e-ISSN: 2980-177X

JCEES

Volume: 3 Issue: 1 Year: 2025

Journal of Computer & Electrical and Electronics Engineering Sciences

MHA
MEDIHEALTH ACADEMY

Journal of
Computer & Electrical and
Electronics Engineering Sciences

EDITORIAL BOARD

Journal of
Computer & Electrical and Electronics Engineering Sciences

EDITORIAL BOARD

EDITORIAL BOARD

Dear Colleagues,

Academic journals play a vital role in the advancement of science by serving as platforms where researchers share their findings with the global scientific community. In this context, sustainability, consistency, and quality are fundamental pillars of successful academic publishing.

It is with great pride and pleasure that I present to you the fifth issue of our journal, Journal of Computer & Electrical and Electronics Engineering Sciences. Since the publication of our inaugural issue, our journal has continued its journey with a strong sense of responsibility towards academic and ethical publishing. With each new issue, we have reinforced our commitment to supporting original, high-quality research in the fields of Computer Engineering and Electrical-Electronics Engineering.

Our journal maintains its adherence to double-blind peer review processes and places utmost importance on ethical publishing principles, including the protection of personal data and academic integrity. As we continue to publish two issues per year, we remain focused on enhancing the scientific value and visibility of our publication.

The achievements we have reached so far would not have been possible without the dedicated efforts of our editorial board, reviewers, and authors. I would like to express my heartfelt gratitude to all who have contributed to our journal with their time, expertise, and passion for science.

As we move forward, we remain committed to our core values and to continuously improving the quality of our journal. We look forward to your continued support, valuable feedback, and insightful contributions that will help shape the future of our publication.

With sincere regards and best wishes,

Asst. Prof. Fuat TÜRK, PhD

Editor-in-Chief

Journal of
Computer & Electrical and
Electronics Engineering Sciences

CONTENTS

Volume: 3 Issue: 1 Year: 2025

ORIGINAL ARTICLES

Journal of
Computer & Electrical and Electronics Engineering Sciences

Original Article

DOI: 10.51271/JCEEES-0021

Effect of different transistor technologies on physical design of arithmatic logic unit

Merve Özcan Korum, Fırat Kaçar
Department of Electrical and Electronics Engineering, Cerrahpaşa Faculty of Engineering, İstanbul University, İstanbul, Turkiye

Cite this article as: Özcan Korum, M., & Kaçar, F. (2025). Effect of different transistor technologies on physical design of arithmatic logic unit. *J Comp Electr Electron Eng Sci,* 3(1),1-7.

Received: 15.10.2024 ◆ **Accepted**: 22.12.2024 ◆ **Published**: 17.04.2025

ABSTRACT

Aims: The aim of this open source physical design tool is to observe the impact of miniaturization in transistor technology.
Methods: This study encompasses the utilization of free or open-source software and the design processes involved in both digital and physical design phases within the realm of VLSI design. Additionally, open-source software was utilized as a synthesis tool during the physical design process. This approach allowed us to observe the impact of transistor technology shrinkage not only with licensed software but also with open-source alternatives. One distinguishing feature of this study lies in the verification of the design process through simulations, conducted both after the digital design phase and following the physical design phase. The physical design phase employed the open-source qflow software. Designs for 8-bit, 16-bit, and 32-bit arithmetic logic units (ALUs) were synthesized using transistor technologies of 180 nm and 350 nm.
Results: The results of the design experiments revealed that reducing the transistor technology scale for designs of the same bit length did not lead to a significant alteration in the number of standard cells comprising the design. However, the number of standard cells increased directly in proportion to the length. As technology continued to shrink, the physical area occupied by the design decreased, while static time analysis (STA) also diminished concurrently with technology scaling.
Conclusion: To promote the proliferation of VLSI design process-es, it is imperative to foster the development and utilization of open-source software. This approach facilitates the expansion of research and education in the field of VLSI design, ultimately augmenting the pool of skilled professionals in this domain
Keywords: Open source, qflow, transistor technology, physical design, arithmetic unit logic

INTRODUCTION

Transistors serve as the building blocks of chips, enabling the functionality of technological products in all aspects of our lives. Thanks to advancements in semiconductor technology, VLSI technology has become an indispensable element in today's world. VLSI technology offers both advantages and comes with expensive and time-consuming production processes. Essentially, the design phase can be divided into two main parts: behavioral design and physical design.

VLSI designs can be crafted professionally using licensed computer-aided design (CAD) programs, as well as open-source software. Some of the licensed CAD programs include Cadence, Synopsys, and Mentor. On the other hand, there are open-source alternatives such as qFlow, OpenLANE, Coriolis, VTR, and SymbiFlow (Chupilko et al., 2021).

The data processing center of the central processing unit is called the arithmetic logic unit (ALU). The ALU is one of the most important units in a microprocessor, where arithmetic and logic operations are performed. Arithmetic operations include addition, subtraction, multiplication, and division, while logical operations include AND, OR, and NOT. It is inevitable that different processors will have different arithmetic logic units, but fundamentally, every ALU includes the basic logical operations listed above (Aydın, M., 2022).

ALU designs find applications in both military and civilian sectors. The study involved the use of 8-bit, 16-bit, and 32-bit ALU designs. Within the framework of this thesis, the behavioral design and physical design stages of a basic-functionality ALU's digital integrated circuit design process were undertaken (Bedir, N., 2018). Behavioral design was conducted using Quartus II in the Verilog language and simulated with the ModelSim tool. For the physical synthesis process, the open-source qflow tool was employed, with the IRSIM tool serving as the simulation tool (Başak, S., 2011).

The behavioral designs of 8-bit, 16-bit, and 32-bit ALU designs were physically synthesized using 180 nm and 350 nm transistor technologies. Parameters such as the number of standard cells, occupied area, and static time analysis (STA) values were assessed. Furthermore, in addition to reviewing the literature, the design was subjected to ModelSim simulation following behavioral modeling. After physical synthesis, the accuracy of the design was validated using the IRSIM tool. Simulations from these two stages were compared

Corresponding Author: Merve Özcan Korum, merve.ozcan@ogr.iuc.edu.tr

across various bit lengths and synthesis processes in different transistor technologies. These comparisons revealed that transistor technology scaling did not have a detrimental impact on the design's functionality (Bakacak, M., 2021).

In the results of the thesis study, it was observed that as the bit length in the design increased, the number of cells used also increased in direct proportion to the bit length. The reduction in transistor technology scale employed during synthesis did not have an impact on the diversity of cells in the design. As transistor technology advances, i.e., as transistors shrink in size, the number of transistors within a unit area increases (Gaurav et al., 2022), (Purohit et al., 2021).

Through the synthesis of designs with varying bit lengths and two distinct transistor technologies, we observed their impact on the design's latency. Surprisingly, even as the bit length increased, the reduction in transistor technology managed to decrease the delay time. Furthermore, we examined the changes in latency resulting from the inclusion of capacitive and resistive values following physical synthesis in an open-source software. The obtained results align with previous studies in the literature (Sırmaçek, B., 2007).

The leading countries in the global semiconductor industry are China, the USA, Taiwan, and Japan. Taiwan meets approximately 60% of the world's semiconductor demand, while its closest competitor, South Korea, accounts for around 20%. The major global semiconductor manufacturers include Texas Instruments, Nvidia Corp, Micron Technology Inc. (MU), Qualcomm Inc. (QCOM), Intel, Samsung Electronics, Taiwan Semiconductor Manufacturing Company (TSMC), SK Hynix Inc., and Broadcom Inc. In Turkiye, significant advancements in chip design and production are being made through academic efforts led by METU MEMS and Bilkent UNAM. Additionally, under the supervision of TÜBİTAK YİTAL (semiconductor technologies research laboratory), companies such as ASELSAN, YONGATEK, and ERMEKSAN are actively involved in both production and sales within Turkiye (Ozcan, M.,2023) (**Table 1**).

Table 1. Semiconductor manufacturing facilities in Turkiye (Ozcan, M., 2023)

Product	Production facility	Production place
CMOS 250 nm	YİTAL Research Center	Gebze
CMOS/SiGe 130nm	YİTAL Research Center/ASELSAN	Gebze/Ankara
CMOS 90 u-65 nm	YİTAL A.Ş / ASELSAN	Gebze/Ankara
MEMS	ODTÜ-MEMS/ASELSAN	Ankara
GaN	AB Mikro Nano/ASELSAN	Ankara
GaAs (infrared)	ODTÜ Kanal, ASELSAN	Ankara
InGaAs (infrared)	ODTÜ Kanal, CÜNAM, ASELSAN	Ankara
HgCdTe (infrared)	ODTÜ Kanal, ASELSAN	Ankara
High power laser (500W CW)	ERMAKSAN	İstanbul

This study encompasses both behavioral design using a free version tool and the entire life cycle of VLSI design processes with the open-source Qflow tool. Consequently, it offers valuable insights and resources for individuals interested in the field of design and synthesis. Additionally, digital and physical designs were executed at different bit lengths and using two distinct transistor technologies, with the accuracy of these processes being confirmed through simulation studies conducted post-synthesis.

METHODS

Materials and methods should be described with sufficient detail to allow others to replicate and build on published results. New methods and protocols should be described in detail while well-established methods can be briefly described and appropriately cited. Give the name and version of any software used and make clear whether computer code used is available. Chapter name may be adapted to the topic of the manuscript or split into more chapters if it benefits the readability of the manuscript.

The first stage, known as behavioral synthesis, generates the gate-level (RTL) working state of the design. In the subsequent stage, the physical aspects of the design necessary to reach the chip size are executed. This stage is referred to as physical synthesis. The entire process represents the procedure for constructing application specific integrated circuits (ASICs). The ASIC design process is illustrated in **Figure 1**.

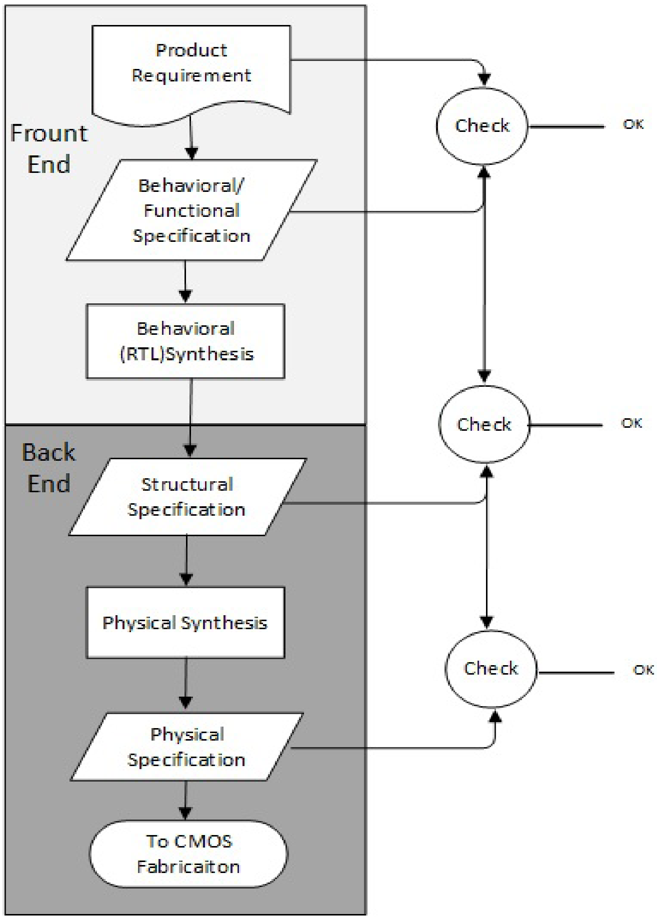

Figure 1. ASIC design process
ASIC: Application specific integrated circuit

Behavioral (Logic) Design

In ASIC development, behavioral design is often referred to as Front-end design. It typically involves four key steps (Purohit et al., 2021). The logic design process entails generating the RTL representation using a Hardware Description Language (HDL) to describe the functional behavior. Once the design steps are completed, each RTL module must undergo separate simulation to ensure accuracy.

Furthermore, IC (integrated circuit) design encompasses more than just logical designs; it also involves incorporating macros into the overall design. Fundamental macros include memory blocks, analog circuits, and input/output blocks.

During the logical design phase, modules for time adjustment and power management are also characterized.

The flowchart depicting the behavioral design processes is presented in Figure 2. The operations carried out in each step are explained sequentially.

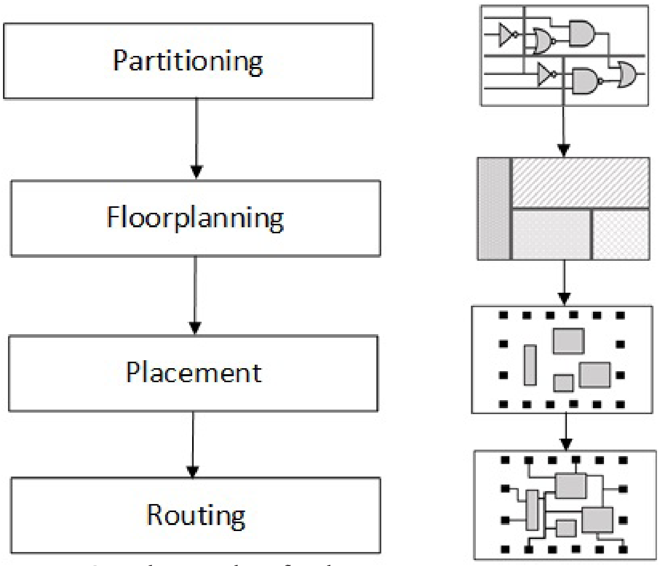

Figure 2. General gate synthesis flowchart

RTL-design entry: This is the stage where the design is represented using a HDL. During this phase, programming languages such as Verilog HDL or VHDL can be employed.

Generic netlist synthesis: This phase involves establishing connections between logic cells and cells generated from the HDL software. The logic synthesis phase encompasses steps that involve gate optimization and the conversion from a behavioral model to a structural model, regardless of transistor technology. The library utilized in this step obtains necessary timing and power information from the .lib file, which is based on previously defined data within the layout information (.lef) files.

Technology structure and optimization: It is mapped based on the alignment of logic gates from the standard cell library of the technology supported by the design software.

Gate level network list: This is the stage where the cells responsible for constructing the physical structure of the design are generated, along with the connection lists between them.

During the validation phase, if any errors are identified, the logic blocks are remapped. This process continues until the design criteria are enhanced either through adjustments made by the synthesis tool or optimizations achieved by resizing the gates.

In RTL design, timing-related errors often surface, which VLSI engineers address through extensive optimization efforts. In some cases, this optimization can be so profound that it necessitates a complete regeneration of the entire synthesis process.

Physical Design

In ASIC design, the phase referred to as physical design is commonly known as the back-end design. It essentially involves four main steps.

This process aims to create the detailed geometric layout required to transform the synthesized output at the gate level, obtained during the design phase, into the final form of the integrated circuit. It is also known as physical design, constituting a critical part of the digital VLSI workflow. During the physical layout, specific positions are assigned to the components that make up the design, and interconnections and routing are established either over or between metal layers using additive manufacturing technology. Once the graphic design system (GDSII) flow concludes as a result of the physical design, the output consists of a set of instructions for production.

The flowchart depicting the physical design processes is illustrated in Figure 3. The operations conducted at each step are subsequently elucidated. Modern physical production tools automate the entire process, significantly reducing the potential for human errors. This process is not limited to licensed physical design tools used in professional settings but also widely employs open-source tools.

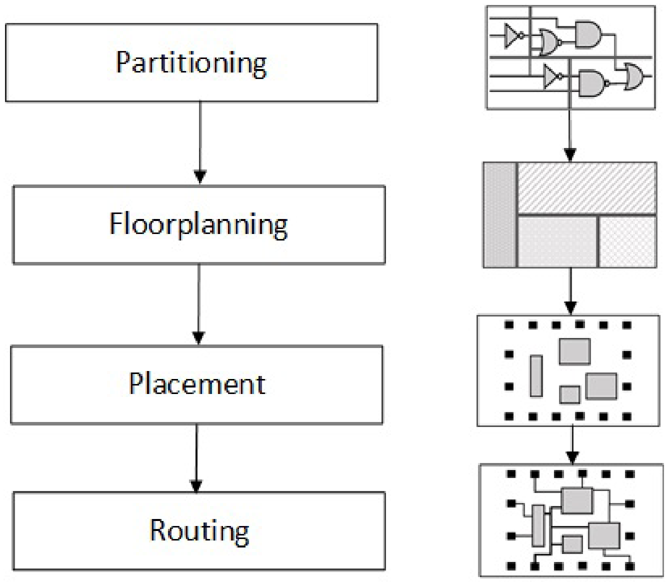

Figure 3. Physical design flowchart

Partitioning: Segmentation is a design approach wherein the larger modules or blocks constituting the design are broken down into smaller sub-components. This segmentation strategy is commonly employed to mitigate design complexity. Through this approach, each block can be individually designed or analyzed. Nevertheless, it's crucial to account for the connections between these blocks; otherwise, incorrect or incomplete connections can lead to performance degradation.

Floor planning: Each block structure can have either a fixed or flexible shape. Rigid blocks maintain fixed dimensions and areas, while flexible blocks have a fixed area but can vary in aspect ratios. This stage involves organizing both rigid and flexible blocks within the design according to the layout rules on the plane.

Placement: This stage involves making decisions regarding the placement of cells within the blocks of the design. All placements are predetermined to occupy specific areas. During this phase, timing constraints come into play, necessitating optimization to minimize the distance or connections between cells while preserving orientability.

An inefficient layout not only consumes more space but also gives rise to more timing errors. Furthermore, if time optimization is the primary objective during the placement step, it is possible to conserve space.

Routing: This stage involves establishing the connections between cells and blocks within the design. The routing process requires determining the topologies of the paths between standard cells and interconnects. The objective of concurrent routing optimization is to minimize the overall connection length and maximize timing efficiency.

Tool-QFLOW

The open-source Qflow tool is a toolchain that simulates the synthesis of a digital circuit design, originally modeled in Verilog, using specific manufacturing technology and processes.

In the commercial domain, widely-used and effective digital synthesis tools for chip design include software such as cadence and synopsys. These digital synthesis algorithms rely on closed-source software and are developed and controlled by various electronic design automation (EDA) software vendors. The prices of these software packages tend to be relatively high, making them less accessible for small and medium-sized digital synthesis projects.

Qflow 1.4 software is a computer-aided design (CAD) program used to transform RTL-modeled designs into GDS II layouts. This software is released under the General Public License (GPL), classifying it as open-source software. Qflow was originally developed by R. Timothy Edward in 2013, with the latest version published on 10th June 2021. Qflow comprises various sub-software components, including external tools such as Magic, Graywolf, and Yoys.

Figure 4 provides a schematic representation of the tools employed by the program in the synthesis process, including logical equivalence check (LEC) and Layout vs. Schematic (LVS) tools.

