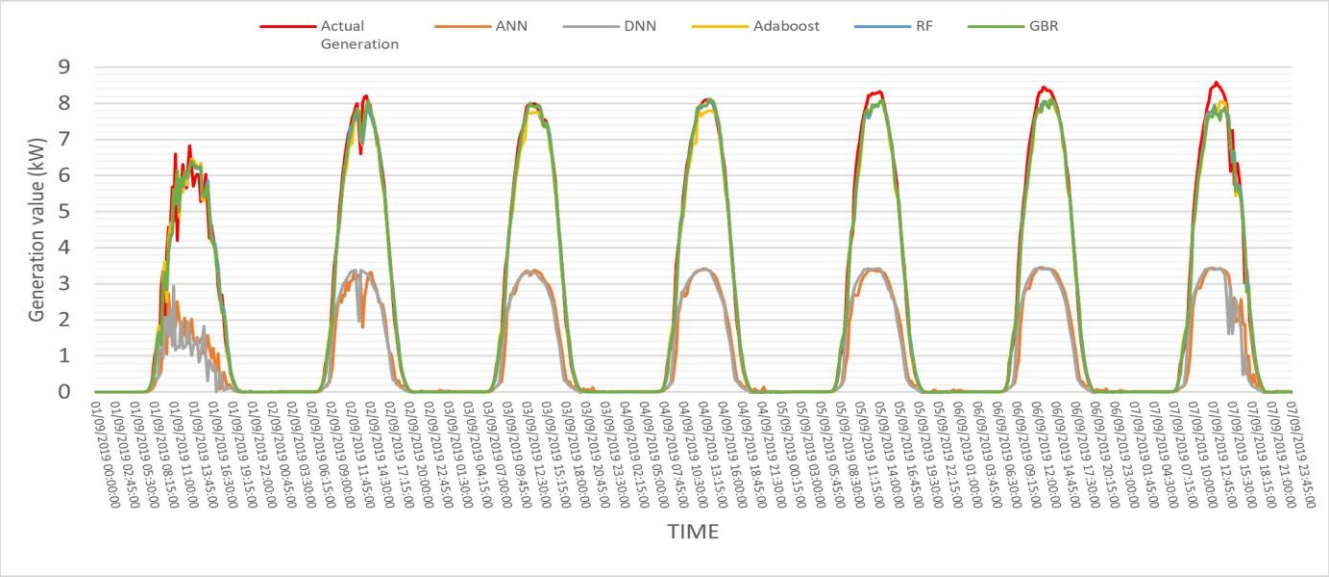


Figure 2. Actual generation and predictions for the 1st week of September, 2019.

Regarding the forecasting models, we could notice, in general, not much difference in the performance of the three considered models. However, we could see relatively better results of Adaboost and Random Forest in the first two experiments (Exp[1,2]). While the performance of all models turned too close during the later observations.

F. Comparison

As mentioned earlier, to better assess our prediction models, we used the three models resulting from Exp[14] to predict the energy generation values during the first week of September 2019. We also compared the results with previously undertaken predictions for the same period using ANN and DNN [27]. All predictions were conducted in a time interval of 15 min. Figure 2 shows the prediction results and the actual generation during the first week of September for all considered models. Furthermore, Table IV presents the accuracy (R^2 score) for the predicted generation values during the considered period.

TABLE IV. PREDICTION ACCURACY FOR EACH MODEL FOR THE 1ST WEEK OF SEPTEMBER. 2019

Prediction Model	Artificial Neural Networks	Deep Neural Networks	Adaboost	Random Forest	Gradient Boosting Regressor
R^2 Score	97.9 %	96.1 %	99.6 %	99.7 %	99.7

It is clearly noticeable that the predictions of the three ensemble learning methods are almost identical to the actual generation along the whole observed period. This is reasonable in terms of the higher accuracy of the trained models obtained during the case study. Likewise, we can observe some deviation in the predictions of the other two ANN and DNN models from the actual generation, especially during the day hours when there is actual energy generation. We could also notice that, although ANN and DNN predictions were consistently low in comparison to the actual generation, they could relatively reflect the overall trend of the generated values.

IV. CONCLUSION

This research is a contribution to the efforts to obtain accurate energy forecasting from photovoltaic panels. In this regard, we conducted a case study to predict the energy production from 3 PV modules installed and running in Porto. The used dataset combines historical records of weather data, time-contextual fields, and previous generation values. We used a dynamic forecasting tool to undergo various prediction experiments using 3 ensemble learning models (Adaboost, Random Forest, and Gradient Boosting Regressor). Obtained results are then compared with the results of two ANN and DNN based models. The results indicate relatively high accuracy of the ensemble approaches in almost all experiments. Which was also much better than the accuracy of the previously conducted neural networks-based methods.

This case study shows interesting new results. However, obtaining high accuracy in forecasting energy generation in specific conditions doesn't eliminate the investigation process. Each forecasting problem has its own circumstances that are not necessarily the same in another environment. Upcoming challenges in the production of renewable energy always require better forecasting models. Future work might imply performing further experiments at multiple scales, utilizing a wider range of combinations between input fields, as well as investigating the effects of solar radiation when combined with other fields already considered in this research. The ultimate goal for such experiments would be to obtain as accurate results as possible within specific prediction conditions.

 ISBN: 978-1-61208-967-6

ACKNOWLEDGMENT

This work has received funding from FEDER Funds through COMPETE program and from National Funds through FCT under the project SPET–PTDC/EEI-EEE/029165/2017. This work has also received funding from projects BENEFICE Project - PTDC/EEI-EEE/29070/2017, UIDB/00760/2020, and UIDP/00760/2020.

REFERENCES

[1] A. J. Chapman, B. C. McLellan, and T. Tezuka, "Prioritizing mitigation efforts considering co-benefits, equity and energy justice: Fossil fuel to renewable energy transition pathways," *Appl. Energy*, vol. 219, pp. 187–198, 2018.