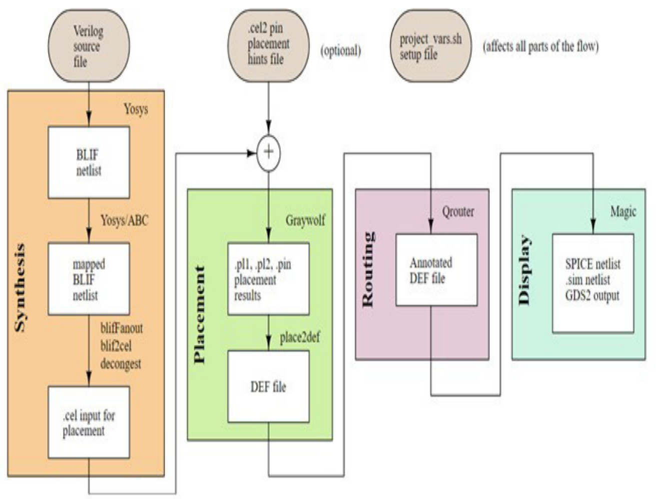

Figure 4. Qflow structural diagram

- Yosys (URL-1, 2022) and Odin II (URL-2, 2022) sub-softwares are used in the Logic Synthesis stage. As a result of this process, netlist links are created in BLIF format.
- The Graywolf (URL-3, 2022) subroutine in the layout and route phase is used in the initial block layouts and routing, and the qrouter (URL-4, 2022) subroutine is used in the specific routing step.
- OpenSTA (URL-5, 2022or vesta (URL-6, 2022) subroutines are used for Static Time Analysis (STA).
- The Magic (URL-7, 2022) subroutine is used for design check (DRC).
- Yosy and Netgen (URL-8, 2022) subprograms are used for LEC and LVC operations, respectively.

Qflow relies on MOSIS's "scalable CMOS rules" as the foundation, with open-source code serving as the standard cell library. Qflow specifically utilizes the open-source cell library from Oklahoma State University (OSU) This library encompasses OSU's 180-500 nm and FreePDK 45 nm technology cell libraries (Purohit et al., 2021). It's worth noting that industry leaders like X-Fab, IBM, or TSMC, which have their production facilities, typically do not disclose their standard cell libraries for sharing. The Oflow screen page is shown in Figure 5.

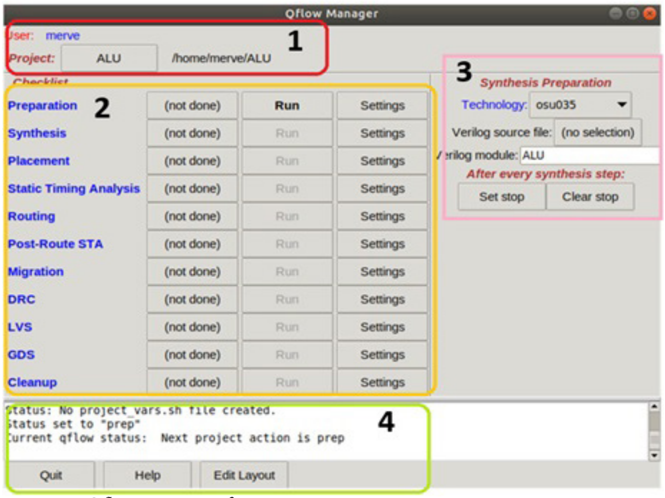

Figure 5. Qflow user interface

Section 1: This section provides the file path where the project is located.

Section 2: This section enumerates the steps conducted during the physical synthesis process. Each line includes checkboxes for initiating the operation, marking the operation as completed, and unique settings pertinent to that specific step.

Various transistor technologies were employed, including 180 nm and 350 nm technologies, for synthesizing ALU designs with different bit lengths. The ALU designs encompassed 8-bit, 16-bit, and 32-bit configurations.

Logic designs were developed using the Verilog language within the Quartus II environment. The logic design was simulated using Modelsim Altera 10.1d. Subsequently, for the physical design phase, the open-source Qflow 1.4 software was employed. Additionally, the accuracy of the physical design was confirmed through IRSIM simulation.

Data from the log/synth.log, log/place.log, and log/synth.log files have been tabulated for further analysis using graphic.

RESULTS

After synthesizing ALU designs with different bit lengths using the same transistor technology, we examined the impact of bit length on the number of standard cells. Following the synthesis of 8-bit, 16-bit, and 32-bit ALU designs with 180 nm transistor technology, the data regarding the standard cell count is recorded in the place.log file, which is one of the Qflow tool's logs. The information extracted from this file is presented in Figure 6. It's evident that as the bit length of the design increases, there is a proportional increase in the number of cells composing the design. This finding aligns with the existing literature (Kang & Leblebici, 2005).

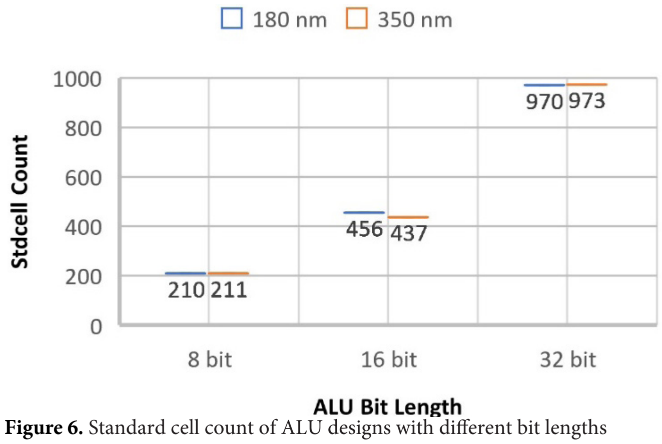

Figure 6. Standard cell count of ALU designs with different bit lengths

ALU: Arithmetic logic unit

Testbenches were created for IRSIM following the synthesis using 180nm and 350nm technologies with Qflow, and the designs were verified through simulations. The layout data in the generated log folder for different technology files is presented in Table 2. Similarly, the static timing analysis (STA) data in the log folder for different technology files is provided in Table 2.

DISCUSSION

As a result of synthesizing ALU designs with identical bit lengths but different transistor technologies, we observed the impact of changes in transistor technology on the occupied design area.

As a result of the physical synthesis of the 8-bit ALU design according to 180 nm transistor technology, the standard cell count was calculated as 211, the cell width as 5.89e+04 µm, the cell height as 2.11e+05 µm, the area covered by the design as 5.89e+07 µm² and the average cell height as 1.00e+03 µm. In addition, the maximum switching time of this design was calculated as 1114.15 ps and the minimum delay time as 187.453 ps.

As a result of the physical synthesis of the 16-bit ALU design according to 180 nm transistor technology, the standard cell count was calculated as 456, the cell width as 1.29e+05 µm, the cell height as 4.56e+05 µm, the area covered by the design as 1.29e+08 µm² and the average cell height as 1.00e+03 µm. In addition, the maximum switching time of this design was calculated as 1482.58 ps and the minimum delay time as 187.453 ps.

As a result of the physical synthesis of the 32-bit ALU design according to 180 nm transistor technology, the standard cell count was calculated as 970, the cell width as 2.74e+05 µm, the cell height as 9.70e+05 µm, the area covered by the design as 2.74e+08 µm² and the average cell height as 1.00e+03 µm. In addition, the maximum switching time of this design was calculated as 1769.47 ps and the minimum delay time as 187.453 ps.

As a result of the physical synthesis of the 8-bit ALU design according to 350 nm transistor technology, the standard cell count was calculated as 210, the cell width as 1.16e+05 µm, the cell height as 4.20e+05 µm, the area covered by the design as 2.31e+08 µm² and the average cell height as 2.00e+03 µm. In addition, the maximum switching time of this design was calculated as 1925.81 ps and the minimum delay time as 267.201 ps.

As a result of the physical synthesis of the 16-bit ALU design according to 350 nm transistor technology, the standard cell count was calculated as 437, the cell width as 2.45e+05 µm, the cell height as 8.74e+05 µm, the area covered by the design as 4.89e+08 µm² and the average cell height as 2.00e+03 µm. In addition, the maximum switching time of this design was calculated as 2451.97 ps and the minimum delay time as 267.201 ps.

As a result of the physical synthesis of the 32-bit ALU design according to 350 nm transistor technology, the standard cell count was calculated as 973, the cell width as 5.36e+05 µm, the cell height as 1.95e+06 µm, the design area as 1.07e+09 µm² and the average cell height as 2.00e+03 µm. In addition, the maximum switching time of this design was calculated as 3041.15 ps and the minimum delay time as 267.201 ps.

After successfully synthesizing them using Qflow, the area occupied by the designs is recorded in the place.log file. Figure 7 presents the area occupied by 8-bit, 16-bit, and 32-bit ALU designs with 180 nm transistor technology, along with synthesis results conducted with 350 nm transistor technology, which are also included in the same table.

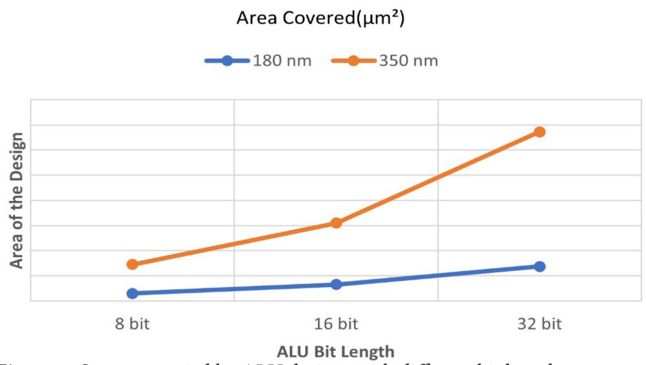

Figure 7. Space occupied by ALU designs with different bit lengths

ALU: Arithmetic logic unit

Table 2. Place.log and sta.log data in 180-350 nm technology

ALU design	Technology	SD cells	Cell width (µm)	Cell height (µm)	Design area (µm²)	Avg. cell height (µm)	Max transition time (ps)	Min delay (ps)
8-bit	180 nm	211	5.89e+04	2.11e+05	5.89e+07	1.00e+03	1114.15	187.453
16-bit	180 nm	456	1.29e+05	4.56e+05	1.29e+08	1.00e+03	1482.58	187.453
32-bit	180 nm	970	2.74e+05	9.70e+05	2.74e+08	1.00e+03	1769.47	187.453
8-bit	350 nm	210	1.16e+05	4.20e+05	2.31e+08	2.00e+03	1925.81	267.201
16-bit	350 nm	437	2.45e+05	8.74e+05	4.89e+08	2.00e+03	2451.97	267.201
32-bit	350 nm	973	5.36e+05	1.95e+06	1.07e+09	2.00e+03	3041.15	267.201

ALU: Arithmetic logic unit, SD: Standard deviation, Avg.: Averaj, Max: Maximum, Min: Minimum

As the transistor technology used in physical synthesis shrinks, the space occupied by designs with the same bit length decreases. However, when the bit length of the designs synthesized using the same transistor technology increases, the occupied design area also increases. These findings align with results reported in existing literature (Cao, 2021).

As a result of synthesizing ALU designs with varying bit lengths using the same transistor technologies, we observed the delays within the designs. Following successful synthesis with Qflow, delay information for the designs is recorded in the sta.log file. Figure 8 provides pre and post static time analysis (STA) values for 8-bit, 16-bit, and 32-bit ALU designs using 180nm transistor technology. The preSTA value in the table represents the value calculated during the rough layout phase in physical synthesis. Subsequently, post STA values, accounting for capacitive effects, were calculated to observe delays using open-source software.

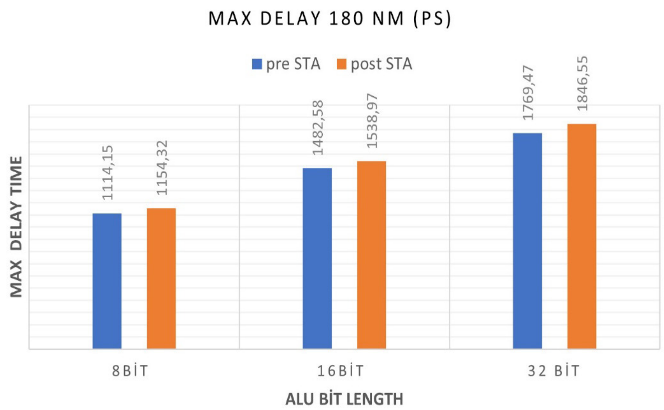

Figure 8. Pre and post STA times in 180 nm technology

STA: Static time analysis

As evident in the table, the post STA value for the 8-bit ALU design exceeds the preSTA value, and the delay time increases as the bit length increases. These observations align with findings reported in existing literature (Cao, 2021).

Using 350 nm transistor technology, we conducted physical synthesis for 8-bit, 16-bit, and 32-bit ALU designs and observed the latency of the designs. The results obtained from the sta.log file following the synthesis with 350 nm transistor technology are presented in Figure 9 Notably, both pre and post static time analysis (STA) values have increased when compared to 180 nm transistor technology. This evaluation underscores a significant result that aligns with the trend of shorter processing times for designs as transistor technology scales down.

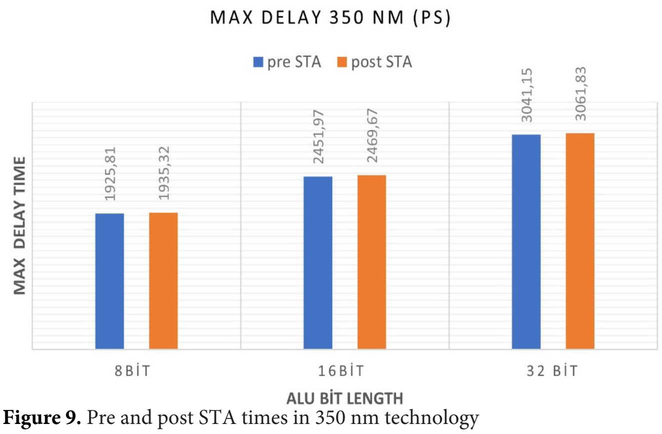

Figure 9. Pre and post STA times in 350 nm technology

STA: Static time analysis

In the thesis study, simulations were conducted to validate the accuracy of both the logic design and physical design stages. These simulations were replicated after synthesizing 8-bit, 16-bit, and 32-bit ALU designs using both 180 nm and 350 nm transistor technologies. The results obtained were found to align with the simulations conducted during the logic design and physical design phases. Figure 10 provides an overview of the ALU design processes and the anticipated simulation results compared to the referenced initial values. For further clarification, Figure 10 illustrates the outputs of the ModelSim simulation conducted for verifying the 8-bit ALU design, while Figure 11 showcases the IRSIM simulation results following physical synthesis in the 180 nm transistor technology.

In both the ModelSim output shown in Figure 10 and the IRSIM output displayed in Figure 11, the expected output for the "OP" value of "00110011" (hexadecimal 51) and "00000001" (hexadecimal 1) is indeed "00110011" (hexadecimal 51) and "00000001" (hexadecimal 1) in binary format, forming part of a sequence of 8 values being processed. This consistent result is observed across both simulations. Moreover, the alignment with the data in Table 3 is maintained in other transactions as well.

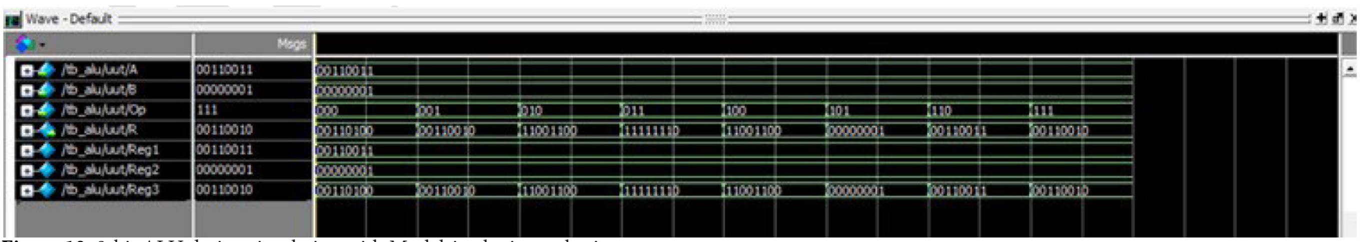

Figure 10. 8-bit ALU design simulation with Modelsim-logic synthesis

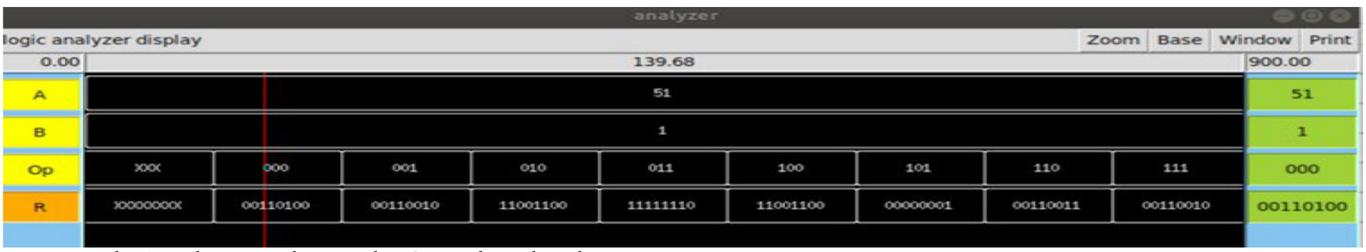

Figure 11. 8-bit ALU design simulation with IRSIM – physical synthesis

Table 3. Evaluation of Modelsim and Irsim results

Op	ALU process	Reg 1/A	Reg 2/B	ModelSim	Irsim
0	Addition	00110011	0000001	00110100	00110100
1	Subtraction	00110011	0000001	00110010	00110010
2	Not gate	00110011	0000001	11001100	11001100
3	Nand gate	00110011	0000001	11111110	11111110
4	Nor gate	00110011	0000001	11001100	11001100
5	And gate	00110011	0000001	00000001	00000001
6	Or gate	00110011	0000001	00110011	00110011

ALU: Arithmetic logic unit

CONCLUSION

This study encompassed a comprehensive simulation analysis, incorporating an open-source tool, following physical synthesis. Additionally, the accuracy of the designs was rigorously validated through simulation comparisons conducted after both logical and physical synthesis phases. The investigation examined the impact of shrinking transistor technology on standard cell counts for different bit lengths, the footprint of designs, and design delays, all assessed using open-source tools. Within the scope of this thesis, changes in transistor technology relative to design bit length were observed using open-source software. The reduction in transistor technology didn't yield a significant shift in the number of standard cells constituting the design. However, the advancement in transistor technology led to a reduction in the design's footprint, despite an increase in bit length. Furthermore, post-physical synthesis, design latency decreased as transistor technology scaled down. This study significantly contributes to the literature by verifying design accuracy through simulations conducted across different transistor technologies using open-source tools. To promote the proliferation of VLSI design processes, it is imperative to foster the development and utilization of open-source software. This approach facilitates the expansion of research and education in the field of VLSI design, ultimately augmenting the pool of skilled professionals in this domain.

ETHICAL DECLARATIONS

Referee Evaluation Process

Externally peer-reviewed.

Conflict of Interest Statement

The authors have no conflicts of interest to declare.

Financial Disclosure

The authors declared that this study has received no financial support.

Author Contributions

All of the authors declare that they have all participated in the design, execution, and analysis of the paper, and that they have approved the final version.

REFERENCES

Chupilko, M., Kamkin, A., & Smolov, S. (2021). Survey of open-source flows for digital hardware design. *Ivannikov Memorial Workshop (IVMEM)*, 11-16. doi:10.1109/IVMEM53963.2021.00008

Gaurav, T., Patel, K., & Parekh, R. (2022). RTL to GDSII implementation of RADIX-4 booth multiplier. *Int Confer Nanoelectr Nanophot Nanomater Nanobiosci Nanotechnol (5NANO)*, 1-5. doi:10.1109/5NANO 53044.2022.9828885

Purohit, S., Laddha, P., & Parekh, R. (2021). Implementation and physical design of 8/4-bit signed divider. *8th Int Confer Signal Process Integrat Networks (SPIN)*, 829-834. doi:10.1109/SPIN52536.2021.9566020

URL-1. (2022). Yosys open synthesis suite. http://www.clifford.at/yosys (date of access: 03-10-2022).

URL-2. (2022). Odin II. https://docs.verilogtorouting.org/en/latest/odin (date of access: 03-10-2022).

URL-3. (2022). Graywolf. https://github.com/rubund/graywolf (date of access: 03-10-2022)

URL-4. (2022). Qrouter. http://opencircuitdesign.com/qrouter (date of access: 03-10-2022)

URL-5. (2022). OpenSTA engine. https://github.com/The-OpenROAD-Project/OpenSTA (date of access: 03-10-2022).

URL-6. (2022). vesta. https://github.com/RTimothyEdwards/qflow/blob/master/src/vesta.c (date of access: 03-10-2022).

URL-7. (2022). vesta. https://github.com/RTimothyEdwards/qflow/blob/master/src/vesta.c (date of access: 03-10-2022).

URL-8. (2022). Netgen 1.5. http://opencircuitdesign.com/netgen (date of access: 03-10-2022).

Kang, S., & Leblebici, Y. (2005). Introduction. In CMOS digital integrated circuits: analysis and design (Tata McGraw Hill ed., Chapter 1, pp. 1-8).

Cao, L. (2021). Multi-objective digital VLSI design optimisation (doctoral dissertation). University of York.

Aydın, M. (2022). Detection of printed circuit board errors with FPGA-based real-time image processing. İstanbul University-Cerrahpaşa.

Bedir, N. (2018). FPGA based 64-bit arithmetic logic unit design. Thesis (PhD), İstanbul University-Cerrahpaşa.

Başak, S. (2011). FPGA based synthesizable processor design. Thesis (PhD), Yıldız Teknik University.

Sırmaçek, B. (2007. Modeling learning algorithm for mobile robot with FPGA. Thesis (PhD), Yıldız Teknik University.

Bakacak, M. (2021). A basic microprocessor design and embedded system implementation for educational purposes. Thesis (PhD), Kırıkkale University.

Ozcan, M. (2023). Different nanometer technologies on ALU (aritmetic logic unit) design. İstanbul University-Cerrahpaşa.

Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0022

Android malware detection with image-based features using transfer learning methods

Oğuzhan Sezer, İbrahim Alper Doğru, Kazım Kılıç, İsmail Atacak
Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara, Turkiye

Cite this article as: Sezer, O., Doğru, İ.A., Kılıç, K., & Atacak, İ. (2025). Android malware detection with image-based features using transfer learning methods. *J Comp Electr Electron Eng Sci,* 3(1),8-13.

Received: 04.11.2024 • **Accepted:** 02.01.2025 • **Published:** 17.04.2025

ABSTRACT

The increase in mobile device usage today has also brought security threats to Android devices. Ensuring the security of Android devices is crucial for protecting user data. Numerous studies have been conducted that encompass both traditional and AI-based methods to secure Android devices. This study aims to enrich the literature by evaluating and comparing the performance of three transfer learning models (ResNet50, DenseNet201, and VGG16) on a hybrid dataset adapted with specific parameters. The dataset obtained from AndroZoo and Drebin classifies the data as benign and malicious for effective malware detection. This research used grayscale images from the dataset to train the aforementioned transfer learning models. The results show that transfer learning models can be successfully used in malware detection. It has been observed that the VGG16 model achieved the best performance in malware detection with an accuracy of 97.24%, a precision of 97.26%, a recall of 97.24%, and an Auc value of 99.33%. These findings may also provide valuable contributions to developing mobile security applications.

Keywords: Mobile Security, Android Malware Detection, Transfer Learning, Mobile App Security, Deep Learning

INTRODUCTION

The rapid increase in internet-connected devices has made cybersecurity increasingly crucial for protecting digital assets from cyberattacks. Over time, the number of devices connected to the Internet has increased significantly and has affected various industries. Organizations and critical sectors equipped with technologies such as the Internet of Things (IoT), healthcare and medical devices, and Internet banking have found broader access opportunities through the Internet. In addition, a wide range of users, such as educational institutions, businesses, and banking, benefit from the services offered over the Internet (Kumar and Panda, 2023; Almomani et al., 2022).

Due to its openness and popularity, the Android operating system is the primary target for attacks on smartphones, accounting for approximately 98.5% of such attacks, with malware being the primary vector. The most worrying thing is that users access Internet services without understanding the impact of cyber threats on the Internet (Kumar and Panda, 2023; Almomani et al., 2022). The services provided by the Internet affect our daily lives. As cyber threats evolve, the situation may become dangerous or even life-threatening (Kumar and Panda, 2023). By 2023, about 5.44 billion users have had access to the Internet. Statistics show that the Internet will continue to transform people's daily lives by providing services that are easily accessible. In 2021, more than half of the people in the world, about 5 billion people, had access to the Internet (Kumar and Panda, 2023).

Static and dynamic analysis methods are generally used to detect Android malware. Static analysis examines the source code of malware without running it (Almomani et al., 2022), and while cost-effective, it can be insufficient to identify zero-day attacks and obfuscated malware (Almomani et al., 2022; Ban et al., 2022). Dynamic analysis observes the real-time malware behaviors in isolated virtual machines; however, this method may have limitations as the malware can alter its behavior in response (Almomani et al., 2022).

Using traditional machine learning methods, the researchers combined static and dynamic analysis features used in malware detection to aim for higher accuracy values. However, these methods are time-consuming and complex, often requiring manual feature selection (Ban et al., 2022). Deep learning methods have demonstrated superior accuracy in malware detection by eliminating the need for manual feature selection. These methods leverage various features, including API calls and permissions, to effectively identify malicious software (Ban et al., 2022).

In recent years, deep learning methods have increasingly been used in malware detection. Image-based models have often been preferred because they achieve high accuracy in malware detection. These models allow us to study the behavior of malware in the file structure (Kumar and Panda, 2023; Almomani et al., 2022; Ban et al., 2022; Zou et al., 2022). The primary problems addressed in this study are the increasing threat level of Android malware, the inadequacies

Corresponding Author: Oğuzhan Sezer, oguzhansezer@gazi.edu.tr

of existing detection methods, and the lack of frequent joint usage of datasets in the literature. This study aims to apply three different transfer learning methods to a hybrid dataset constructed with specific parameters. The rationale behind employing a hybrid dataset is to create a unique and distinct dataset. The limited use of hybrid datasets is highlighted as an advantage in the literature, as it allows for creating a unique dataset. Additionally, merging multiple datasets is considered a factor that reduces overfitting. Another significant advantage is that, even if one dataset contains corrupted or missing data, the hybrid dataset mitigates such issues, ensuring data integrity.

Six thousand data were used from the Androzoo dataset. In addition, 5100 data were planned to be used from the Drebin dataset. However, 42 data were lost due to errors in the creation of the Drebin dataset. A total of 11058 data were obtained. Malware detection was performed on three different transfer learning models using grayscale images from the hybrid dataset. The results of these models were compared with similar studies in the literature, and it was determined which model worked best on this new dataset.

The main contributions of this study are as follows:

- **Use of a hybrid dataset:** By leveraging the advantages of hybrid datasets rarely utilized in the literature, a balanced and unique dataset was created by combining the Androzoo and Drebin datasets. This approach offers benefits such as preventing data loss and reducing the risk of overfitting.
- **Utilization of grayscale images:** Grayscale images were used for Android malware detection, and this method enabled the evaluation of the performance of different transfer learning models. By converting the images into pixel matrices, the behavioral patterns of malware were effectively analyzed.
- **Comparison of three transfer learning models:** Widely used transfer learning models, including ResNet50, DenseNet201, and VGG16, were compared on the same dataset, and their impact on Android malware detection was investigated. This comparison revealed which model achieved higher accuracy.
- **Malware detection using static analysis methods:** The study focused on malware detection through static analysis methods and demonstrated that these methods achieve high accuracy rates.

METHODS

Androzoo is a dataset that works on mobile devices with the Android operating system. This dataset was created so that applications can be analyzed. It is also frequently used in mobile security research. Androzoo contains a huge number of Android app examples. In this way, it is beneficial to researchers. Samples on Androzoo have been obtained in various ways from the Google Play Store and other sources. The dataset contains many unique features. In this way, they have made great contributions to mobile security analysis (Allix et al., 2016). In the AndroZoo dataset, filtering was applied to select APK files that meet specific criteria for analysis. APK files with a VirusTotal detection rate of 10 or more were considered for the malicious dataset. Additionally, the analysis focused exclusively on APKs smaller than 2 MB, targeting less complex and smaller-sized files. Lastly, APKs sourced from the Google Play Store were selected, limiting the scope to applications from this specific marketplace. For the benign dataset, APKs with a VirusTotal detection rate of zero were utilized. Similarly, APKs smaller than 2 MB and obtained from the Google Play Store were chosen to ensure security and consistency.

Drebin was developed to detect Android malware. Performs analysis using static analysis methods. It is not necessary to run Android applications while using Drebin. Drebin makes classifications based on application characteristics. In this way, threats are detected, and reports are generated. Drebin is used in areas such as malware detection and security analysis (Arp et al., 2014). The APK files included in the dataset predominantly represent malicious software and are categorized based on specific behavioral traits. The APKs were collected from both official sources and third-party application stores. The files in the dataset are generally smaller than 10 MB, enabling fast and efficient analysis. The APKs were labeled as either malicious or benign using tools like VirusTotal and analyzed based on their behaviors.

A total of 11058 data obtained from Drebin and Androzoo datasets were used in this study. This dataset is divided into 5529 benign APKs and approximately 5529 malignant APKs. APKs created for benign and malignant were obtained from both Drebin and Androzoo datasets. The large number of samples in the data set made the study more efficient. Two different classes were used in the data set: benign and malignant. 70% of the dataset was used for training, 20% for testing, and 10% for validation.

The datasets used in this study, Androzoo, and Drebin, comprise approximately 5,529 benign and 5,529 malware samples, making it a balanced dataset. The 11,058 APK files that were obtained were converted into Classes.dex files. From the converted files, the classes.dex file was further transformed to obtain byte files. These byte files were converted into pixel matrices ranging between 0-255. This process was applied to transform the image data into pixels. Finally, these pixel matrices were used to obtain grayscale images. Thus, the data obtained from the APK files were converted into grayscale data to be used for Android malware detection. The dimensions of the grayscale images are set to 255*255.

In this study, the TensorFlow library, commonly used in deep learning, was utilized. Data augmentation was performed using the ImageGenerator function in TensorFlow. ImageGenerator does not physically augment the dataset. Data augmentation was applied to the training dataset, which was split by 70%. Diversifying image data helped the model learn more generalized features. Transformations such as flipping, rotating, and zooming images allowed the model to learn from different perspectives. Additionally, data augmentation increased the diversity of the training dataset, reducing the risk of overfitting. Finally, results were obtained by training with three different transfer learning models. These trained models were compared.

Figure 1 shows the overall architecture of the model used in this study.

Transfer learning is the process of applying the experiences obtained by a trained model to a new problem. Here, three different transfer learning methods are examined.

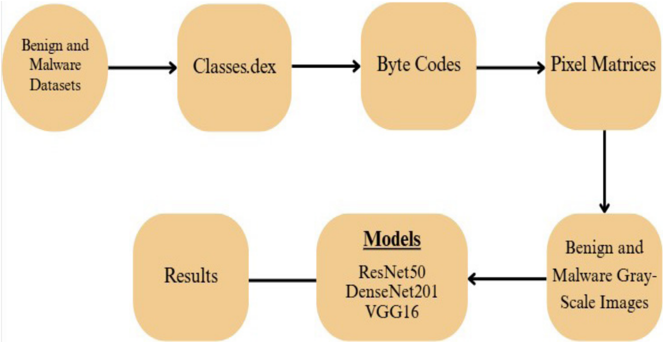

Figure 1. Architecture of the model used

ResNet50 is a convolutional neural network (CNN) model developed by Microsoft Research. This transfer learning model uses an approach known as "skip connection" (Kumar and Panda, 2023). This model consists of 50 layers. These layer blocks include convolutional layers and ReLU activation functions (Mukti and Biswas, 2019). The "skip connection" structure allows the flow of information to switch between layers. Resnet50 is frequently used in tasks such as object recognition and classification. (Mukti and Biswas, 2019; Abdulazeez et al., 2024). ResNet50 is a pre-trained CNN model that incorporates a 50-layer deep architecture with residual connections. The final layers of the model were adapted to address the classification problem in this study, generating outputs for two classes: benign and malware. The output layer consists of two fully connected layers followed by a softmax activation function. To mitigate overfitting, dropout rates of 0.25 and 0.5 were applied, respectively.

The Adam optimization algorithm was utilized during training. The Adam optimizer adjusts momentum and adaptive learning rates with beta values set to 0.9 and 0.99, respectively. The weight decay value was set to 0.01 to regularize the model's parameters. The model loss was calculated using the cross entropy loss function.

The batch size during training was set to 64. The learning rate was defined as 0.001, as recommended by FastAI, and a one-cycle learning rate scheduling strategy was adopted. This approach allows for an efficient and dynamic adjustment of the learning rate throughout the training process, improving convergence and generalization. The model was trained for 50 epochs.

DenseNet201 is a convolutional neural network model. This model contains 201 layers. It uses the output of each layer as an input to all the layers after it. In this way, it ensures efficiency (Pramesti et al., 2023). This structure facilitates deep learning by improving the flow of information. It also allows the model to represent the feature map with a small number of parameters. These features enable DenseNet201 to achieve high accuracy rates in training deep networks (Altaf et al., 2023). The final layers of the DenseNet201 model were restructured to suit the classification problem addressed in this study. The model was tailored for two classes (benign and malware), with the output layer comprising two fully connected layers followed by a softmax activation function. To prevent overfitting, dropout rates of 0.25 and 0.5 were applied sequentially.

The Adam optimization algorithm was employed during training, which adjusts momentum and adaptive learning rates with beta values set to 0.9 and 0.99. The model loss was calculated using the Cross-Entropy Loss function to minimize the discrepancy between the classes. The batch size used during the training process was set to 64.

The initial learning rate was defined as 0.001. The ReduceLROnPlateau method was utilized to adjust the learning rate dynamically. This method reduces the learning rate by a factor of 0.2 if the accuracy metric does not improve for five consecutive epochs. The model was trained for a total of 50 epochs.

The VGG16 model is a 16-layer convolutional neural network. These layers consist of 13 convolutional and three fully connected layers. Convolutional layers learn features from images using 3x3 filters. Max pooling layers, on the other hand, reduce the size of feature maps, reducing computational cost. The model classifies through fully connected layers using flattened feature maps. In addition, thanks to the ReLU activation functions, learning efficiency increases (Kumar and Panda, 2023). VGG16 with batch normalization is a pre-trained CNN model that enhances stability during training. The final layers of the model were restructured to address the binary classification problem between benign and malware classes. The output layer consists of two fully connected layers and a softmax activation function, which generates probabilities for the two target classes.

The Adam optimization algorithm was utilized for model optimization, effectively handling momentum and adaptive learning rate adjustments. The beta values for optimization were set to 0.9 and 0.99. To regularize the model, the weight decay parameter was set to 1e-3 (0.001). The model loss was computed using the cross-entropy loss function, which is suitable for minimizing the discrepancy between the predicted and actual class labels in classification tasks.

During training, a batch size of 64 was chosen to balance computational efficiency and convergence performance. The learning rate was initialized with the min_grad_lr parameter, which was calculated during the learning rate finder step. This method ensures an optimal starting learning rate for training. Dynamic learning rate adjustments were implemented using the ReduceLROnPlateau method, which reduces the learning rate by a factor of 0.2 when the accuracy metric fails to improve.

The model was trained for a total of 50 epochs using a one-cycle learning policy, facilitating efficient training and convergence.

RESULTS

The experiment was conducted on a server with an Intel® 12700K CPU running at 2.40 GHz, an NVIDIA GeForce RTX 3070 Ti GPU, and 64 GB of RAM. Jupyter Notebook was used as part of the Python code editor for development. Several commonly used Python packages, such as vision and callbacks, were used to process large data volumes. Fastai libraries were employed for deep-learning image processing. The graphical representation of performance metrics was done using visualization tools like Matplotlib and Seaborn.

Accuracy is the ratio of the samples that the model correctly predicted to the total number of predicted samples. Recall is the ratio of the number of samples that the model correctly identifies as positive to the total number of true positive samples. Precision is the ratio of the model's true positive predictions to its total positive predictions (Kumar and Panda, 2023).

- **TP (true positive):** True positives are examples that are correctly identified as malware (Kumar and Panda, 2023).
- **TN (true negative):** True negatives are examples that are correctly identified as harmless (Kumar and Panda, 2023).
- **FP (false positive):** False positives are examples defined as malware even though they are harmless (Kumar and Panda, 2023).
- **FN (false negative):** False negatives are examples defined as harmless even though they are malware (Kumar and Panda, 2023).

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{Recall}=\frac{TP}{TP+FN} \quad (2)$$

$$\text{Precision}=\frac{TP}{TP+FP} \quad (3)$$

Image-based detection is a method in computer security that analyzes visual content to detect malware or security threats. This approach focuses on application icons, interface elements, screenshots, and other visual components to identify potential risks (Zou et al., 2022).

The accuracy graph of the ResNet50 model is shown in Figure 2.