[2] A. Qazi *et al.*, "Towards sustainable energy: a systematic review of renewable energy sources, technologies, and public opinions," *IEEE access*, vol. 7, pp. 63837–63851, 2019.

[3] C. Sweeney, R. J. Bessa, J. Browell, and P. Pinson, "The future of forecasting for renewable energy," *Wiley Interdiscip. Rev. Energy Environ.*, vol. 9, no. 2, p. e365, 2020.

[4] R. Blaga *et al.*, "A current perspective on the accuracy of incoming solar energy forecasting," *Prog. energy Combust. Sci.*, vol. 70, pp. 119–144, 2019.

[5] J. Huertas Tato and M. Centeno Brito, "Using smart persistence and random forests to predict photovoltaic energy production," *Energies*, vol. 12, no. 1, p. 100, 2019.

[6] M. H. Alomari, J. Adeeb, and O. Younis, "Solar photovoltaic power forecasting in jordan using artificial neural networks," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 1, p. 497, 2018.

[7] A. Aliberti *et al.*, "Forecasting Short-term Solar Radiation for Photovoltaic Energy Predictions.," in *SMARTGREENS*, 2018, pp. 44–53.

[8] L. Bottaccioli, E. Patti, E. Macii, and A. Acquaviva, "GIS-based software infrastructure to model PV generation in fine-grained spatio-temporal domain," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2832–2841, 2017.

[9] D. Cannizzaro *et al.*, "Solar radiation forecasting based on convolutional neural network and ensemble learning," *Expert Syst. Appl.*, vol. 181, p. 115167, 2021.

[10] A. Gellert, A. Florea, U. Fiore, F. Palmieri, and P. Zanetti, "A study on forecasting electricity production and consumption in smart cities and factories," *Int. J. Inf. Manage.*, vol. 49, pp. 546–556, 2019.

[11] A. Gellert, U. Fiore, A. Florea, R. Chis, and F. Palmieri, "Forecasting Electricity Consumption and Production in Smart Homes through Statistical Methods," *Sustain. Cities Soc.*, vol. 76, p. 103426, 2022.

[12] J. Segarra-Tamarit, E. Pérez, E. Moya, P. Ayuso, and H. Beltran, "Deep learning-based forecasting of aggregated CSP production," *Math. Comput. Simul.*, vol. 184, pp. 306–318, 2021.

[13] A. Ozbek, A. Yildirim, and M. Bilgili, "Deep learning approach for one-hour ahead forecasting of energy production in a solar-PV plant," *Energy Sources, Part A Recover. Util. Environ. Eff.*, pp. 1–16, 2021.

[14] D. Anagnostos *et al.*, "A method for detailed, short-term energy yield forecasting of photovoltaic installations," *Renew. Energy*, vol. 130, pp. 122–129, 2019.

[15] A. Jakoplić, D. Franković, V. Kirinčić, and T. Plavšić, "Benefits of short-term photovoltaic power production forecasting to the power system," *Optim. Eng.*, vol. 22, no. 1, pp. 9–27, 2021.

[16] G. Dec, G. Drałus, D. Mazur, and B. Kwiatkowski, "Forecasting Models of Daily Energy Generation by PV Panels Using Fuzzy Logic," *Energies*, vol. 14, no. 6, p. 1676, 2021.

[17] S. Wannous, I. Praça, and R. Andrade, "Intelligence as a Service: A Tool for Energy Forecasting and Security Awareness," in *Practical Applications of Agents and Multi-Agent Systems*, 2021, pp. 176–186.

[18] H. Drucker, "Improving regressors using boosting techniques," in *ICML*, 1997, vol. 97, pp. 107–115.

[19] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[20] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Ann. Stat.*, pp. 1189–1232, 2001.

[21] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

[22] G. D. Hutcheson, "Ordinary least-squares regression," *L. Moutinho GD Hutcheson, SAGE Dict. Quant. Manag. Res.*, pp. 224–228, 2011.

[23] R. Andrade, I. Praça, S. Wannous, and S. Ramos, "The Impact of Attacks in LEM and Prevention Measures Based on Forecasting and Trust Models," *Processes*, vol. 9, no. 2, p. 314, 2021.

[24] R. Andrade, S. Wannous, T. Pinto, and I. Praça, "Extending a Trust model for Energy Trading with Cyber-Attack Detection," *Electronics*, vol. 10, no. 16, p. 1975, 2021.

[25] Z. Foroozandeh, S. Ramos, J. Soares, Z. Vale, and M. Dias, "Single contract power optimization: A novel business model for smart buildings using intelligent energy management," *Int. J. Electr. Power \& Energy Syst.*, vol. 135, p. 107534, 2022.

[26] "World Weather Online." https://www.worldweatheronline.com/ [retrieved: March, 2022].

[27] I. Tavares *et al.*, "Comparison of PV Power Generation Forecasting in a Residential Building using ANN and DNN." CPES 2022, 11th Symposium on Control of Power and Energy Systems, June 21-23, 2022, in press.