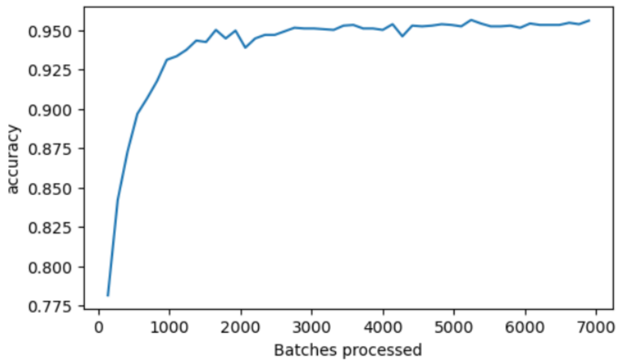

Figure 2. Accuracy graph of the ResNet50 model

Figure 2 illustrates that as iterations on the training set progress, there is an initial sharp increase, indicating that the model enters a rapid learning phase. After approximately 3000 iterations, the accuracy rate stabilizes at a high level. This demonstrates the effectiveness of the model's overall learning capacity and the success of the training process.

The confusion matrix values of the ResNet50 model are shown in Figure 3.

Figure 3 shows that the model detects malicious and benign software with generally high accuracy; it makes 1090 true positive and 1025 true negative predictions, with only a small number of false positives (66) and false negatives (30).

The accuracy graph of the DenseNet201 model is shown in Figure 4.

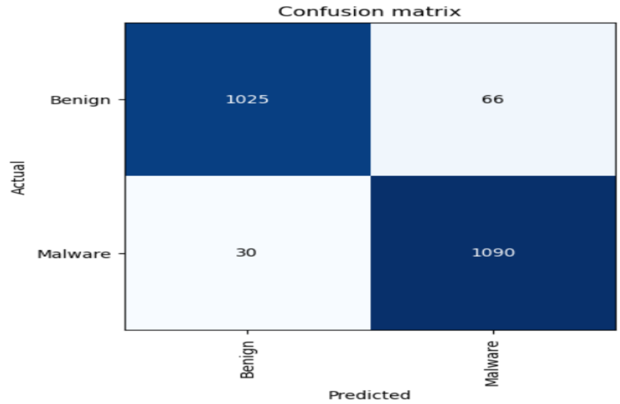

Figure 3. Confusion matrix values of the ResNet50 model

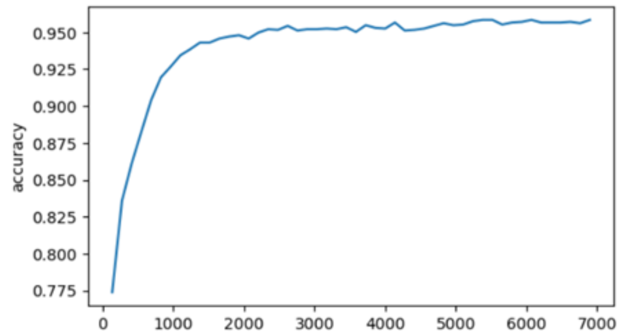

Figure 4. Accuracy graph of the DenseNet201 model

Figure 4 reflects an effective learning process with a rapidly increasing accuracy rate at the beginning of the training, demonstrating stable performance with approximately 95% accuracy after 3000 iterations.

The confusion matrix values of the DenseNet201 model are shown in Figure 5.

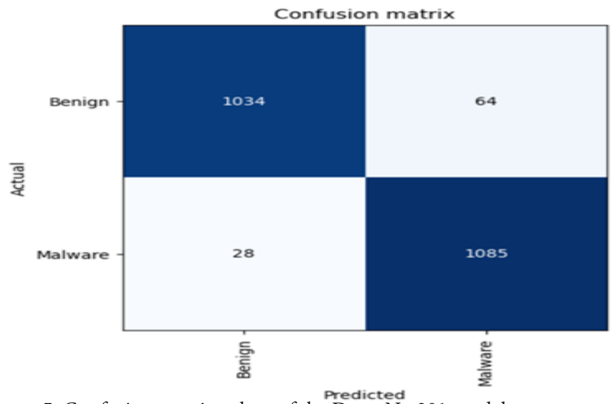

Figure 5. Confusion matrix values of the DenseNet201 model

Figure 5 shows that the model distinguishes between malicious and benign software with generally high accuracy. The model made 1085 true positive and 1034 true negative predictions, with only 64 false positive and 28 false negative errors. These results demonstrate that the model can effectively recognize malicious and benign software.

The accuracy graph of the VGG16 model is shown in Figure 6.

Figure 6 shows that during the training process, the model initially exhibits rapid improvement, surpassing 90% accuracy, and achieves values between 96% and 98% after approximately 4000 iterations. This indicates that the model reaches high performance relatively early and maintains this performance throughout the training, highlighting its stable and reliable learning dynamics.

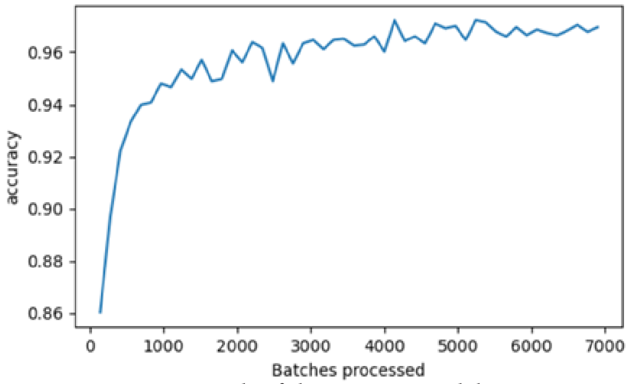

Figure 6. Accuracy graph of the VGG16 model

The confusion matrix values of the VGG16 model are shown in Figure 7.

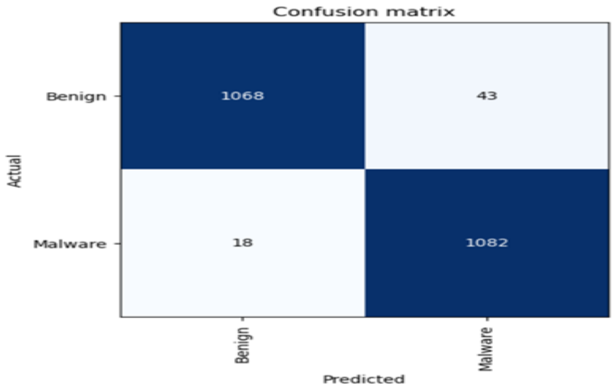

Figure 7. Confusion matrix values of the VGG16 model

Figure 7 demonstrates that the model is highly effective in distinguishing between malicious and benign software. The model successfully classified malicious software with 1082 true positive predictions and benign software with 1068 true negative predictions, making only 43 false positive and 18 false negative errors. These results indicate that the model can detect malicious and benign software with high accuracy and perform reliably in these tasks.

Table 1 presents the performance results of three deep-learning models for Android malware detection. The ResNet50 model shows balanced performance with 95.65% accuracy, 95.72% precision, and 95.63% recall, while its Auroc value of 98.69% indicates high success. DenseNet201 delivers similar results to ResNet50 with 95.83% accuracy and 95.89% precision, demonstrating high sensitivity in detecting malware with 95.82% recall and a remarkable ability to handle misleading examples with a 98.98% Auroc value. The VGG16 model offers more precise results than the other two models, with 97.24% accuracy and 97.26% precision, effectively detecting malware with a 97.24% recall and exhibiting exceptionally high classification performance with a 99.33% Auroc value. Each model possesses strong predictive capabilities despite deceptive or challenging examples, demonstrating their proficiency in Android malware detection.

Table 1. Overall results of the trained models

	Evaluation metrics			
Trained models	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**AUC (%)**
Resnet50	95.65	95.72	95.63	98.69
DenseNet201	95.83	95.89	95.82	98.98
VGG16	97.24	97.26	97.24	99.33

AUC: Area under curve, ResNet50, DenseNet201, and VGG16: Three transfer learning models

Table 2 presents various studies in Android malware detection, comparing the models used, datasets, analysis methods, and the accuracy rates achieved. The MSAE and SHLMD models used static analysis with the MUDFLOW and VirusShare datasets to achieve 94.46% accuracy (Zhu et al., 2022). In comparison, the MADRF-CNN model achieved 96.9% accuracy using the Google Play Store and VirusShare datasets (Zhu et al., 2023). A group of transfer learning models reached an accuracy rate of 95.83% using static analysis on the CICInvesAndMal20 dataset (Ksibi et al., 2024). The Bi-LSTM model achieved an accuracy of 94.12% using dynamic analysis on the KISA-data challenge 2019-Malware dataset (Jeon et al., 2022). The GuardDroid model achieved an accuracy of 91.74% using both dynamic and static analysis on the MUDFLOW dataset (Vasu et al., 2023).

Limitations

In this study, the VGG16 model achieved 97.24% accuracy with static analysis on the Androzoo and Drebin datasets, while the DenseNet201 and ResNet50 models reached 95.83% and 95.65%, respectively. Each of these studies highlights the interaction of different datasets and analysis methods for malware detection, emphasizing the diversity of the field. This comparative table is a valuable resource for evaluating the advantages and limitations of different approaches.

CONCLUSION

This research examines deep learning-based models as effective tools for combating malware threats on the Android platform. With the increasing accessibility of the Internet and

Table 2. Comparison of related works

	Compared features			
Studies	**Models**	**Datasets**	**Method**	**Accuracy (%)**
Zhu et al., 2022.	MSAE, SHLMD	MUDFLOW, VirusShare	Static	94.46
Zhu et al., 2023.	MADRF-CNN	Google Play Store and VirusShare	Static	96.9
Ksibi et al., 2024.	DenseNet169, Xception, InceptionV3, ResNet50 and VGG16	CICInvesandMal2019	Static	95.83
Jeon et al., 2022.	Bi-LSTM	KISA-data challenge 2019-Malware	Static dynamic	94.12
Vasu et al., 2023.	GuardDroid	MUDFLOW	Dynamic static	91.74
This study	VGG16	Androzoo and Drebin	Static	97.24
	DenseNet201	Androzoo and Drebin	Static	95.83
	Resnet50	Androzoo and Drebin	Static	95.65

MSAE: Mean squared approximation error, ResNet50, DenseNet201, and VGG16: Three transfer learning models, Bi-LSTM: Bidirectional long short-term memory the integration of various functionalities, mobile devices have become attractive targets for cybercriminals. In this context, deep learning models such as ResNet50, DenseNet201, and VGG16 were evaluated through static analysis using the AndroZoo and Drebin datasets. Transfer learning models demonstrated high accuracy, precision, and recall rates, proving their effectiveness in identifying such threats. Notably, the VGG16 model outperformed others with an accuracy of 97.24% and an AUC score of 99.33%, highlighting its superior performance in classifying malware and distinguishing deceptive samples. This study achieved better results than previous works and partially addressed the challenge of hybrid datasets, which have been underrepresented in the literature. In conclusion, this study underscores the potential of deep learning models for Android malware detection and reveals how different datasets can lead to varying outcomes. Additionally, the strengths and limitations of existing methodologies were highlighted by evaluating the proposed models within a comparative framework against other studies in the literature. Future research should focus on enhancing these models' performance by incorporating more datasets and advanced analytical methods. Hybrid models should also be utilized alongside hybrid datasets. Specifically, a combination of transfer learning methods, machine learning algorithms, and attention mechanisms can facilitate the development of diverse hybrid models. Furthermore, in addition to image processing, audio-based analyses should also be explored. Such advancements will strengthen Android security solutions, ultimately aiding end-users and organizations in mitigating malware threats.

ETHICAL DECLARATIONS

Referee Evaluation Process

Externally peer-reviewed.

Conflict of Interest Statement

The authors have no conflicts of interest to declare.

Financial Disclosure

The authors declared that this study has received no financial support.

Author Contributions

All of the authors declare that they have all participated in the design, execution, and analysis of the paper, and that they have approved the final version.

REFERENCES

Abdulazeez, F.A., Ahmed, I.T., & Hammad, B.T. (2024). Examining the performance of various pretrained convolutional neural network models in malware detection. *Appl Sci*, 14(6),2614. doi:10.3390/app14062614

Allix, K., Bissyandé, T.F., Klein, J., & Le Traon, Y. (2016, May). Androzoo: collecting millions of android apps for the research community. In Proceedings of the 13th international conference on mining software repositories (pp. 468-471).

Almomani, I., Alkhayer, A., & El-Shafai, W. (2022). An automated vision-based deep learning model for efficient detection of android malware attacks. *IEEE Access*, 10,2700-2720. doi:10.1109/ACCESS.2022.3140341

Altaf, Y., Wahid, A., & Kirmani, M.M. (2023, February). Deep learning approach for sign language recognition using densenet201 with transfer learning. In 2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1-6). IEEE.

Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., Rieck, K., & Siemens, C.E.R.T. (2014, February). Drebin: effective and explainable detection of android malware in your pocket. In Ndss (Vol. 14, No. 1, pp. 23-26).

Ban, Y., Lee, S., Song, D., Cho, H., & Yi, J.H. (2022). Fam: featuring android malware for deep learning-based familial analysis. *IEEE Access*, 10, 20008-20018. doi:10.1109/ACCESS.2022.3151357

Chaganti, R., Ravi, V., & Pham, T.D. (2022). Image-based malware representation approach with EfficientNet convolutional neural networks for effective malware classification. *J Informat Security Appl*, 69,103306. doi:10.1016/j.jisa.2022.103306

Chen, S., Lang, B., Liu, H., Chen, Y., & Song, Y. (2024). Android malware detection method based on graph attention networks and deep fusion of multimodal features. *Expert Systems Appl*, 237,121617. doi:10.1016/j.eswa.2023.121617

Jeon, J., Jeong, B., Baek, S., & Jeong, Y.S. (2021). Hybrid malware detection based on Bi-LSTM and SPP-Net for smart IoT. *IEEE Transact Industr Informat*, 18(7),4830-4837. doi:10.1109/TII.2021.3119778

Ksibi, A., Zakariah, M., Almuqren, L., & Alluhaidan, A.S. (2024). Efficient android malware identification with limited training data utilizing multiple convolution neural network techniques. *Eng Appl Artific Intell*, 127,107390. doi:10.1016/j.engappai.2023.107390

Kumar, S., & Panda, K. (2023). SDIF-CNN: stacking deep image features using fine-tuned convolution neural network models for real-world malware detection and classification. *Appl Soft Comput*, 146,110676. doi: 10.1016/j.asoc.2023.110676

Mukti, I.Z., & Biswas, D. (2019, December). Transfer learning based plant diseases detection using ResNet50. In 2019 4th International conference on electrical information and communication technology (EICT) (pp. 1-6). IEEE.

Pramesti, E., Mutiara, A.B., & Refianti, R. (2023, December). Implementation of Deep Learning using convolutional neural network for skin disease classification with DenseNet-201 architecture. In 2023 Eighth International Conference on Informatics and Computing (ICIC) (pp. 1-6). IEEE.

Vasu, T., Fiza, S., Kumar, A.K., Devi, V.S., Kumar, C.N., & Kubra, A. (2023). Improved chimp optimization algorithm (ICOA) feature selection and deep neural network framework for internet of things (IoT) based android malware detection. *Measur Sensors*, 28,100785. doi: 10.1016/j.measen.2023.100785

Yang, S., Wang, Y., Xu, H., Xu, F., & Chen, M. (2022). An android malware detection and classification approach based on contrastive lerning. *Comput Securit*, 123,102915. doi:10.1016/j.cose.2022.102915

Zhu, H., Wei, H., Wang, L., Xu, Z., & Sheng, V.S. (2023). An effective end-to-end android malware detection method. *Expert Syst Appl*, 218,119593. doi:10.1016/j.eswa.2023.119593

Zhu, H.J., Wang, L.M., Zhong, S., Li, Y., & Sheng, V.S. (2021). A hybrid deep network framework for android malware detection. *IEEE Transact Knowl Data Eng*, 34(12),5558-5570. doi:10.1109/TKDE.2021.3067658

Zou, B., Cao, C., Tao, F., & Wang, L. (2022). IMCLNet: a lightweight deep neural network for ımage-based malware classification. *J Informat Security Appl*, 70,103313. doi:10.1016/j.jisa.2022.103313

Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0023

Hourly energy consumption forecasting by LSTM and ARIMA methods

Ali Nasser Ali, Taha Etem
Department of Computer Engineering, Faculty of Engineering, Çankırı Karatekin University, Çankırı, Turkiye

Cite this article as: Ali, A.N., & Etem, T. (2025). Hourly energy consumption forecasting by LSTM and ARIMA methods. *J Comp Electr Electron Eng Sci,* 3(1),14-20.

Received: 23.10.2024 • **Accepted**: 08.01.2025 • **Published**: 17.04.2025

ABSTRACT
Energy consumption represents the overall quantity of energy necessary for a specific activity or process, typically quantified in kilowatt-hours (kWh). In 2019, global electricity final consumption amounted to 22.848 terawatt-hours (TWh), indicating a 1.7% growth compared to 2018. During the same period, the Organization for Economic Co-operation and Development (OECD) recorded a total electricity final consumption of 9.672 TWh, reflecting a 1.1% decrease from the previous year. Conversely, non-OECD countries experienced a rise in final electricity consumption, reaching 13.176 TWh, marking a 3.8% increase over the 2018 figures. Time series analysis is a statistical method that examines data collected over successive time intervals. By identifying patterns and trends in historical data, this approach facilitates predictions and forecasts about future values. In our problem, forecasting and estimating the energy consumption in megawatts for the west regions of USA have achieved a good performance, with 97% for the R-squared metrics for LSTM and 98% for ARIMA models.
Keywords: Energy consumption, time series analysis, long short-term memory, LSTM, auto-regressive integrated moving average, ARIMA

INTRODUCTION

Energy consumption refers to the total amount of energy used over a specified period, typically measured in units like kilowatt-hours (KWh) or joules. It encompasses the utilization of various energy sources, such as electricity, fuel, or renewable resources, for activities ranging from household operations and indusratrial processes to transportation. Understanding and monitoring energy consumption are crucial for assessing resource efficiency, environmental impact, and the development of sustainable practices (Energy consumption: definition & types | StudySmarter, n.d.).

Economic trends indicate a significant deceleration in the global growth of energy consumption in 2022, with the growth rate dropping by half, from +4.9% in 2021 to 2.1% in 2022. Despite this slowdown, the 2022 growth rate remains above the average observed during the period from 2010 to 2019, which was +1.4% per year. In 2022, the growth of energy consumption decelerated in the world's two largest consuming countries. China, holding the position of the largest energy consumer globally with a 25% share in 2022, experienced a 3% increase, down from the +5.2% growth observed in 2021. Similarly, the United States saw a rise of 1.8%, compared to the +4.9% growth in the previous year. Noteworthy expansions in energy consumption were driven by robust economic growth in India (+7.3%), Indonesia (+21%), and Saudi Arabia (+8.4%). Additionally, there were moderate increases in Canada (+3.8%) and Latin America (+2.7%), including gains in Brazil and Mexico (+2.4%), and a more substantial rise in Argentina (+4.5%). The Middle East and Africa also experienced a collective growth of around 3%, despite a 4.5% decline in South Africa attributed to coal supply tensions and mandated load shedding in the power sector (World Energy Consumption Statistics | Enerdata, n.d.).