 ISBN: 978-1-61208-967-6

Energy Efficiency of Parallel File Systems on an ARM Cluster

Timm Leon Erxleben*, Kira Duwe*, Jens Saak†, Martin Köhler† and Michael Kuhn*
*Otto von Guericke University Magdeburg
Magdeburg, Germany
E-mail: timm.erxleben@ovgu.de, kira.duwe@ovgu.de, michael.kuhn@ovgu.de
†Max Planck Institute for Dynamics of Complex Technical Systems
Magdeburg, Germany
E-mail: saak@mpi-magdeburg.mpg.de, koehlerm@mpi-magdeburg.mpg.de

***Abstract*—Parallel distributed file systems are typically run on dedicated storage servers that clients connect to via the network. Regular x86 servers provide high computational power, often not required for storage management and handling I/O requests. Therefore, storage servers often use low core counts but still have a relatively high idle power consumption. This leads to high energy consumption, even for mostly idle file systems. Advanced Reduced Instruction Set Computer Machines (ARM) systems are very energy-efficient but still provide adequate performance for file system use cases. Leveraging this fact, we built an ARM-based storage system, on which we tested both CephFS and OrangeFS. We compare the performance and energy efficiency of x86 and ARM systems using several metrics. Results show that while our ARM-based approach currently provides less throughput per Watt for reads, it achieves an approximately 121 % higher write efficiency when compared to a traditional x86 Ceph cluster.**

***Keywords*—energy efficiency, CephFS, OrangeFS, x86, ARM**

I. Introduction

Storage systems are scaled up steadily to satisfy increasing storage demands, leading to growing energy consumption [1]. High-Performance Computing (HPC) storage systems are currently built from regular x86 servers, whose computing power is not fully utilized by storage applications. Traditional x86 servers feature a relatively high power consumption even when idle: It is not uncommon to measure idle consumption of more than 100 W for just the processor, main memory, and mainboard. In comparison, low-power ARM computers are often required to stay below 5–10 W maximum consumption by design. To offset the high idle consumption of x86 servers, they have to be equipped with large amounts of storage devices, such as hard disk drives (HDDs) and solid-state disk (SSDs). However, depending on the used network interconnect, only a limited number of devices can be saturated. For instance, on a 100 Gbit/s network, two to three NVMe SSDs are enough to provide the necessary throughput. This proportion gets even worse on slower networks.

Therefore, we evaluate the use of low-energy ARM-based single-board computers as a replacement for traditional servers in storage systems. To assess the feasibility of an ARM-based storage system, we evaluated the ARM cluster using CephFS and OrangeFS. Furthermore, we compared it to a productive CephFS cluster running at the computer science faculty of the Otto von Guericke University, using different metrics.

The contributions of our paper are:

1) We propose to apply the energy-delay product, typically used to evaluate the energy efficiency of computations, as a metric for storage systems as well to measure energy efficiency while still accounting for the performance needed by HPC applications.
2) We show that low-power ARM-based storage clusters can achieve throughput efficiencies comparable to or even exceeding traditional x86 systems.

The remainder of the paper is organized as follows. In section II, CephFS and OrangeFS are briefly described followed by a summary of related works. Section III describes the benchmarks which were done and discusses metrics that can be derived from the measurement data. Next, in section IV both cluster setups, ARM and x86, are described, followed by the presentation of the results. Results and setups are discussed in section V. Finally, section VI concludes the paper.

II. Background and Related Work

This section introduces background on used technologies, such as Ceph and OrangeFS, and related work.

a) Ceph: Ceph is a popular, clustered object store, which is highly scalable due to its Controlled Replication Under Scalable Hashing (CRUSH) placement algorithm, which enables all participating services, that can access the cluster map to locate and place objects [2]. A typical Ceph cluster is made of Object Storage Devices (OSDs), monitoring and management services. All components may be redundant to enable automatic failover. Apart from access through the library `librados`, many interfaces might be used. The POSIX access via CephFS, realized by additional Metadata Services (MDSs) interacting with Ceph storage pools, is particularly interesting for HPC systems. CephFS has a rich feature set, including replication, multiple storage pools, file systems, snapshots, and high control over data placement [3].

b) OrangeFS: OrangeFS is a traditional parallel file system designed for HPC [4]. Only one type of server is needed, which can handle both data and metadata, though it can be configured to handle only one type. In OrangeFS, data is striped according to a distribution function that can be specified for each file. The default is to start at a random server and use all servers in a round-robin fashion with a stripe size of 64 KiB. Unlike Ceph, which uses its own object store *Bluestore* [5], OrangeFS relies on a separate local file system. As of the current version, 2.9.8, there are no redundancy

 ISBN: 978-1-61208-967-6

features for data that is not marked as read-only, though this is planned for OrangeFS version 3 [6]. Many interfaces may be used to interact with OrangeFS. Most popular choices include access via the OrangeFS Linux kernel module or direct access using the library `libpvfs2`. Noteworthy is the direct Message Passing Interface I/O (MPI-IO) support by using ROMIO's [7] Abstract-Device Interface for I/O (ADIO), for which OrangeFS provides an implementation [8].

c) State of the Art and Related Work: There have been various endeavors to measure and increase the energy efficiency of large systems as energy consumption is becoming a possible constraint on HPC systems in the future. Many different aspects have to be considered, ranging from the system's energy efficiency to the scalability of the applications. As ARM processors aim to offer better energy efficiency, they have been heavily studied across the years [9–11]. Deployments, such as Fugaku [12], show that they can provide competitive performance and even work in exascale systems. Earlier research on systems like Tibidabo at Barcelona Supercomputing Center indicated that single instruction multiple, data stream (SIMD) instructions limited to single precision were a severe bottleneck for the performance [10][11][13].

Energy efficiency is also a relevant aspect in distributed systems, as examined for Peer-to-Peer systems. A survey by Brienza et al. showed that often simple energy models were used, disregarding other hardware components like intermediate routers [14]. An early approach, and still very prominent solution to energy savings in storage, is sending idle peers to sleep [15]. However, it introduces problems when the load varies. To have systems benefit from the increased energy efficiency, in the long run, applications have to be considered as well. The optimization towards energy efficiency comes indeed with its challenges for applications [13][16–18]. Reducing the performance of a single core in order to cap the power consumption means that scalability is of increased importance [13].

Gudu and Hardt evaluated the use of an ARM-based Ceph cluster, made of Cubieboards, as a replacement for traditional network-attached storage (NAS) controllers [19]. They measured the throughput of their cluster via Ceph's Reliable Autonomic Distributed Object store (RADOS) and RADOS Block Device (RBD) access and found that the Cubieboard cluster is a viable alternative to NAS controllers. However, the limited network capabilities were the bottleneck of the system.

Apart from using low-power hardware [20], there have been efforts to reduce the power consumption of existing HPC storage clusters [21][22]. For example, it was proposed to assign subsets of storage clusters to specific users and only run them at full power when said user uses the compute-cluster [23].

Considering that local file systems are often part of the storage stack, their influence on energy efficiency and performance were analyzed in [24] using simulated workloads of web, database, and file servers. It was found that the choice of file system and its configuration greatly influence performance and energy efficiency. However, no file system performed best for all workloads.

In contrast to Gudu and Hardt, we measure data throughput at the CephFS level and evaluate ARM-based clusters as a replacement for HPC storage clusters.

III. Benchmark and Metrics

We measured the performance of the clusters for sequential, independent accesses from one to four clients using `IOR` v3.3 [25] with the POSIX backend, individual files per client and five iterations for each data point. The transfer size was set to 4 MiB, which corresponds to the default stripe size of CephFS and is aligned to the stripe size of 64 KiB on OrangeFS. On the x86 Ceph cluster, 96 GiB were written and read. The amount of data was reduced to 36 GiB for the ARM setup to keep run-times manageable.

For every iteration, the power consumption of the storage cluster was measured using the methods as described in Section IV. As a result, several energy efficiency metrics can be derived from the collected data. However, choosing a specific metric is not trivial, as there is no single optimal metric indicating energy efficiency [26].

We decided to compare the results obtained by using the **energy-delay product** (EDP) [27], **throughput per Watt** and **capacity per Watt** [28].

Throughput per Watt is a commonly used metric for evaluating and comparing storage energy efficiency. The transferred data may differ between systems, so it is well suited to compare systems that greatly vary in their performance. However, this metric alone is insufficient when analyzing and optimizing storage systems, as no insight into performance is given. Geveler et al. [16] found that for simulations, in some cases, energy savings might lead to performance drops. In such cases, they motivated using the EDP as a fused metric describing energy efficiency and performance at once. The EDP is computed as the product of the total energy E consumed while performing a task and the time t needed to complete the task (Equation (1)). Depending on the performance requirements, the time may be weighted [29]. As we want to focus on energy consumption, we set $w = 1$.

$$\mathrm{EDP} = E \cdot t^w, \quad w \in \mathbb{N} \tag{1}$$

Though the energy-delay product was initially developed for hardware design, it is also useful when evaluating software, as done by Georgiou et al. [30]. Nevertheless, the amount of work needs to stay constant to compare different systems, so only the two ARM setups are compared using the EDP. Because its unit is hard to interpret and even changes with different weights, we normalized the EDP using the lowest value per comparison.

The third metric considered measures the capacity of the storage system per Watt. Because of growing storage demands and, therefore, growing storage systems, optimizing systems regarding this metric is critical for the cost-efficient and environmentally friendly operation of data centers.

 ISBN: 978-1-61208-967-6

IV. Evaluation

In this section, the hardware and software setup is described, followed by an analysis of the respective clusters' theoretical peak performance and the presentation of the results.

a) Reference Cluster: The reference cluster is a four-node subset of the productive Ceph cluster running at the computer science faculty at the Otto von Guericke University using Ceph 16.2.7 deployed as containers. Three nodes of the subset are part of the Supermicro AS 2124BT-HNTR [31] multi-node system, each of which is equipped with four Intel P4510 NVMe SSDs [32]. The fourth server is a Gigabyte R282-Z94 [33] equipped with one Intel P4510 NVMe SSD and eight Samsung MZQL23T8HCJS-00A07 NVMe SSDs [34]. All nodes are connected by 100 Gbit Ethernet, with a separate 100 Gbit network for communication between Ceph OSDs. Though Ceph does not exclusively use the nodes, they are idle most of the time. The average idle power consumption of the four nodes was measured to be **699.3 W**. This power measurement was done on a Sunday since the servers are mostly idle on the weekend. It lasted for one hour, starting at 14:00, and had a standard deviation of 13.98 W. While running, the benchmark power consumption peaked at 1,057 W. The existing monitoring solution, gathering power samples over IPMI every 15 seconds, was used to collect power samples.

For each SSD, two Ceph OSDs are deployed. The Ceph monitor and a standby metadata service are located at the Gigabyte server, while the active metadata service runs on one of the Supermicro servers. Ceph pools use the default replication settings and, therefore, produce three replicas of the data and return to the client after two replicas are written. The clients used for the benchmark were four servers equipped with an AMD Epyc 7443, with 24 cores at 2.85 GHz, 128 GB RAM, and 100 Gbit Ethernet.

b) ARM cluster nodes: The low-power cluster is built of six Odroid HC4 nodes featuring the Amlogic S905X3 SoC, with four cores at 1.8 GHz, 4 GiB DDR4 RAM, two SATA-3 ports, and a 1 Gbit NIC [35]. We used Armbian Buster [36], and Ceph version 14.2.21, which is available in the Buster backports repository. We built OrangeFS version 2.9.8 with GCC version 8.3.0 and LMDB 0.9.22 from the Buster repository. Four of the nodes are equipped with two 1 TB WD Black HDDs [37] and one is equipped with two 512 GB Samsung V-NAND SSD 860 PRO SSDs [38]. All nodes are connected to a Netgear GS110EMX switch [39].

One OSD is deployed for each storage device. The node which is equipped with SSDs additionally runs one MDS. The Ceph monitor and management daemon run on the sixth node, which has no disks attached. The two storage pools needed for CephFS use different CRUSH rules to distribute objects. While the data pool uses all HDDs and manages replicas on the node level, the metadata pool uses the two SSDs and manages replicas on the OSD level. Both pools are configured to use 64 placement groups. Ceph is configured to produce two replicas and return immediately after one replica is written, allowing a fairer comparison with OrangeFS.