In 2021, Halit Çetiner and İbrahim Çetiner employed various machine learning algorithms based on artificial intelligence to predict energy consumption. The models implemented in their work include XGBoost, LSTM algorithms, classical linear regression, and RANSAC. Among the models they used, the LSTM showed good performance in all the measuring metrics with an r-squared of 99.3% (Çetiner & Çetiner, 2021).

In 2021, Wei Qin, Linhong Wang, Yuhan Liu and Cheng Xu developed a support vector machine regression (SVR) model using variables like state of charge (SOC) and environmental factors. The grey wolf optimization (GWO) algorithm was used to select optimal parameters, resulting in the proposed GROW-SVR model. Validation on real bus routes showed a mean average percentage error of 14.47% and mean average error of 0.7776. The model also outperformed other methods in both accuracy and training time (Qin et al., 2021).

In 2022, Mohammed Nasih Ismael employed a variety of artificial intelligence-based regression models to predict energy consumption. The implemented models in their study

Corresponding Author: Taha Etem, tahaetem@karatekin.edu.tr

encompass LSTM for the prediction. The model they used demonstrated superior performance with an r-squared of 95% (Ismael, 2022).

In 2023, Wei Cai, Xiaodong Wen, Chaoen Li, Jingjing Shao and Jianguo Xu used the SVR-supervised machine learning algorithms, which offer strong predictive accuracy but lacks a defined rule for parameter fitting. To solve this, they explored six meta-heuristic optimization algorithms and found the best fit for the SVR model. Their analysis showed the hybrid buildings, with R^2 values of 0.9975 and 0.99955 for cooling and heating load predictions, respectively, in the training data (Cai et al., 2023). In **Table 1**, a comparison with the literature is shown (Xiong et al., 2014), (Purlu et al., 2022), (Son & Van Cuong, 2023), (Irfan et al., 2023).

Our study builds upon these efforts, focusing on the prediction of energy consumption using machine learning models already existing in libraries such as statsmodels, Tensorflow and a dataset called hourly energy consumption obtained from Kaggle (10. PJME_hourly, n.d.), which provides features of the house value. While prior studies on PJME hourly electricity consumption forecasting have predominantly relied on conventional statistical methods or standard machine learning models, our research introduces a hybrid deep learning architecture integrating LSTM and easy to apply autoregressive, integrated moving average (ARIMA) model. This novel combination addresses non-linear consumption patterns and enhances the model's responsiveness to exogenous variables that are often overlooked in traditional models. By doing so, our study not only improves prediction accuracy but also contributes a framework for adaptive load forecasting that can be extended to other regional markets.

In the following section, we will provide a detailed overview of the dataset and the employed models. We will then go through the exploration and analysis of the data, explaining the methods used. Finally, we will compare the performance of the models based on metrics such as MSE, MAE, RMSE, and R2 Squared, providing insights into their effectiveness in predicting median house value.

METHODS

The method proposed for this study consists of five components, as illustrated in **Figure 1**, working collaboratively to attain the study objectives. Initially, the process involves collecting the dataset, followed by subsequent pre-processing steps. In the pre-processing steps, we checked for missing data and the presence of outliers. Then, the split-validation technique was employed to divide the dataset into training samples, encompassing data from 2002 to 2018, and the data specifically from the year 2018 for testing.

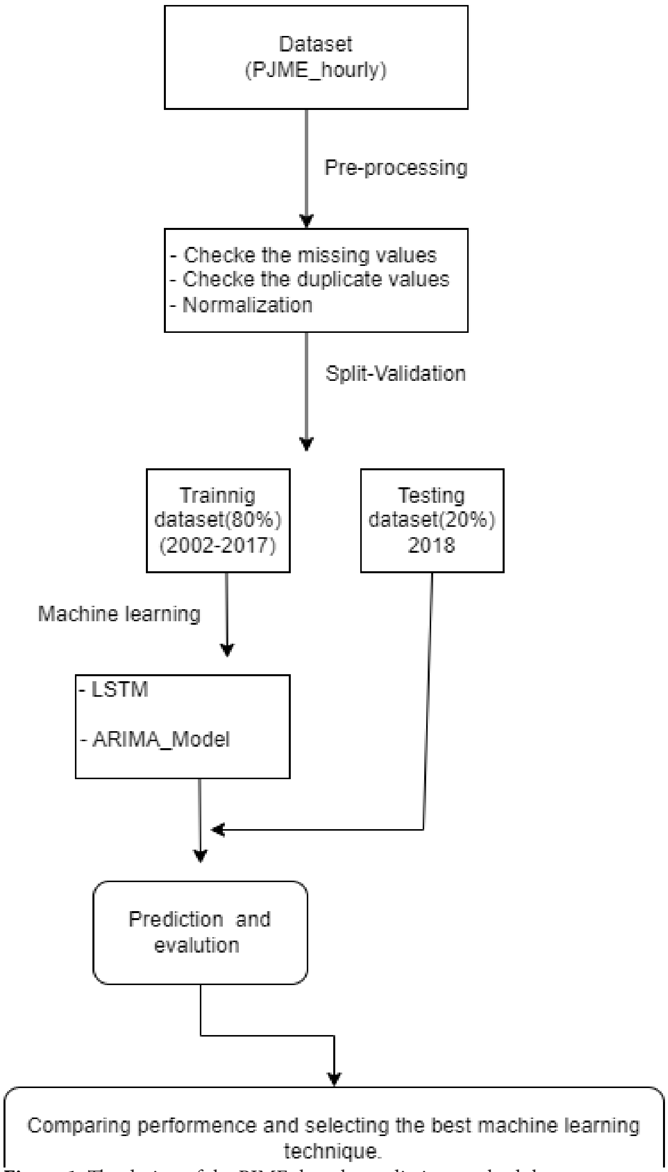

Figure 1. The design of the PJME_hourly prediction methodology

PJME: Preheated Jatropha Methyl Ester

Next, we applied specific models for time prediction, including the LSTM and ARIMA models. These algorithms were trained, tested, and evaluated to achieve the best performance. The evaluation process involved assessing key metrics such as MSE, MAE, and R-squared to comprehensively evaluate the effectiveness of each model in achieving the study objectives. However, we will focus primarily on the R-squared metric.

Table 1. Comparison with the literature

Source	Data source	Model (s) tested	Performance metrics	Advantages	Disadvantages
Xiong et al., 2014	PJM dataset	SVR, BEMD	Just Prediction Results	The lower and upper bounds are decomposed well	None of the performance metrics are evaluated.
Purlu et al., 2022	Energy consumption data	ANN	MAE, MSE, R^2	Low error rate	Potential overfitting without careful tuning
Son & Van Cuong, 2023	Kaggle PJME dataset	NARX	MSE, MAPE	Educational, demonstrates preprocessing	Limited rigorous validation
Irfan et al., 2023	Kaggle PJME dataset	Ensemble DNN	XGBoost: MAPE ~4.5%, RMSE ~900 MW**	Broad overview of methods, guidance on methodology	Excessive computational power need
This study	Kaggle PJME dataset	ARIMA, LSTM	MAE, MSE, RMSE, R^2	Addresses non-linear consumption patterns well	Dataset limitations

PJM: Pennsylvania-New Jersey-Maryland Interconnection, PJME: Preheated Jatropha Methyl Ester, SVR: Support vector machine regression, BEMD: Bi-dimensional empirical mode decomposition, ANN: Artificial neural networks, NARX: Nonlinear autoregressive exogenous, ARIMA: Autoregressive, integrated moving average, LSTM: Long short-term memory, MAE: Mean_absolute_error, MSE: Mean_squared_error, MAPE: Mean absolute percent error, RMSE: Root_mean_square_error, MW: Megawatts

PJMW Dataset

The dataset used in our study, called PJMW_hourly, stands for Pennsylvania-New Jersey-Maryland Interconnection (PJM), a regional transmission organization (RTO) in the United States, with 'W' representing the west region, and 'hourly' indicating the consumption of estimated energy in megawatts (MW) per hour. This dataset, obtained from Kaggle (10. PJME_hourly, n.d.), is a time series dataset of energy consumption from an eastern state region in the USA.

As indicated in Table 1, the dataset comprises 145,366 entries, with one feature and one target variable. The features of this dataset include:

PJMW_MW: Information on energy consumption in Megawatts.

Datetime: Date and time information.

Long Short-Term Memory

Long short-term memory (LSTM) networks are a type of recurrent neural network (RNN) that are well-suited for sequence prediction tasks, such as time series forecasting, because they can capture long-term dependencies in the data (Koşar & Barshan, 2023).

Before delving into LSTM, let's briefly explain the difference between recurrent neural networks (RNN) and traditional neural networks (TNN). TNNs are limited in their ability to comprehend sequences of data, such as events in a movie, because they cannot retain information from previous events to inform subsequent ones. Recurrent neural networks (RNNs) address this limitation by incorporating loops that enable information to persist from one step to the next. Essentially, RNNs can be visualized as multiple copies of the same network passing messages to each other, making them well-suited for handling sequences and lists of data. They have achieved significant success in tasks such as speech recognition, language modeling, translation, and image (Understanding LSTM Networks-Colah's Blog, n.d.).

In Figure 2 above, a segment of a neural network denoted as a takes an input named xt and generates an output referred to as ht. A loop is present, facilitating the movement of information from one network step to the next (Understanding LSTM Networks-Colah's Blog, n.d.).

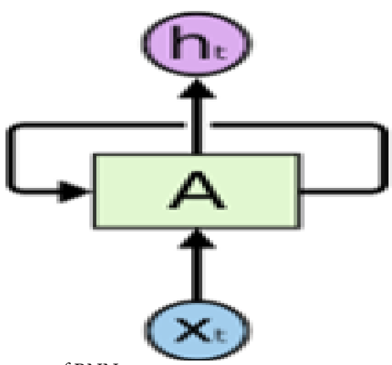

Figure 2. A segment of RNN

RNN: Recurrent neural networks

The loops within recurrent neural networks (RNNs) may appear mysterious initially, but they share similarities with regular neural networks. We can conceptualize an RNN as numerous replicas of the same network exchanging messages among themselves, as depicted in the Figure 3 below.

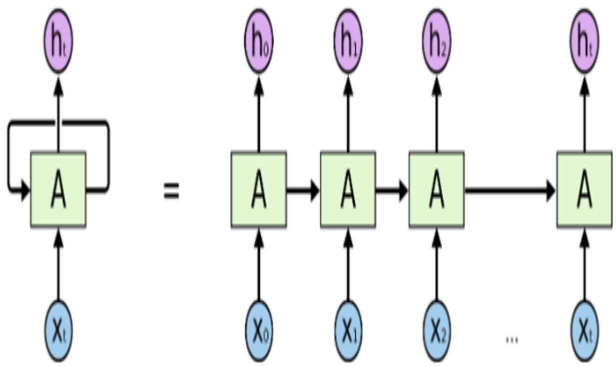

Figure 3. A general representation segment of RNN

RNN: Recurrent neural networks

When we unroll this loop, it forms a chain-like structure, as illustrated in the figure above, highlighting the suitability of RNNs for handling sequences and lists of data, which aligns with our problem statement. RNNs have demonstrated success in various domains such as speech recognition, language modeling, translation, and image captioning (Understanding LSTM Networks-Colah's Blog, n.d.).

However, RNNs are most effective when the information is recent, meaning the gap between relevant information and its application is small, as shown in the Figure 4 below in bold blue. For instance, when predicting the next word in a sentence like 'I like to drink Turkish tea', the word 'tea' is predictable without requiring further context.

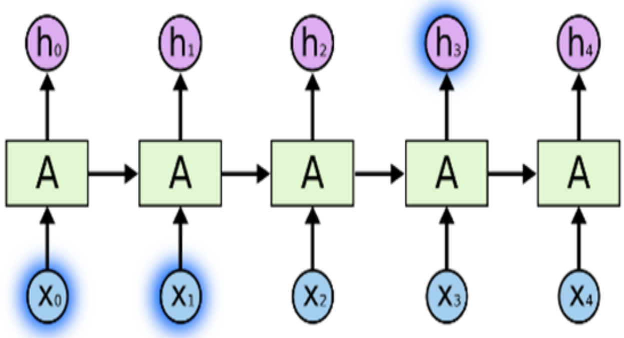

Figure 4. Example of a small gap of information in RNN

RNN: Recurrent neural networks

In certain situations, more extensive context is necessary. For instance, when predicting the last word in the text 'I study in Turkey, but I grew up in Djibouti ... I speak fluent Turkish', recent information hints that the next word might be the name of a language. However, to determine which language specifically, we require the context of 'Turkey' from earlier in the text. As the gap between the relevant information and its application grows larger, RNNs struggle to establish connections between the information, and their learning ability diminishes. Fortunately, LSTMs, a special type of RNN, do not encounter this problem because they are designed to retain information over extended periods, unlike traditional RNNs. Their architecture consists of a chain of repeating neural network modules with LSTMs featuring a more intricate structure compared to traditional RNNs.

This structure typically includes components such as gating mechanisms and memory cells. These layers include the input gate, the forget gate, the output gate, and the memory cell. Each gate and the memory cell serve specific functions within the LSTM architecture, enabling it to effectively capture and manage long-term dependencies in sequential data.

Finally, to respond to our problem, we have built a LSTM model (Figure 5) of a high-level neural networks API running on top of TensorFlow or other backend engines.

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_8 (LSTM)               (None, 50, 50)            10400

 lstm_9 (LSTM)               (None, 50, 50)            20200

 lstm_10 (LSTM)              (None, 50, 50)            20200

 lstm_11 (LSTM)              (None, 50)                20200

 dense_2 (Dense)             (None, 1)                 51

=================================================================
Total params: 71051 (277.54 KB)
Trainable params: 71051 (277.54 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 5. LSTM model parameters

LSTM: Long short-term memory

Sequential model indicates that we're creating a linear stack of layers in our neural network. Data flows sequentially from one layer to the next.

The first layer is an LSTM layer with 50 units. input_shape=[X_train.shape (1), 1] defines the shape of the input data. In this case, it expects input sequences with X_train.shape (1) time steps and 1 feature per timestep. There are three more LSTM layers with 50 units each. They are stacked on top of each other. The purpose of stacking multiple LSTM layers is to allow the model to learn more complex temporal patterns in the data by abstracting features at different levels of temporal abstraction. it only returns the output of the last timestep instead of the full sequence. This setup is common in sequence prediction tasks where we're interested in predicting a single value or the next value in the sequence.

dense layer is a standard fully connected dense layer with one unit. It's a typical setup for regression problems where the network is tasked with predicting a single continuous value. This layer provides the final output of the model.

In summary, the model architecture consists of a sequence of LSTM layers followed by a dense layer for prediction.

Auto-Regressive Integrated Moving Average

The ARIMA model, short for auto-regressive integrated moving average, is a powerful tool used for analyzing time series data (What is an ARIMA model? Taking a quick peek into ARIMA modeling | by Miranda Auhl | Towards Data Science, n.d.). It creates a mathematical equation to describe and predict patterns in our time series data. This equation has three main parts:

AR (auto-regressive): This part looks at past data points to predict future ones. It's like saying, "Today's weather is similar to yesterday's."

I (integration or differencing): This part accounts for overall trends or changes in the data over time. It's like adjusting for a gradual increase or decrease in temperature over the years.

MA (moving average): This considers errors or noise in the data based on past observations. It's like smoothing out fluctuations to see the underlying pattern.

ARIMA (p, d, q) is used to represent the ARIMA model, where each letter corresponds to a specific parameter (What is an ARIMA model? Taking a quick peek into ARIMA modeling | by Miranda Auhl | Towards Data Science, n.d.);

- **p:** This parameter determines the number of autoregressive (AR) terms in the model. Autoregressive terms are based on past observations, so p indicates how many past observations are considered.
- **d:** This parameter determines the order of differencing. Differencing is used to make the time series data stationary, which means removing trends or patterns that change over time. The order of differencing, represented by d, shows how many times you need to differ the data to achieve stationarity.
- **q:** This parameter determines the number of moving average (MA) terms in the model. Moving average terms represent the error terms or noise in the data based on past observations. So, q indicates how many past error terms are considered.

In order to determine the order of p for the autoregressive (AR) component and and q for the MA component of our model, we have plotted the autocorrelation (ACF) and partial autocorrelation function (PACF) as shown in Figure 6.

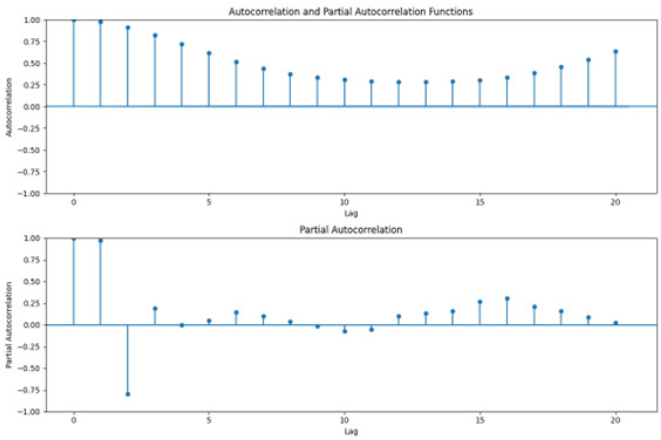

Figure 6. Autocorrelation and partial autocorrelation functions (ACF na PACF)

ACF: autocorrelation function, PACF: Partial autocorrelation function

The ACF helps identify the order of the MA component, while the PACF helps us identify the order of the AR component. Finally, to determine the order of differencing d, it involves assessing the stationarity of the time series data. We opted to compute the augmented Dickey-Fuller (ADF) test, which is a common statistical test used to determine whether a given time series is stationary or not (Bhattacharya & Burman, 2016).

The ADF test statistic is compared with critical values to determine the stationarity of the time series. We have computed the ADF statistic: -19.88829608963819 and p-value: 0.0001 since the p-value is less than the chosen significance level (usually 0.05), we conclude that the time series is stationary.

Model Design and Dataset Preparation

The process of a machine learning project typically involves several stages, from defining the problem to deploying the model. The general overview of the machine learning project lifecycle is shown in Figure 7 below. It begins with gathering relevant data for the problem. Then, it conducts data analysis and preprocessing, including standardizing the features for equal scaling and performing feature engineering and moves on to model selection and evaluation, where an appropriate algorithm is chosen to assess its performance.