As explained above, OrangeFS has only a single type of daemon, which is running on all nodes with disks. Metadata is stored by the daemon, which is deployed on the SSD node, while the other nodes store the data. As OrangeFS offers no data redundancy for data that is not read-only, ZFS version 2.0.3 was used to mirror disks locally.

The complete cluster, including the switch, is powered by an MW HRP450-15 PSU [40] and consumes 56.36 W, measured over one hour with a standard deviation of 0.14 W, in idle state, with HDDs spun up. For comparison with the reference cluster, which does not include the switch in the power measurements, we subtracted the average idle power of the switch, which was measured to be 15.46 W, with a standard deviation of 1.13 W over one hour. The adjusted idle power consumption of the ARM cluster, therefore, is **40.9 W**. The highest peak in power consumption measured while running the benchmark was 58.9 W.

For power measurements, the ZES Zimmer LMG 450 [41] is used to measure the power consumption of the PSU for the whole cluster. The power meter is connected to one of the clients via USB, which collects samples with 20 Hz. The clients used to perform the benchmark were four Dell Precision 3650 Tower workstations [42] each with an Intel Core i7-11700 CPU with 8 cores at 2.5 GHz, 8 GB RAM, and a 1 Gbit NIC. They were connected via the network infrastructure of the Max Planck Institute Magdeburg.

c) Theoretical Peak Performance: As can be seen in Table I the theoretical peak performance (TPP) of the ARM cluster is limited by the network throughput of each node which is not as high as the aggregated throughput of all storage devices of the node. As no measurements could be made in the productive reference cluster, the maximum throughput of the components is taken from the respective datasheets. Adding together the TPP of the two-node types, the reference cluster's TPP is **47.3 GB/s**.

This analysis neglects metadata operations which are reasonably assumed not to limit the data throughput of the cluster for a few files in use. Furthermore, the table only presents the performance for writes. However, as the network already limits peak performance for the ARM cluster and aggregated throughput of the SSDs in Supermicro nodes of the reference cluster is close to the network speed, the same applies approximately to reads.

d) Results: The results of the performance efficiency metrics are shown in Figure 1. Each value of the throughput per Watt metric is computed as the mean of five samples, each divided by the mean power consumption of their iteration. Error bars on the plots depict the standard deviation. As explained above, the EDP (see Figure 2) is normalized by the lowest value per comparison. The capacity metric was computed using the idle power consumption of the clusters and the raw storage capacity. The usable storage capacity depends on the respective setup. The ARM cluster achieved **0.196 TB/W** and the reference cluster **0.073 TB/W**, see Figure 3.

 ISBN: 978-1-61208-967-6

TABLE I. THROUGHPUT OF COMPONENTS RELEVANT FOR THEORETICAL PEAK PERFORMANCE (TPP) THROUGHPUT

Cluster	Network	Throughput Storage Devices	Storage Devices per Node	# Nodes	TPP
ARM	124.1 MB/s	115 MB/s	2	4	496.4 MB/s
Supermicro	12.5 GB/s	2.9 GB/s	4	3	34.8 GB/s
Gigabyte	12.5 GB/s	2.9 GB/s / 4 GB/s	1+8	1	12.5 GB/s

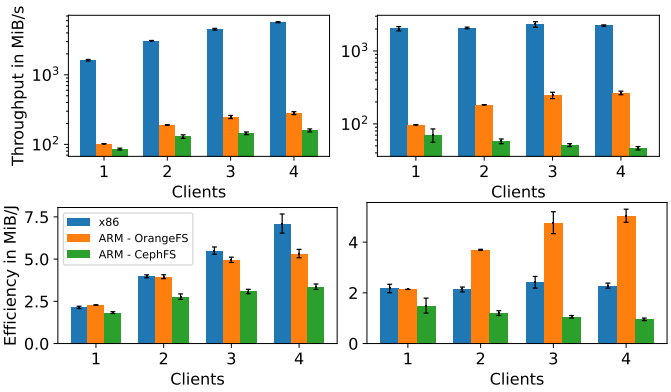


Figure 1. Throughput (top), Throughput per Watt (bottom) for reading (left) and writing (right)

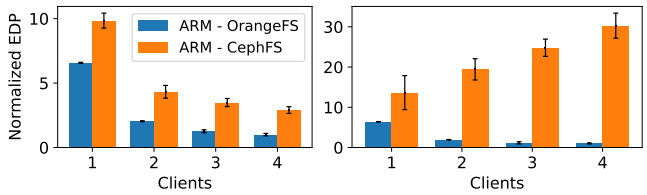


Figure 2. Normalized energy-delay product for reading (left) and writing (right)

V. DISCUSSION

All results need to be seen in relation to the respective systems' cost, as the ARM cluster nodes and disks cost only about €1,350, while the reference cluster nodes and disks cost around €40,000. In addition, the reference cluster only uses NVMe SSDs, while the ARM cluster uses HDDs for data object storage. Due to the low sampling rate of the power measurements for the reference cluster, some spikes in the energy consumption are possibly missed, resulting in an underestimation. In contrast, power measurements on the ARM cluster can be expected to overestimate the actual power consumption of the nodes and disks, as only the average idle power consumption of the switch is subtracted.

During previous experiments on a BananaPi M1 single-board computer cluster, the deployment of traditional parallel file systems proved difficult. Tested file systems were CephFS, OrangeFS and BeeGFS. Both CephFS and BeeGFS needed small patches to run on the unusual setup. OrangeFS could not run the client on ARM 32-bit using the upstream kernel module. Additionally, we observed low read throughput if no direct I/O was used. For four clients reading a 2 GiB file each, only 12.41 MiB/s could be achieved. Consequently, measurements on OrangeFS are done with direct I/O.

Our prototype cannot compete with the throughput of the reference cluster. For real world HPC applications, more storage nodes need to be added to achieve higher throughput. This cluster was built as a proof-of-concept for throughput efficiency and to gain insight in ARM single-board computer storage clusters.

The different read and write sizes on both setups were chosen to achieve reasonable run-times of the benchmarks on both settings. Neither throughput nor throughput efficiency are influenced by the different amounts of transferred data if run-times are long enough.

Both clusters show good scaling behavior in all metrics. Exceptions occur for writes. On the reference cluster, one client achieves close to the observed maximum performance, and no further improvement can be seen when adding more clients. In addition, both Ceph-based systems only reached a fraction of the theoretical peak performance, as can be seen in Table II.

For the ARM cluster, this is most likely related to data replication over the public network. Ceph OSDs reported slow operation warnings due to waiting times for sub-operations. As pointed out by Just [43], the Ceph OSD service utilizes many threads, leading to performance issues for a few cores as context switches introduce additional overhead. Ceph's behaviour is strongly influenced by the number of placement groups per OSD [3]. While a higher ratio of placement groups to OSDs ensures a balanced data distribution, management of each placement group consumes memory and CPU time. To minimize overhead we set both pools to 64 placement groups. The number of placement groups per OSD also influences recovery behavior for larger clusters as more placement groups need to be replicated in case of a server crash. Further experiments are needed to evaluate different placement group counts and placement group to OSD ratios for productive usage of Ceph on large ARM clusters.

Nevertheless, replication cannot explain the performance drop for the reference cluster, which needs further investigation. One impacting factor for reads was that only one process

TABLE II. MAXIMUM THROUGHPUT ACHIEVED IN MIB/S AND PERCENT OF TPP.

System	Write / % TPP	Read / % TPP
ARM - CephFS	95.22 / 20.11	172.12 / 36.36
ARM - OrangeFS	289.23 / 61.10	296.82 / 62.70
Reference	2322.47 / 5.15	5705.0 / 12.65

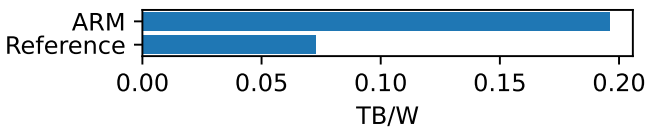


Figure 3. Storage capacity per Watt

per client was used, resulting in only one network stream, insufficient to saturate the network. This decision was made for comparability with the ARM cluster.

Both systems might be impacted by CephFS' lazy deletes [3], which are done asynchronously by an MDS and probably overlapped with reads and writes, resulting in lower throughput.

OrangeFS performs better than CephFS on ARM in nearly all measurements. In contrast to CephFS, the OrangeFS daemon is lightweight and does not use many threads. As a consequence, context switches introduce less overhead on low core counts. Because no replication is done between nodes, less data needs to be transferred via the network, and the management of replicas does not consume resources. The downside is that faults of nodes can lead to data loss. Even though performance is higher compared to CephFS, only about 60 % of the TPP (see Table II) can be achieved. This can certainly be improved by tuning the stripe size of OrangeFS and the record size of ZFS. Compared to the defaults of other parallel file systems, OrangeFS has a low default stripe size of 64 KiB. Further benchmarks should be done to evaluate bigger stripes which could result in larger disk accesses depending on server-side cache size and cache times. As shown by traces of MPI-IO calls and OrangeFS' internal Trove layer, which does the actual disk I/O, single client-side write calls can result in multiple server-side Trove write calls [44]. Those should align to ZFS record sizes, if possible, to minimize read-modify-write cycles.

Compared to the other metrics, the EDP, as shown in Figure 2 is a fused metric that measures performance and energy efficiency at once. The use of this metric for tuning storage systems enforces that balanced configurations are found. Neither performance nor energy-saving efforts are neglected in favor of the other one. Considering that OrangeFS achieves both higher performance and energy efficiency, the EDP of CephFS is up to 30 times higher.

In terms of capacity per Watt, the ARM cluster is superior to the reference cluster, achieving 2.68 more TB per Watt. The ARM cluster's low idle power and maximum power consumption allow for usage of the cluster in places or situations where power restrictions apply, enabling the usage as a mobile storage solution.

VI. Conclusion and Future Work

We evaluated CephFS for HPC workloads on a productive cluster based on traditional x86 servers and an ARM-based low-power cluster. We compared the results in terms of throughput and efficiency. The ARM cluster is able to provide more than twice as much TB per Watt as the reference cluster and can achieve comparable throughput efficiency. OrangeFS has been shown to perform better than CephFS on the ARM cluster. Due to the low idle power consumption and low power peaks, ARM-based storage solutions are helpful in situations where power restrictions apply, for example, when used as a mobile storage cluster. In summary, we have shown that the energy efficiency of storage solutions depends significantly on both the used architecture and the file system. Lightweight solutions can reduce energy consumption and thus cost, which is becoming increasingly important due to the exponentially growing volumes of data.

As a next step, we will evaluate the use of other parallel file systems, such as MooseFS, and compare the results with an x86 setup, which is more similar in terms of network and disks compared to the ARM-based cluster. Throughput scaling of the ARM cluster while adding more storage nodes needs to be measured, so that the use in real world applications can be evaluated. In addition to sequential throughput other workloads, such as metadata-focused or mixed workloads, are of interest.

Acknowledgment

This work is partly funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 417705296. More information about the CoSEMoS (Coupled Storage System for Efficient Management of Self-Describing Data Formats) project can be found at https://cosemos.de.