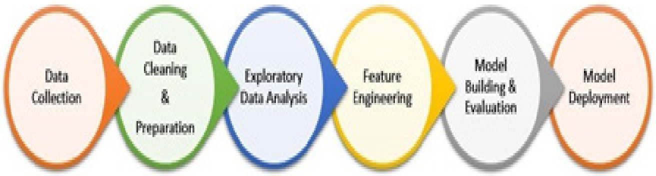

Figure 7. Steps of the proposed method

We have plotted the daily average energy consumption (**Figure 8**) and the box plot (**Figure 9**) to investigate if there are outliers that should be processed. Additionally, we have opted to standardize the values in order to improve the model training.

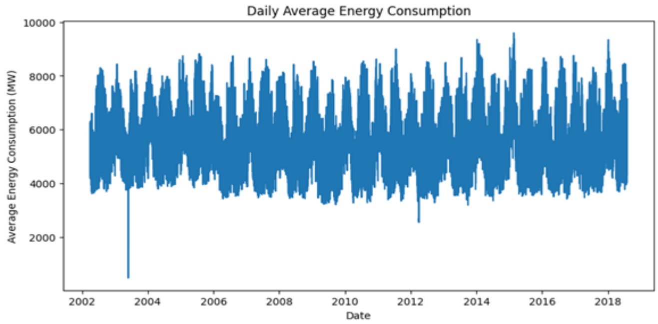

Figure 8. Daily average energy consumption

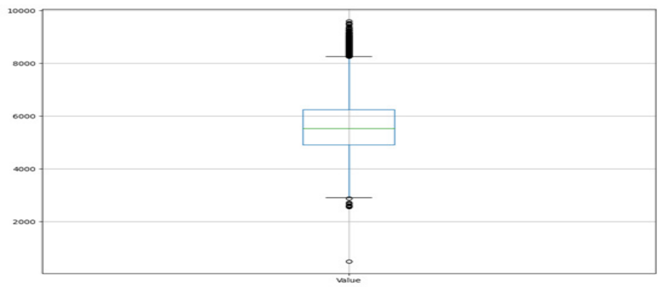

Figure 9. Box plot of the target value

The Box plot shows that there are outliers that should be preprocessed for the best performance. We split our data into training test dataset, in which the blue part as shown in **Figure 10** is the train part (from 2002 to 2018) and the green is the test part (2018).

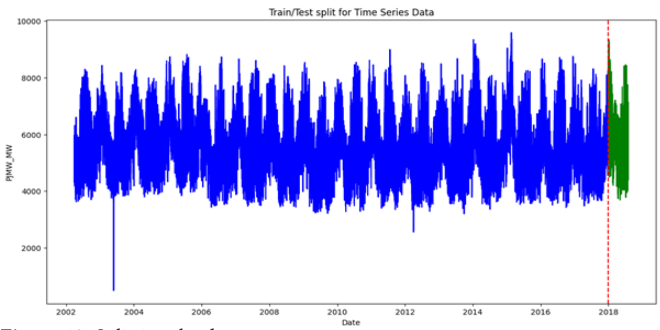

Figure 10. Splitting the dataset

In our case study, we observed that our dataset contains missing values, particularly in the 'total bedrooms' column, and numerous outliers exist in all features, including the target variable, which need to be processed. Additionally, we have created new columns based on our existing columns in order to improve model performance.

To handle the outliers in our dataset, we utilized quartiles, interquartile range (IQR), and an upper bound, all of which are statistical techniques. Firstly, quartiles divide a dataset into four equal parts: Q1, Q2, and Q3. Q1 (the first quartile) represents the 25^{th} percentile, indicating that 25% of the data falls below this value. Q2 (the second quartile) is the median, representing the middle value of the dataset when sorted in ascending order. Q3 (the third quartile) denotes the 75th percentile, meaning that 75% of the data falls below this value. Once computed, the interquartile range (IQR) is determined as the difference between Q3 and Q1: IQR=Q3-Q1. The IQR provides insight into the spread of the middle 50% of the data and is resistant to outliers, helping to compute the upper bound (Jambu, 1991).

In outlier detection, the upper bound identifies data points significantly above the typical range of values in the dataset. A common method calculates the upper bound using a multiplier (e.g., 1.5 or 3) times the IQR above Q3: upper bound=Q3+(1.5* IQR) or upper bound=Q3+(3* IQR) and lower bound=Q3-(1.5*IQR). Data points above the upper bound and below the lower bound are potential outliers and may require further investigation or treatment.

Quartiles (Q1 and Q3), the interquartile range (IQR), the upper and lower bound are calculated to identify and potentially remove outliers from all columns in a dataset. This process ensures the quality and reliability of the data for subsequent analysis or modeling tasks. We have changed the outliers by Q3, the values that are above the upper bound and Q1, the values that are below the lower bound.

As mentioned earlier, we opted to normalize the dataset using min-max scaler to enhance our model for training. Equation 1 below, depicts how min-max scaling is computed. It involves subtracting the minimum value of the dataset and then dividing by the difference between its maximum and minimum values (Wang et al., 2024).

$$\overline{X} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

The metrics used to assess the model performance include Mean_absolute_error (MAE), Mean_squared_error (MSE), R^2_score and Root_Mean_Square_Error (RMSE) (Lewinson, 2024). These evaluation metrics were employed to compare ensemble learning models.

The MSE measures the average squared difference between the actual and predicted values. It provides a comprehensive view of the model's overall accuracy. A lower MSE indicates a better fit, with smaller errors.

The MAE measures the average absolute difference between the actual and predicted values. It provides a more robust measure of error, as it is less sensitive to outliers than MSE.A lower MAE indicates better model performance, with smaller absolute errors.

The R-squared quantifies the proportion of the variance in the dependent variable that is explained by the independent variables. It assesses how well the model captures the variability in the data. Its values range from 0 to 1, where 1 indicates a perfect fit. Higher values which mean close to 1 suggest a better model fit.

The RMSE serves as a metric to gauge the average disparity between the predicted values of a statistical model and the actual values. It is calculated as the standard deviation of the residuals, which are the differences between the predicted values and the corresponding observed values. RMSE essentially indicates the extent of spread among these residuals, providing insight into how closely the observed data aligns with the predicted values derived from the model.

RESULTS

Regarding the LSTM model, we have visualized the actual and predicted energy consumption for August 2, 2018, allowing for a comparison between the predicted and true values of the data. As shown in Figure 11 below, the comparison between the predicted and actual values fits well, indicating that the model performs well.

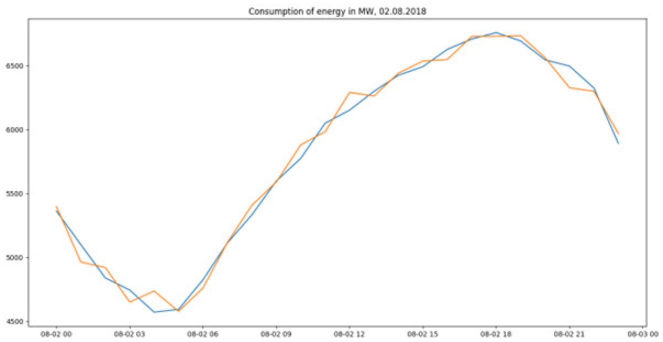

Figure 11. Comparison of the predictions and actual values for one day

The **Figure 12** below shows the actual and predicted energy consumption trends for the entire year 2018, which confirms the performance of our LSTM model.

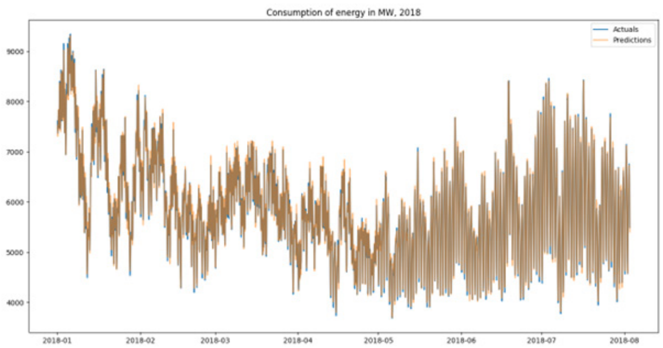

Figure 12. Comparison of the predictions and actual values for one year

The results of the metrics on the test dataset for each trained model are presented in **Table 2** below. As indicated in **Table 2**, ARIMA demonstrated good performance across all metrics: with values of 89.90, 15247.42, 123.50, and 98%, respectively. The model exhibits strong performance and indicates no overfitting, given its similarity in the metrics of the training dataset. LSTM also achieved good performance and is very close to ARIMA (**Table 3**).

Table 2. Description of PJME_hourly dataset

Number of the data	Attribute	Type
1	PJMW_MW	Float
2	Datetime	Object

PJME: Preheated Jatropha Methyl Ester, MW: Megawatts

Table 3. Metrics results of the models on test dataset

Algorithms	MAE	MSE	RMSE	R-squared
LSTM	96.06	30467.07	174.5	97.0%
ARIMA	89.90	15247.42	123.50	98.0%

LSTM: Long short-term memory, ARIMA: Autoregressive, integrated moving average, MAE: Mean_absolute_error, MSE: Mean_squared_error, RMSE: Root_Mean_Square_Error

CONCLUSION

In conclusion, this study introduced LSTM and ARIMA architecture for short-term load forecasting on the PJME hourly consumption dataset. By integrating LSTM networks, the model demonstrated the ability to capture intricate temporal relationships and adapt to exogenous influences such as weather conditions and holidays, ultimately outperforming conventional methods and several state-of-the-art approaches found in the literature. These findings underscore the potential for next-generation deep learning architectures to not only enhance prediction accuracy but also to provide more flexible and interpretable frameworks for understanding load patterns. Moreover, beyond the immediate forecasting gains, this methodology contributes to more informed decision-making in energy markets, supporting improved operational efficiency and resource allocation for grid operators. In the future, incorporating additional socioeconomic factors, exploring transfer learning to other regional datasets, and further optimizing the model's hyperparameters could drive even greater predictive accuracy and robustness. As the energy landscape continues to evolve with the integration of renewable sources and dynamic pricing schemes, adaptive, data-driven forecasting models like the one presented here will become increasingly indispensable.

ETHICAL DECLARATIONS

Referee Evaluation Process

Externally peer-reviewed.

Conflict of Interest Statement

The authors have no conflicts of interest to declare.

Financial Disclosure

The authors declared that this study has received no financial support.

Author Contributions

All of the authors declare that they have all participated in the design, execution, and analysis of the paper, and that they have approved the final version.

REFERENCES

10.PJME_hourly. (n.d.). Retrieved October 14, 2024, from https://www.kaggle.com/datasets/hajarshokrgozarr/10-pjme-hourly/data

Bhattacharya, P.K., & Burman, P. (2016). Time series. *Theory Method Statist*, 431-489. doi:10.1016/B978-0-12-802440-9.00013-8

Cai, W., Wen, X., Li, C., Shao, J., & Xu, J. (2023). Predicting the energy consumption in buildings using the optimized support vector regression model. *Energy*, 273,127188. doi:10.1016/J.ENERGY.2023.127188

Çetiner, H., & Çetiner, İ. (2021). Analysis of different regression algorithms for the estimate of energy consumption. *Eur J Sci Technol*, 31(1),23-33. doi:10.31590/ejosat.969539

Energy consumption: definition & types | StudySmarter. (n.d.). Retrieved October 14, 2024, from https://www.studysmarter.co.uk/explanations/geography/energy-security/energy-consumption/

Irfan, M., Shaf, A., Ali, T., Zafar, M., Rahman, S., Mursal, S.N.F., AlThobiani, F., Almas, M.A., Attar, H.M., & Abdussamiee, N. (2023). Multi-region electricity demand prediction with ensemble deep neural networks. *PLoS One*, 18(5),e0285456. doi:10.1371/JOURNAL.PONE.0285456

Ismael, M.N. (2022). Time series analysis of household energy using recurrent neural network. *Texas J Engineer Technol*, 8,44-52.

Jambu, M. (1991). 1-D statistical data analysis. *Explorat Multivar Data Analys*, 27-62. doi:10.1016/B978-0-08-092367-3.50007-1

Koşar, E., & Barshan, B. (2023). A new CNN-LSTM architecture for activity recognition employing wearable motion sensor data: enabling diverse feature extraction. *Engineer Appl Artific Intellig*, 124,106529. doi:10.1016/j.engappai.2023.106529

Lewinson, E. (2024). A comprehensive overview of regression evaluation metrics. *Data Sci*, 117-147. doi:10.1007/978-981-97-3245-6_9

Qin, W., Wang, L., Liu, Y., & Xu, C. (2021). Energy consumption estimation of the electric bus based on grey wolf optimization algorithm and support vector machine regression. *Sustainability 2021,* 13(9),4689. doi:10.3390/SU13094689

Son, N.N., & Van Cuong, N. (2023). Neuro-evolutionary for time series forecasting and its application in hourly energy consumption prediction. *Neural Comput Appl*, 35(29),21697-21707. doi:10.1007/S00521-023-08942-X/TABLES/6

Understanding LSTM Networks-Colah's Blog. (n.d.). Retrieved October 14, 2024, from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Wang, X., Liesaputra, V., Liu, Z., Wang, Y., & Huang, Z. (2024). An in-depth survey on deep learning-based motor imagery electroencephalogram (EEG) classification. *Artific Int Med*, 147,102738. doi:10.1016/J.ARTMED.2023.102738

What is an ARIMA model? Taking a quick peek into ARIMA modeling | by Miranda Auhl | Towards Data Science. (n.d.). Retrieved October 14, 2024, from https://towardsdatascience.com/what-is-an-arima-model-9e200f06f9eb

World Energy Consumption Statistics | Enerdata. (n.d.). Retrieved October 14, 2024, from https://yearbook.enerdata.net/total-energy/world-consumption-statistics.html

Xiong, T., Bao, Y., & Hu, Z. (2014). Interval forecasting of electricity demand: a novel bivariate EMD-based support vector regression modeling framework. *Int J Electric Power Energy Syst*, 63,353-362. doi:10.1016/J.IJEPES.2014.06.010

Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0024

Economic evaluation of solar-hydrogen hybrid system with installed power of 1 MW and 5 MW in İskenderun Organized Industrial Region

Ahmet Arslan, Battal Doğan
Department of Energy Systems Engineering, Faculty of Technology, Gazi University, Ankara, Turkiye

Cite this article as: Arslan, A., & Doğan, B. (2025). Economic evaluation of solar-hydrogen hybrid system with installed power of 1 MW and 5 MW in İskenderun Organized Industrial Region. *J Comp Electr Electron Eng Sci,* 3(1),21-28.

Received: 24.11.2024 ◆ **Accepted**: 26.01.2025 ◆ **Published**: 17.04.2025

ABSTRACT

In the study, the production potential of hydrogen production for İskenderun organized industrial zone was evaluated. Considering the geographical characteristics of the region, the potential of 30000 hours of electricity and hydrogen production per year for solar power plants that have been established or are likely to be established is examined. In the study, 4 different scenarios were selected and economic analysis was carried out. In scenarios 1 and 2, the cost of H_2 (hydrogen) was calculated by integrating a 120 kW AEM (anion electrolyte membrane) electrolyzer in addition to the 1 MW solar power plant and storing the produced hydrogen in type 1 (low pressure, metal material tank) and type 4 (high pressure composite material tank) tanks. In scenarios 3 and 4, a 1 MW AEM electrolyzer is used with the solar power plant increased to 5 MW. In these scenarios, the hydrogen produced is stored in type 1 and type 4 tanks and the H_2 cost for transportation to the end user is calculated. In addition, in 4 scenarios, different evaluations were made considering the situation with and without the solar power plant included in the system. The production cost of H_2 gas reaching the end user is more economical in the system including the solar power plant. For example, when scenario 2 is considered, while the production cost of H_2 gas is \$16.22/kg in the system without solar power plant, this value is calculated as \$10.69/kg in the hybrid system where solar power plant is included. In the 4 scenarios where solar to hydrogen hybrid system was created, the production cost of H_2 gas was determined as 58.87 \$/kg, 10.69 \$/kg, 6.65 \$/kg and 4.87 \$/kg, respectively.
Keywords: Solar-hydrogen, renewable energy, hybrid system, green hydrogen

INTRODUCTION

The energy sector is in constant evolution as a fundamental element of modern societies. The increasing demand for energy increases the importance of sustainable, environmentally friendly and renewable energy sources that aim to ensure global energy security and combat climate change (Sachdev et al., 2015). The limited reserves and high carbon emissions of fossil fuels are increasingly being replaced by renewable energy sources (Kumar et al., 2010). Legislative changes in recent years have encouraged the use of renewable energy, energy saving and efficient energy use. Renewable energy sources such as solar, wind, geothermal, hydroelectric, bioenergy and hydrogen make significant contributions to sustainable energy production by reducing environmental impacts (Sayed et al., 2023). These developments have led researchers to focus on renewable energy sources (Olabi, 2017; Bocklisch, 2015).

Among renewable energy sources, solar energy attracts attention with its ease of use and wide application possibilities in various fields (Suberu et al., 2014). Solar energy can be used in a wide range of applications from daily life to housing, from agriculture and communication to industry, from power plants to military services (Kutlu, 2002). In Dağtekin's study, (2012) it was aimed to provide the electricity required for the evaporative cooling system of a broiler house with a capacity of 20.000 with solar energy. Within the scope of this study, a 15 kWe photovoltaic (PV) power plant was planned to generate 22,935 kWh of electricity on an area of 98.01 m^2 and it was calculated that this system would reduce annual CO_2 emissions by 20.259 kg. Furthermore, the net payback period of this PV system is stated as 9.2 years. The economic analysis of the solar power plant planned to be established in Sivas Cumhuriyet University campus was carried out by Sarı and Özyiğit (2020). In this study where monocrystalline and polycrystalline modules were compared, it was concluded that monocrystalline modules provide higher efficiency, but polycrystalline modules are more economical. In a 30-year period, the energy sales revenues were calculated as \$4.140.900 for monocrystalline modules and \$4.111.100 for polycrystalline modules. Karaağaç et al. (2020) conducted a solar energy system design and cost analysis for a chicken farm. In the study, it was determined that the average monthly electricity consumption of the farm was 2.778 kWh, the

Corresponding Author: Battal Doğan, battaldogan@gazi.edu.tr

monthly cost of this consumption was 1.290 TL and the total annual cost was 15.480 TL. It was calculated that the average monthly production of the solar panel system to be installed on the roof of the farm was 2.875 kWh and the total annual production was 34.510 kWh, and it was reported that the payback period of this system was 6 years.