References

[1] J. G. Koomey, "Worldwide electricity used in data centers," *Environmental Research Letters*, vol. 3, no. 3, jul 2008. [Online]. Available: https://doi.org/10.1088/1748-9326/3/3/034008

[2] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: A Scalable, High-Performance Distributed File System," in *7th Symposium on Operating Systems Design and Implementation (OSDI '06), November 6-8, Seattle, WA, USA*, B. N. Bershad and J. C. Mogul, Eds. USENIX Association, 2006, pp. 307–320. [Online]. Available: http://www.usenix.org/events/osdi06/tech/weil.html

[3] Ceph authors and contributors, "Ceph Documentation," https://docs.ceph.com/en/latest, 2021, [retrieved: 04, 2022].

[4] M. M. D. Bonnie *et al.*, "OrangeFS: Advancing PVFS," in *USENIX Conference on File and Storage Technologies (FAST)*, 2011.

[5] K. Duwe and M. Kuhn, "Using Ceph's BlueStore as Object Storage in HPC Storage Framework," in *CHEOPS@EuroSys'21*. ACM, 2021, pp. 3:1–3:6.

[6] J. Edge, "The OrangeFS distributed filesystem," https://lwn.net/Articles/643165/, 2015, [retrieved: 04, 2022].

[7] R. Thakur, W. Gropp, and E. Lusk, "A Case for Using MPI's Derived Datatypes to Improve I/O Performance," in *Proceedings of SC98: High Performance Networking and Computing*. ACM Press, November 1998. [Online]. Available: http://www.mcs.anl.gov/~thakur/dtype/

[8] M. Vilayannur, R. Ross, P. Carns, R. Thakur, A. Sivasubramaniam, and M. Kandemir, "On the performance of the POSIX I/O interface to PVFS," in *12th Euromicro Conference on Parallel, Distributed and Network-Based Processing, 2004. Proceedings.*, 2004, pp. 332–339.

 ISBN: 978-1-61208-967-6

[9] Z. Ou, B. Pang, Y. Deng, J. K. Nurminen, A. Ylä-Jääski, and P. Hui, "Energy- and Cost-Efficiency Analysis of ARM-Based Clusters," in *CCGRID*. IEEE Computer Society, 2012, pp. 115–123.

[10] E. L. Padoin, D. A. G. de Oliveira, P. Velho, and P. O. A. Navaux, "Evaluating Performance and Energy on ARM-based Clusters for High Performance Computing," in *41st International Conference on Parallel Processing Workshops, ICPPW 2012, Pittsburgh, PA, USA, September 10-13, 2012*. IEEE Computer Society, 2012, pp. 165–172. [Online]. Available: https://doi.org/10.1109/ICPPW.2012.21

[11] N. Rajovic, A. Rico, N. Puzovic, C. Adeniyi-Jones, and A. Ramírez, "Tibidabo: Making the case for an ARM-based HPC system," *Future Gener. Comput. Syst.*, vol. 36, pp. 322–334, 2014. [Online]. Available: https://doi.org/10.1016/j.future.2013.07.013

[12] M. Sato *et al.*, "Co-Design for A64FX Manycore Processor and "Fugaku"," in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2020, pp. 1–15.

[13] D. Göddeke *et al.*, "Energy efficiency vs. performance of the numerical solution of PDEs: An application study on a low-power ARM-based cluster," *J. Comput. Phys.*, vol. 237, pp. 132–150, 2013. [Online]. Available: https://doi.org/10.1016/j.jcp.2012.11.031

[14] S. Brienza, S. E. Cebeci, S. S. Masoumzadeh, H. Hlavacs, Ö. Özkasap, and G. Anastasi, "A Survey on Energy Efficiency in P2P Systems: File Distribution, Content Streaming, and Epidemics," *ACM Comput. Surv.*, vol. 48, no. 3, pp. 36:1–36:37, 2016. [Online]. Available: https://doi.org/10.1145/2835374

[15] G. Lefebvre and M. J. Feeley, "Energy efficient peer-to-peer storage," Technical Report TR-2003-17. Department of Computer Science, University of British Columbia, Tech. Rep., 2000.

[16] M. Geveler, B. Reuter, V. Aizinger, D. Göddeke, and S. Turek, "Energy efficiency of the simulation of three-dimensional coastal ocean circulation on modern commodity and mobile processors," *Comput. Sci. Res. Dev.*, vol. 31, no. 4, pp. 225–234, 2016. [Online]. Available: https://doi.org/10.1007/s00450-016-0324-5

[17] F. Mantovani *et al.*, "Performance and energy consumption of HPC workloads on a cluster based on Arm ThunderX2 CPU," *CoRR*, vol. abs/2007.04868, pp. 800–818, 2020. [Online]. Available: https://arxiv.org/abs/2007.04868

[18] M. Ponce *et al.*, "Deploying a Top-100 Supercomputer for Large Parallel Workloads: the Niagara Supercomputer," in *PEARC*. ACM, 2019, pp. 34:1–34:8.

[19] D. Gudu and M. Hardt, "ARM Cluster for Performant and Energy-Efficient Storage," in *Computational Sustainability*, ser. Studies in Computational Intelligence, J. Lässig, K. Kersting, and K. Morik, Eds. Springer, 2016, vol. 645, pp. 265–276. [Online]. Available: https://doi.org/10.1007/978-3-319-31858-5_12

[20] A. Kougkas, A. Fleck, and X.-H. Sun, "Towards Energy Efficient Data Management in HPC: The Open Ethernet Drive Approach," in *2016 1st Joint International Workshop on Parallel Data Storage and data Intensive Scalable Computing Systems (PDSW-DISCS)*, 2016, p. 43–48.

[21] L. Zhang, Y. Deng, W. Zhu, J. Zhou, and F. Wang, "Skewly replicating hot data to construct a power-efficient storage cluster," *Journal of Network and Computer Applications*, vol. 50, pp. 168—-179, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804514001362

[22] X. Ruan *et al.*, "ECOS: An energy-efficient cluster storage system," in *2009 IEEE 28th International Performance Computing and Communications Conference*, 2009, p. 79–86.

[23] C. Karakoyunlu and J. A. Chandy, "Techniques for an energy aware parallel file system," in *2012 International Green Computing Conference, IGCC 2012, San Jose, CA, USA, June 4-8, 2012*. IEEE Computer Society, 2012, pp. 1–5. [Online]. Available: https://doi.org/10.1109/IGCC.2012.6322247

[24] P. Sehgal, V. Tarasov, and E. Zadok, "Evaluating Performance and Energy in File System Server Workloads," in *8th USENIX Conference on File and Storage Technologies, San Jose, CA, USA, February 23-26, 2010*, R. C. Burns and K. Keeton, Eds. USENIX, 2010, pp. 253–266. [Online]. Available: http://www.usenix.org/events/fast10/tech/full_papers/sehgal.pdf

[25] H. Shan and J. Shalf, "Using IOR to Analyze the I/O Performance for HPC Platforms," in *In: Cray User Group Conference (CUG'07)*, 2007.

[26] S. Rivoire, M. A. Shah, P. Ranganathan, C. Kozyrakis, and J. Meza, "Models and Metrics to Enable Energy-Efficiency Optimizations," *Computer*, vol. 40, no. 12, pp. 39–48, 2007.

[27] M. Horowitz, T. Indermaur, and R. Gonzalez, "Low-power digital design," in *Proceedings of 1994 IEEE Symposium on Low Power Electronics*, 1994, pp. 8–11.

[28] D. Chen *et al.*, "Usage centric green performance indicators," *SIGMETRICS Perform. Evaluation Rev.*, vol. 39, no. 3, pp. 92–96, 2011. [Online]. Available: https://doi.org/10.1145/2160803.2160868

[29] J. H. Laros III *et al.*, *Energy Delay Product*. London: Springer London, 2013, p. 51–55. [Online]. Available: https://doi.org/10.1007/978-1-4471-4492-2_8

[30] S. Georgiou, M. Kechagia, P. Louridas, and D. Spinellis, "What Are Your Programming Language's Energy-Delay Implications?" in *Proceedings of the 15th International Conference on Mining Software Repositories*, ser. MSR '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 303–313. [Online]. Available: https://doi.org/10.1145/3196398.3196414

[31] Super Micro Computer, Inc., "Supermicro AS 2124BT-HNTR Datasheet," https://www.supermicro.com/en/Aplus/system/2U/2124/AS-2124BT-HNTR.cfm, 2020,

[retrieved: 04, 2022].

[32] Intel Corporation, "Intel P4510 Datasheet," https://ark.intel.com/content/www/us/en/ark/products/122579/intel-ssd-dc-p4510-series-4-0tb-2-5in-pcie-3-1-x4-3d2-tlc.html, 2018, [retrieved: 04, 2022].

[33] GIGA-BYTE Technology Co., "Gigabyte R282-Z94 Datasheet," https://www.gigabyte.com/Enterprise/Rack-Server/R282-Z94-rev-100#Specifications, 2021, [retrieved: 04, 2022].

[34] Samsung, "Samsung MZQL23T8HCJS-00A07 Datasheet," https://semiconductor.samsung.com/ssd/datacenter-ssd/pm9a3/mzql23t8hcjs-00a07/, 2021, [retrieved: 04, 2022].

[35] HARDKERNEL CO., LTD., "Odroid HC4 Datasheet," https://wiki.odroid.com/odroid-hc4/hardware/hardware, 2021, [retrieved: 04, 2022].

[36] Armbian, "Armbian Odroid HC4," https://www.armbian.com/odroid-hc4/, 2022, [retrieved: 04, 2022].

[37] Western Digital Corporation, "WD Black WD10SPSX Datasheet," https://documents.westerndigital.com/content/dam/doc-library/en_us/assets/public/western-digital/product/internal-drives/wd-black-hdd/product-brief-western-digital-wd-black-mobile-hdd.pdf, 2020, [retrieved: 04, 2022].

[38] Samsung, "Samsung V-NAND SSD 860 PRO Datasheet," https://www.samsung.com/semiconductor/global.semi.static/Samsung_SSD_860_PRO_Data_Sheet_Rev1_1.pdf, 2018, [retrieved: 04, 2022].

[39] NETGEAR, Inc., "Netgear GS110EMX Datasheet," https://www.netgear.com/images/datasheet/switches/webmanagedswitches/GS110EMX_GS110MX.pdf, 2021, [retrieved: 04, 2022].

[40] MEAN WELL, "MW HRP 450-15 Datasheet," https://www.meanwell.com/webapp/product/search.aspx?prod=HRP-450, 2021, [retrieved: 04, 2022].

[41] ZES ZIMMER Electronic Systems GmbH, "ZES Zimmer LMG 450 Brochure," https://www.zes.com/en/content/download/286/2473/file/lmg450_prospekt_1002_e.pdf, 2010, [retrieved: 04, 2022].

[42] Dell Inc., "Dell Precision 3650 Tower Hardware Specification," https://www.delltechnologies.com/asset/en-us/products/workstations/technical-support/precision-3650-spec-sheet.pdf, 2021, [retrieved: 04, 2022].

[43] S. Just, "Crimson: A new ceph OSD for the age of persistent memory and fast NVMe storage," Santa Clara, CA, Feb. 2020.

[44] T. Ludwig, S. Krempel, J. Kunkel, F. Panse, and D. Withanage, "Tracing the MPI-IO Calls' Disk Accesses," in *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, B. Mohr, J. L. Träff, J. Worringen, and J. Dongarra, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 322–330.

 ISBN: 978-1-61208-967-6