Electricity generated by solar panels can sometimes be insufficient to meet instantaneous energy demands (Babu & Ponnambalam, 2017). In this case, hydrogen stands out as an important alternative, as solar-generated electricity can be used to produce green hydrogen with the help of electrolyzers. This hydrogen can then be stored to meet energy demand or utilized in various sectors such as industry and industry (Wu et al., 2015). The fact that hydrogen is a clean energy carrier and produces only water vapor and energy as a result of combustion offers a solution that is compatible with carbon emission reduction targets. In this context, hydrogen integration into solar power plants can contribute to making energy systems more reliable, flexible and environmentally friendly (Ferrero & Santarelli, 2017). In the literature, there are many studies on the utilization of electricity generated from renewable energy sources in green hydrogen production. For example, Akyüz (2010) investigated the potential of hydrogen production using solar and wind energy in the Balıkesir region and economically evaluated that 20.3 kg of hydrogen can be produced annually with 1 kW wind energy and 0.56 kW electrolysis unit. Veziroğlu and Şahin (2008) compared solar to hydrogen energy system with synthetic fossil fuel system in terms of production costs, environmental impacts and utilization. Abbas et al. (2023) investigated the possibilities of solar hydrogen production in four main cities of Iraq. Using a 22.0 kWp off-grid photovoltaic system and an 8.0 kW alkaline electrolyzer, the simulations showed that the annual hydrogen production ranged from 1713.92 kg to 1891.12 kg, oxygen production ranged from 1199.74 kg to 1323.78 kg, and water consumption ranged from 7139.91 L to 7877.29 L. The production cost of hydrogen was calculated as 3.79 dollars per kilogram.

Nnabuife et al. (2024) investigate the integration of wind and solar energy in green hydrogen production. They stated that hybrid systems reduce the overall hydrogen production and infrastructure costs while increasing the use of renewable energy sources, resulting in a reduction in the balanced cost of hydrogen (LCOH) ranging from 3.5 to 8.9 Euro per kilogram.

Al-Ghussain et al. (2022) investigate the feasibility of green hydrogen production using solar, wind and hybrid PV-wind renewable energy systems (RES) in Jordan for the year 2050. The results show that the PV-based system has the highest demand-supply ratio (>99%), but the wind-based system is more economically viable with installed capacities of RES, ESS and PEM of only 23.88 GW, 2542 GWh and 20.66 GW. They also calculated the highest hydrogen annual production rate (172.1×10 tonnes) and the lowest hydrogen cost as 1.082 USD/kg.

Nasser et al. (2022) evaluated a case study in Egypt, investigating hydrogen production using solar energy, wind energy and electrolyser. As a result of their calculations, the system can produce 1.912 kg of hydrogen per year, energy efficiency is 16.42% and exergy efficiency is 12.76%. As a result of the economic analysis, LCOH is 3.73-4.656 $/kg and LCOCH (compressed variable balanced hydrogen cost) is 5.922-7.35 $/kg, and the payback period ranges from 7 to 13.85 years. At the same time, they stated that the system will provide CO_2 emission reduction up to 689.4 tonnes during its lifetime.

Liu et al. (2023) evaluated the WSH-MES system designed by combining wind, solar and hydrogen energy. According to the case study in Zhangbei, the system produces 931.39 kg of hydrogen in one year, providing 16.03% energy efficiency and 17.94% exergy efficiency. According to the calculations, it provides payback in 3 years with a net present value of $5.06 million and reduces 4.220.000 tonnes of CO_2 emissions.

The main reasons for installing the system in İskenderun OSB are presented in **Table 1**.

Table 1. Reasons for preference of IOSB

Category	Details
Strategic location and infrastructure	**Port and logistics facilities:** İskenderun Port is an important gateway to the Mediterranean Sea and provides ease of transportation of raw materials and products by sea. This offers a great advantage for hydrogen distribution. **Industrial zone infrastructure:** The region has the necessary infrastructure for heavy industry. In addition to basic services such as electricity, water, natural gas and transportation, logistics and support services required by industrial enterprises are available.
Energy and raw material access	**Renewable energy potential:** The region is suitable for renewable energy projects due to its proximity to solar and wind energy sources. This provides a significant advantage for sustainable energy use in hydrogen production. **Access to raw materials:** The raw materials (water) to be used in hydrogen production are readily available within İskenderun's existing industrial ecosystem.
Large scale industrial facilities and market potential	**Steel and chemical industry:** Green hydrogen can be produced in industrial annealing furnaces instead of purchasing H_2. **Potential customers:** Large industrial organizations operating in the region could be potential large-scale customers of hydrogen. This ensures strong demand for hydrogen.
Environmental and economic benefits	**Carbon reduction:** The use of hydrogen reduces carbon emissions by replacing fossil fuels. This plays an important role in achieving environmental sustainability goals. **Economic incentives:** Turkiye offers various incentives for renewable energy and low-carbon technologies. These incentives can make hydrogen investments more attractive.

IOSB: İskenderun Organized Industrial Zone

This study examines the potential of hybrid energy systems that can be integrated with 1 MW and 5 MW solar power plants planned to be built in İskenderun Organised Industrial Zone. The study aims to economically evaluate these hybrid systems combining hydrogen production and storage systems. Different from the literature, four different scenarios are created and the costs of green hydrogen produced from each scenario are compared. The scenarios were created by taking into account the main components of the system, namely solar power plants, anion exchange membrane (AEM) electrolysers used in hydrogen production and different types of tanks used for hydrogen storage (type 1-low pressure metal tanks and type 4-high pressure composite tanks). The research analysed how the total costs are shaped based on the initial investments and annual operating costs of these systems. Accordingly, the economic efficiency of hybrid energy systems is evaluated through the cost advantages of hydrogen production processes integrated with solar power plants.

METHODS

Photovoltaic System

İskenderun Organized Industrial Zone operates in İskenderun district of Hatay. The sunshine duration (hours) of Hatay province taken from the official page of the General Directorate of Meteorology is given in the table below (**Table 2**) (GEPA, 2024).

Solar panels are made up of a combination of photovoltaic cells. These panels are able to convert sunlight directly into electrical energy. Photovoltaic cells are made up of thin layers, usually of semiconductor materials such as silicon (Maghami et al, 2016). When sunlight hits the surface of the cells, electrons are released and the movement of these electrons creates an electric current. Solar energy is a clean and renewable resource and plays an important role in combating climate change by reducing greenhouse gas emissions (Jacobson & Delucchi, 2011). However, studies show that the efficiency of photovoltaic systems is between 15% and 20% (Maghami et al, 2016). The amount of electricity produced by solar energy is calculated using equation 1 (Arslan et al., 2024).

$$W = P.t.d \tag{1}$$

This includes electrical power (W), installed capacity (P), daily operating hours of the plant (t) and the number of days it operates in a year (d) (**Figure 1**).

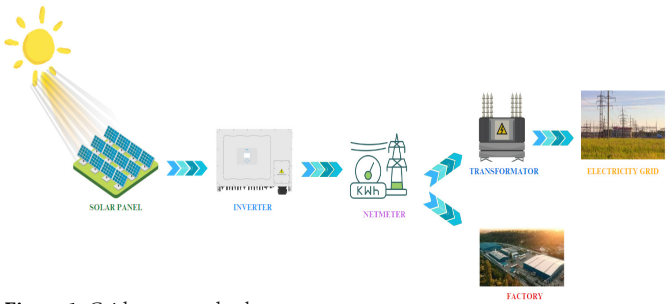

Figure 1. Grid connected solar energy system

Electrolyzer

An electrolyzer is a device designed to trigger chemical reactions using electric current from an electricity source (usually a battery or solar panel). In this process, a special substance called an electrolyte takes place inside the electrolyzer. The electric current travels between two electrodes inside the electrolyzer, the cathode and the anode. When water is electrolyzed, the water molecules are transformed into two separate gaseous forms, hydrogen and oxygen, as a result of chemical reactions, as shown in equation 2 (Garrigós et al., 2010).

$$H_2O + \text{Electricity} \rightarrow H_2 + 1/2\, O_2 \tag{2}$$

The power required for the electrolyzer (P_E) being the current density (i) and the cell voltage (V) (Nasser, & Hassan, 2023):

$$P_E = i \times V \tag{3}$$

Compressor

Compressors are mechanical devices used to convert gases from their low-pressure state to a higher pressure. These devices reduce the volume contained in the gases, making them more dense at the molecular level. Isentropic compression represents an idealized situation. Compression is assumed to take place in a frictionless and fully reversible environment. This work done by the compressor is given in equation 4. Calculation is made by taking inlet pressure (P_1), outlet pressure (P_2), inlet temperature (T_1) ve and k=1.4 (Kumar et al., 2017).

$$W_{comp} = \frac{kRT_1}{k-1}\left[\left(\frac{P_2}{P_1}\right)^{(k-1)/k} - 1\right] \tag{4}$$

The polytropic work done by the compressor (W_{comp}), can be calculated from the following equation depending on the inlet pressure (P_1), inlet temperature (T_1) and outlet pressure (P_2) with n=1.3 (Al-Mamari et al., 2023):

$$W_{comp} = \frac{nRT_1}{n-1}\left[\left(\frac{P_2}{P_1}\right)^{(n-1)/n} - 1\right] \tag{5}$$

Isothermal compression is a compression process that takes place at a constant temperature. In this case, the internal energy of the gas increases, but its entropy remains constant. In equation 6, the amount of power required for compressor operation under isothermal compression conditions can be calculated (Li et al., 2020). Here the gas constant (R), of hydrogen is the inlet pressure (P_1) nd outlet pressure (P_2).

$$W_{comp} = RT_1 \ln\frac{P_2}{P_1} \tag{6}$$

Hydrogen Storage Tank

Although hydrogen has high energy, it is a low-density gas. Therefore, it is difficult to store and transport. Among pressurized storage tanks, type 1 and type 4 tanks have notable differences in construction and performance. Type 1 tanks are manufactured with all-metal construction and the metal body carries the entire structural load. These tanks can withstand low storage pressure and have a relatively high weight. They have a gravimetric density of about 1% and cost around 83 USD/kg. Type 4 tanks, on the other hand, are high-tech and are manufactured using high-density polyethylene (HDPE) lining and full carbon fiber or carbon glass composite material with metallic protrusions. In these tanks, the load is carried entirely by the composite material and the storage pressure can be up to 1000 bar. Type 4 tanks are lighter than type 1 tanks and their gravimetric density ranges from 4.4% to 5.7%. Their cost is higher than type 1, at USD 633/kg. Hydrogen storage tanks are designed to store hydrogen under high pressure or in liquefied form. The volume of the tank is determined from the following equation depending on the tank temperature (T_{tank}), the gas constant of hydrogen (R), the mass of hydrogen stored in the tank (M_{tank}) and its pressure (P_{tank}) (Shaygan et al., 2019).

Table 2. Average insolation periods for Hatay province

Hatay	January	February	March	April	May	June	July	August	September	October	November	December	Average
Average insolation time (hours)	3.3	4.6	6	7.3	9.1	10.8	11	10.3	9.1	7	5.1	3.3	7.2

$$V_{tank} = \frac{M_{tank} T_{tank} R}{P_{tank}} \quad (7)$$

Solar Energy System

In the study, the turnkey project cost for a photovoltaic field with 1 MW and 5 MW capacity is approximately 1000 $/kWp. Table 3 below shows the price list of the detailed components of the system.

SCENARIOS FOR HYDROGEN PRODUCTION IN HYBRID SYSTEM

In this study, different scenarios for green hydrogen production using a hybrid system are considered. Each scenario is focused on optimizing critical elements of the process such as cost effectiveness and storage solutions. Using solar energy as a sustainable source, these scenarios aim to demonstrate how green hydrogen production can be realized with different configurations and equipment. In the following sub-headings, the unique configurations and operational details of each scenario are presented to determine the most suitable solution for hydrogen production (Figure 2).

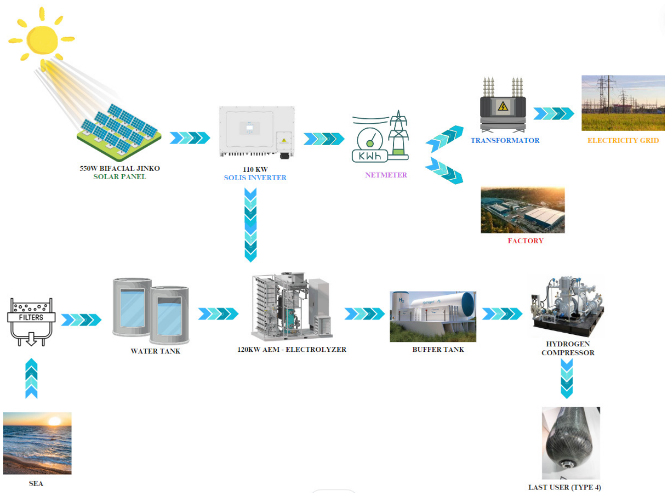

Figure 2. Workflow diagram of the hybrid system for scenario 1

The number of equipment such as electrolyzers, compressors, tanks, solar panels and inverters used in scenario 1 and the features considered in their selection are presented in Table 4.

Table 4. System elements used in scenario 1 and their properties	
Scenario 1	
Electrolyzer 1 piece	120 kW
Electrolyzer output pressure	35 Bar
H_2 compressor inlet pressure	35 Bar
H_2 compressor output pressure	350 Bar
H_2 compressor gas flow rate	25 Nm³/h-3kg/h
Buffer tank capacity 1 piece	(1Nm³=1000 L water volume=type 1 tank with 2.835 kg H_2 capacity)
Type 4 tank 180 pieces (350 Bar)	10.8 L water volume tank
Daily H_2 production (7 hours of sunbathing)	15.73 kg
Solar panel (2150 units)	550 W
Inverter (9 units)	110 kVA

Scenario 2

Scenario 2, as shown in Figure 3, all the materials used in scenario 1 are the same, but the storage tank for delivery to the end user is changed. The effect of the different type of storage tank on the H_2 cost of delivering hydrogen to the end user was evaluated. For scenario 2, the compressor power and tank specifications used in scenario 1 are changed. The rest of the system elements are the same. The compressor outlet pressure used for scenario 2 is 200 bar. This reduces the cost in scenario 2. The industrial cylinders (type 1) used to store H_2 gas have a water volume of 50 L and 66 units were used.

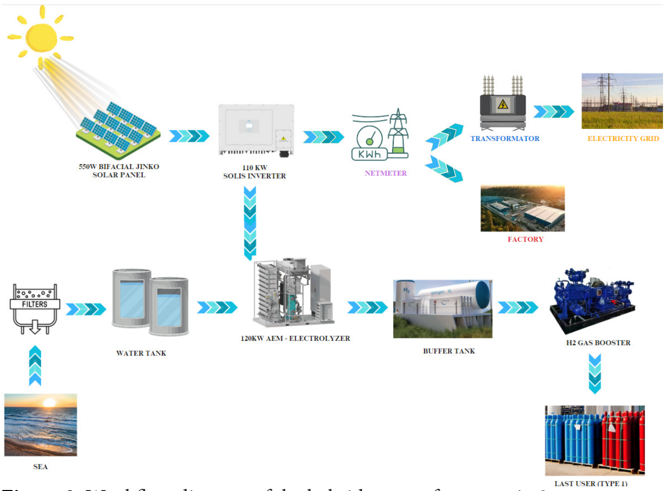

Figure 3. Workflow diagram of the hybrid system for scenario 2

Table 3. Materials and details for photovoltaic field		
1 MW components	**Materials and details**	**Price**
Photovoltaic panels	550 W Panel	$120.00/piece
Inverters	110 kW Inverter	€3.250,00/piece
Assembly and infrastructure materials	Roof mounting systems, cables, fasteners	Constriction: $16.27/kWp DC cable 1x6: 0.84 $/m
Electrical components	Distribution panels, switchgear, transformers, measurement systems	Ges panel: $20.000/piece GES scada panel: €2.200/piece transformer 4000kV: $70.000,00
Safety and protection equipment	Grounding systems, lightning protection systems, fire protection systems	Earthing 1000 €/system
Infrastructure and construction	Roof assessment, permits and licenses	$14000/system
Software and monitoring systems	-	Scada integration: scada panel included price
Auxiliary components	Maintenance and repair equipment	1500 $/system

Scenario 3

In the solar power plant with an installed capacity of 5 MW, the flow chart of which is given in Figure 4, the production of hydrogen with the help of an electrolyzer and its sale to buyers is considered as scenario 3. With the inclusion of a 1 MW AEM electrolyzer in the system, it is determined that 132 kg H_2 can be produced in a 7-hour sunshine period per day and the remaining electricity is transferred to the grid in this scenario.

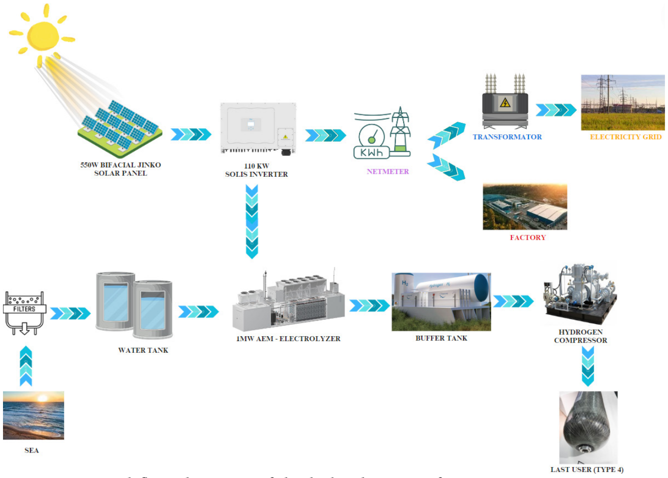

Figure 4. Workflow diagram of the hybrid system for scenario 3

For the equipment used in scenario 3 such as electrolyzer, compressor, tanks, solar panels and inverter, factors such as capacity and cost were taken into consideration. The technical specifications of each equipment were evaluated to determine the most suitable materials for the scenario (Table 5).

Scenario 4

Scenario 4 is designed by changing only the tank type by using the equipment in scenario 3 as can be seen in Figure 5. As in scenario 3, the final economic value of 132 kg of H_2 gas produced with 7 hours of sunshine time was determined by considering all costs. The materials to be used for scenario 4 are the same as in scenario 3 and only the compressor size and tank type are changed. In scenario 4, 540 type 1 storage tanks and a 200-bar compressor were selected to press H_2 gas to these tanks.

Table 5. System elements used in scenario 3 and their properties

Scenario 3	
Electrolyzer	1000 kW
Electrolyzer output pressure	35 Bar
H_2 compressor inlet pressure	35 Bar
H_2 compressor output pressure	450 Bar
H_2 compressor gas flow rate	210 Nm³/hour
Buffer tank capacity	28.5 H_2 capacity (2 500 NL tanks)
Type 4 storage tank 9 units (450 Bar-1500 L)	390 kg H_2 capacity tank
Daily H_2 production (7 hours of sunbathing)	132 kg
Solar panel (2150x5 pieces)	550 W
Inverter (45 units)	110 kVA

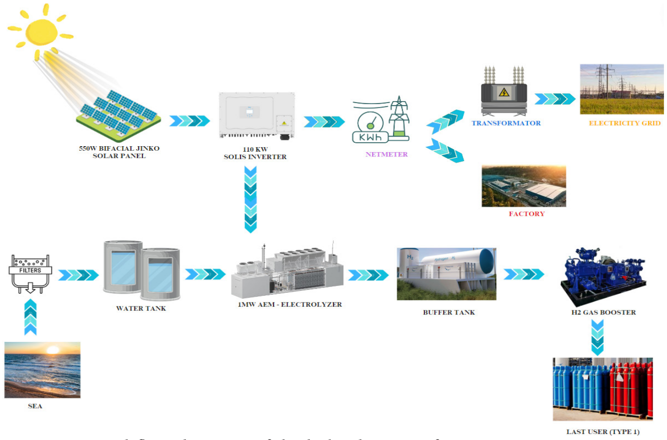

Figure 5. Workflow diagram of the hybrid system for scenario 4

RESEARCH FINDINGS

In the study prepared within the scope of the economic evaluation of the solar to hydrogen hybrid system for 1 MW and 5 MW solar power plants to be established in İskenderun organized industrial zone, the equipment suitable for hybrid power plants were selected and their prices were obtained from commercial companies. Initial investment cost, engineering procurement and construction (EPC) services and maintenance costs are included in the designed system. In Table 6, the system prepared without a solar power plant and operating 7 hours a day is evaluated. EPC services

Table 6. Costs of system elements not including solar power plan

	Scenario 1				Scenario 2		
	Equipment	Initial investment amount (CAPEX $)	Maintenance cost ($/year)		Equipment	Initial investment amount (CAPEX $)	Maintenance cost ($/year)
1	AEM electrolyzer	385.000		1	AEM electrolyzer	385.000	
3	H_2 compressor	200.000		3	H_2 compressor	100.000	
4	Buffer tank	25.000	428.000	4	Buffer tank	25.000	107.760
5	Type 4 storage tank	1.530.000		5	Type 1 storage tank	28.800	
6	EPC services	214.000		6	EPC services	53.880	
Total price ($)			2.782.000		Total price ($)		700.440
	Scenario 3				Scenario 4		
	Equipment	Initial investment amount (CAPEX $)	Maintenance cost ($/year)		Equipment	Initial investment amount (CAPEX $)	Maintenance cost ($/year)
1	AEM electrolyzer	1.500.000		1	AEM electrolyzer	1.500.000	
3	H_2 compressor	900.000		3	H_2 compressor	500.000	
4	Buffer tank	60.000	555.000	4	Buffer tank	60.000	455.876
5	Type 4 storage tank	315.000		5	Type 1 storage tank	219.375	
6	EPC services	277.500		6	EPC services	227.938	
Total price ($)			3.607.500		Total price ($)		2.963.189

AEM: Anion electrolyte membrane, EPC: Engineering procurement and construction, CAPEX: Capital expenditures and maintenance costs are calculated annually to show in the table. The cost of the equipment used in the system is presented for different scenarios. In scenarios 1 and 2, all equipment is similar and only the storage tank is different. Type 4 tank increases the cost of the system. In scenario 1 the cost of the storage tank is 1530000 $ while in scenario 2 this value is 28800 $ due to the type 1 tank.

While calculating the cost of hydrogen when it reaches the end user, EPC Services and maintenance costs are calculated based on 30000 hours of operation of the system. The production cost of H_2 gas reaching the end user for the scenarios is given in **Figure 6**. The 120 kW AEM electrolyzer used in scenarios 1 and 2 produced 67375 kg of H_2 gas in 30000 hours, while the 1 MW AEM electrolyzer used in scenarios 3 and 4 produced 566250 kg of H_2 gas. Upon 30000 hours of operation of the system without a solar power plant, the total cost in scenario 1 was calculated as 4338850 $, in scenario 2 as 1092580 $, in scenario 3 as 5626312 $ and in scenario 4 as 4621442 $. As shown in **Figure 6**, the cost of H_2 gas calculated using these values was 64.40 $/kg in scenario 1, while this value was determined as 16.22 $/kg in scenario 2. The cost of H_2 in scenarios 3 and 4 using 1 MW electrolyzer was calculated as $9.94/kg and $8.16/kg, respectively.

The economic analysis of the hydrogen production system given in **Table 6** in the study is shown in **Table 7** by adding a solar power plant to the hydrogen production system according to the operating conditions of 7 hours per day. In the hybrid system, EPC services and maintenance costs vary compared to the hydrogen production system. The amount of electricity produced from the solar power plant is calculated on the basis of 30000 hours and the electricity sales price is accepted as 0.08 $. The profit from the electricity generated is subtracted from the total cost. Type 4 tank increases the cost of the system. Due to the increase in the amount of hydrogen in the hybrid system, the cost of the storage tank in scenario 3 is $315000, while in scenario 4 this value is $219375 due to the type 1 tank.

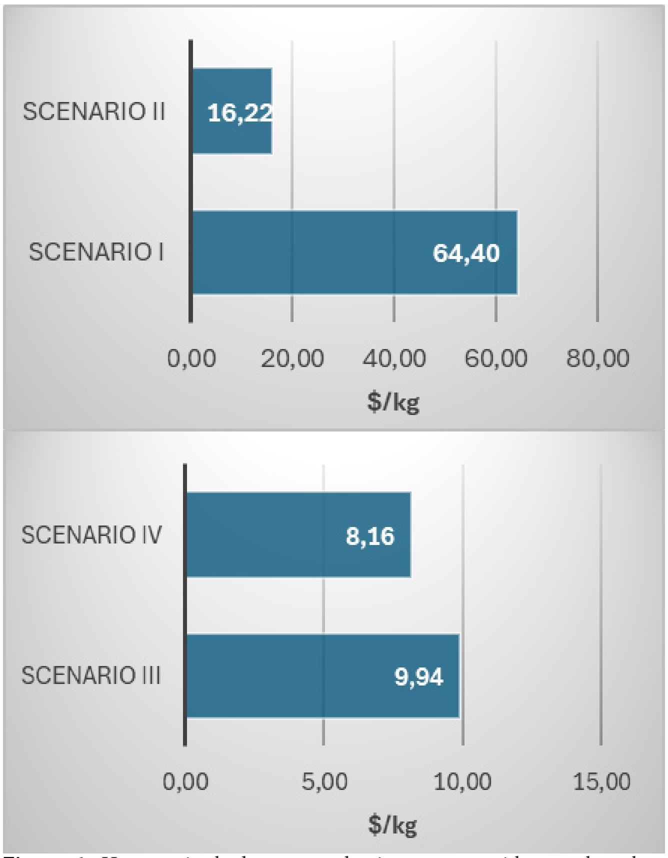

Figure 6. H_2 cost in hydrogen production system without solar plant connected

The 120 kW AEM electrolyzer used in scenarios 1 and 2 produced 67375 kg of H_2 gas in 30000 hours and the 1 MW AEM electrolyzer used in scenarios 3 and 4 produced 566250 kg of H_2 gas. The total cost in scenario 1 is calculated as 6366350 $, in scenario 2 as 3120079 $, in scenario 3 as 15763812 $ and in scenario 4 as 14758943 $. The profit from the sale of electricity was deducted from the scenarios upon the operation of the system's solar power plant for 30000 hours. Calculated using

Table 7. Costs of system elements included in the solar power plant

	Scenario 1				**Scenario 2**		
	Equipment	**Initial investment amount (CAPEX $)**	**Maintenance cost ($/year)**		**Equipment**	**Initial investment amount (CAPEX $)**	**Maintenance cost ($/year)**
1	AEM electrolyzer	385.000		1	AEM electrolyzer	385.000	
2	1 MW solar power plant	1.000.000		2	1 MW solar power plant	1.000.000	
3	H_2 compressor	200.000	628.000	3	H_2 compressor	100.000	307776
4	Buffer tank	25.000		4	Buffer tank	25.000	
5	Type 4 storage tank	1.530.000		5	Type 1 storage tank	28.800	
6	EPC services	314.000		6	EPC services	153888	
Total price ($)			4.082.000		Total price ($)		2.000.544
	Scenario 3				**Scenario 4**		
	Equipment	**Initial investment amount (CAPEX $)**	**Maintenance cost ($/year)**		**Equipment**	**Initial investment amount (CAPEX $)**	**Maintenance cost ($/year)**
1	AEM electrolyzer	1.500.000		1	AEM electrolyzer	1.500.000	
2	5 MW solar power plant	5.000.000		2	5 MW solar power plant	5.000.000	
3	H_2 compressor	900.000	1.555.000	3	H_2 Compressor	500.000	1.455.876
4	Buffer tank	60.000		4	Buffer tank	60.000	
5	Type 4 storage tank	315.000		5	Type 1 storage tank	219.375	
6	EPC services	777.500		6	EPC services	727.938	
Total price ($)			10.107.500		Total price ($)		9.463.194

AEM: Anion electrolyte membrane, EPC: Engineering procurement and construction, CAPEX: Capital expenditures

these values, the cost of H_2 gas in scenario 1 is 58.87 $/kg while in scenario 2 this value is calculated as 10.69 $/kg. The cost of H_2 in scenarios 3 and 4 using 1MW electrolyzers is calculated as $6.45/kg and $4.87/kg, respectively (Figure 7).

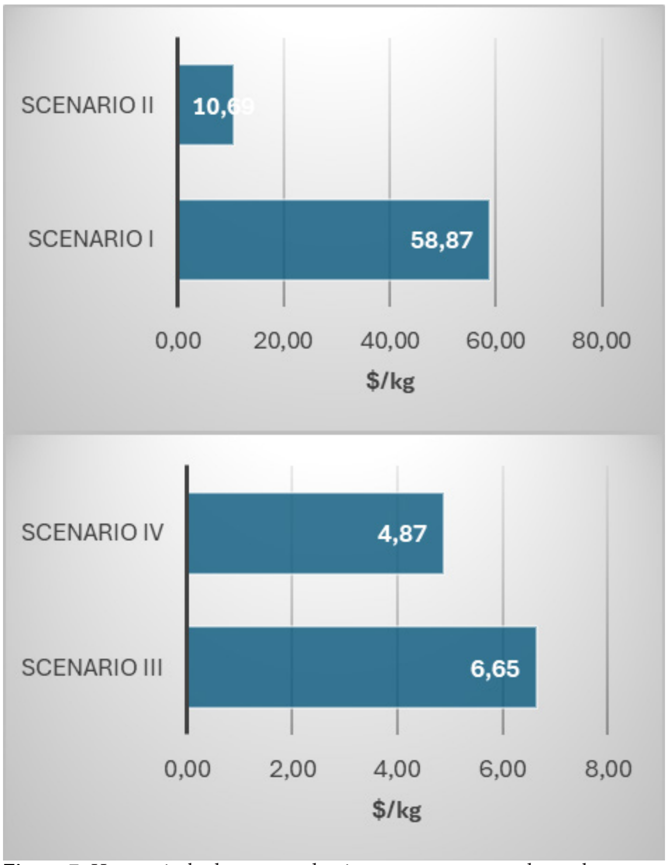

Figure 7. H_2 cost in hydrogen production system connected to solar power plant

DISCUSSION

According to the overall results of the study, solar to hydrogen hybrid systems become more economically advantageous when integrated with a solar power plant. This reveals the potential of solar energy as a renewable and sustainable energy source for hydrogen production. The establishment of such hybrid systems in regions with high solar energy potential such as İskenderun Organised Industrial Zone will provide both economic efficiency and contribute to energy independence. The correct design and optimisation of hybrid energy systems will play an important role in the energy sector in the future. Reducing hydrogen production costs becomes possible with the integration of renewable energy sources. The addition of solar power plants provides significant reductions in production costs. In the selection of storage tanks, factors such as safety and efficiency as well as cost should not be ignored. In the future, the implementation of such hybrid systems on a larger scale will create a significant transformation in the energy sector by encouraging the integration of renewable energy sources.

CONCLUSION

In this study, the economic feasibility and hydrogen production costs of solar to hydrogen hybrid systems for İskenderun Organised Industrial Zone are investigated. The evaluations made under different scenarios clearly revealed the economic advantages of hybrid systems. Hydrogen production costs are quite high in the calculations made without including the solar power plant. In a system with 1 MW solar power plant and 120 kW AEM electrolyser, the hydrogen production cost is determined as 64.40 $/kg when type 4 tank is used and 16.22 $/kg when type 1 tank is used. In the case of electricity production with solar power plant, the electricity production cost of hybrid systems decreases. In the systems using type 4 tank and type 1 tank, solar energy assisted hydrogen production costs were calculated as 58.87 $/kg and 10.69 $/kg, respectively. This clearly demonstrates the economic contribution of solar energy to the system. The costs, which are 9.94 $/kg and 8.16 $/kg without solar power plant, decrease to 6.45 $/kg and 4.87 $/kg, respectively, when solar power plant is included. This finding shows that larger scale hybrid systems are more economically advantageous. The type of storage tank was also found to have a significant impact on costs. Type 4 tanks stored hydrogen at higher costs, while type 1 tanks reduced costs.

ETHICAL DECLARATIONS

Referee Evaluation Process

Externally peer-reviewed.

Conflict of Interest Statement

The authors have no conflicts of interest to declare.

Financial Disclosure

The authors declared that this study has received no financial support.

Author Contributions

All of the authors declare that they have all participated in the design, execution, and analysis of the paper, and that they have approved the final version.

REFERENCES

Abbas, M.K., Hassan, Q., Tabar, V.S., Tohidi, S., Jaszczur, M., Abdulrahman, I.S., & Salman, H.M. (2023). Techno-economic analysis for clean hydrogen production using solar energy under varied climate conditions. *Int J Hydr Energ,* 48(8),2929-2948. doi:10.1016/j.ijhydene.2022.10.073

Akyüz, E. (2010). Hibrid yenilenebilir enerji sistemleri ile elektrik ve hidrojen üretiminin araştırılması. Doctoral dissertation, Balıkesir University, Turkiye.

Al-Ghussain, L., Ahmad, A.D., Abubaker, A.M., & Hassan, M.A. (2022). Exploring the feasibility of green hydrogen production using excess energy from a country-scale 100% solar-wind renewable energy system. *Int J Hydr Energ,* 47(51),21613-21633. doi:10.1016/j.ijhydene.2022.04.289

Al-Mamari, A., Al-Azri, N., & Al-Rawahi, N. (2023). A thermodynamic approach to the enhancement of reliability and the optimization of energy efficiency of centrifugal compressors. In Abu Dhabi International Petroleum Exhibition and Conference (p. D021S050R006). SPE.

Arslan, A., Ertürk, T., Doğan, B., & Yeşilyurt, M.K. (2024). Antalya ilindeki yenilenebilir enerji kaynaklarından yeşil hidrojen üretiminin ekonomik analizi. *Düzce Üni Bil Teknol Derg,* 12(2),932-946. doi:10.29130/dubited.1266905

Babu, C., & Ponnambalam, P. (2017). The role of thermoelectric generators in the hybrid PV/T systems: a review. *Energy Convers Manag,* 151,368-385. doi:10.1016/j.enconman.2017.08.060

Bocklisch, T. (2015). Hybrid energy storage systems for renewable energy applications. *Energy Procedia,* 73,103-111. doi:10.1016/j.egypro.2015.07.582

Dağtekin, M. (2012). Etlik piliç kümeslerinin serinletilmesinde güneş enerjisi kullanımının tekno-ekonomik analizi. *Çukurova Üni Ziraat Fak Derg,* 27(2),11-20.

Ferrero, D., & Santarelli, M. (2017). Investigation of a novel concept for hydrogen production by PEM water electrolysis integrated with multi-junction solar cells. *Energy Convers Manag,* 148,16-29. doi:10.1016/j.enconman.2017.05.059

Garrigós, A., Blanes, J.M., Carrasco, J.A., Lizán, J.L., Beneito, R., & Molina, J.A. (2010). 5 kW DC/DC converter for hydrogen generation from photovoltaic sources. *Int J Hydr Energy,* 35(12),6123-6130. doi:10.1016/j.ijhydene.2010.03.131

GEPA. 2024. https://gepa.enerji.gov.tr/, son erişim tarihi: 15.06.2024.

Jacobson, M.Z., & Delucchi, M.A. (2011). Providing all global energy with wind, water, and solar power, part I: technologies, energy resources, quantities and areas of infrastructure, and materials. *Energy Policy,* 39(3),1154-1169. doi:10.1016/j.enpol.2010.11.040

Karaağaç, M.O., Oğul, H. & Bardak, S., (2020). Kanatlı hayvan çiftliği için güneş enerji sisteminin tasarımı ve maliyet hesabı. *Düzce Üni Bil Teknol Derg,* 8(1),711-722. doi:10.29130/dubited.490154

Kumar, A., Kumar, K., Kaushik, N., Sharma, S., & Mishra, S. (2010). Renewable energy in India: current status and future potentials. *Renewab Sustainab Energy Rev,* 14(8),2434-2442.

Kumar, A., Singhania, A., Sharma, A.K., Roy, R., & Mandal, B.K. (2017). Thermodynamic analysis of gas turbine power plant. *Int J Innov Res Eng Manag,* (3),648-654.

Kutlu, S. (2002). Güneş tarlası ile elektrik enerjisi üretimi ve SDÜ kampüs alanında bir uygulama analizi. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Isparta.

Li, Y., Pei, P., Ma, Z., Ren, P., & Huang, H. (2020). Analysis of air compression, progress of compressor and control for optimal energy efficiency in proton exchange membrane fuel cell. *Renewab Sustainab Energy Rev,* 133,110304. doi:10.1016/j.rser.2020.110304

Liu, L., Zhai, R., & Hu, Y. (2023). Performance evaluation of wind-solar-hydrogen system for renewable energy generation and green hydrogen generation and storage: energy, exergy, economic, and enviroeconomic. *Energy,* 276,127386. doi:10.1016/j.energy.2023.127386

Maghami, M.R., Hizam, H., Gomes, C., Radzi, M.A., Rezadad, M.I., & Hajighorbani, S. (2016). Power loss due to soiling on solar panel: a review. *Renewab Sustainab Energy Rev,* 59,1307-1316. doi:10.1016/j.rser.2016.01.044

Nasser, M., & Hassan, H. (2023). Assessment of standalone streetlighting energy storage systems based on hydrogen of hybrid PV/electrolyzer/fuel cell/desalination and PV/batteries. *J Energy Storage,* 63,106985. doi:10.1016/j.est.2023.106985

Nasser, M., Megahed, T.F., Ookawara, S., & Hassan, H. (2022). Performance evaluation of PV panels/wind turbines hybrid system for green hydrogen generation and storage: energy, exergy, economic, and enviroeconomic. *Energy Convers Manag,* 267,115870. doi:10.1016/j.enconman.2022.115870

Nnabuife, S.G., Quainoo, K.A., Hamzat, A.K., Darko, C.K., & Agyemang, C.K. (2024). Innovative strategies for combining solar and wind energy with green hydrogen systems. *Appl Sci,* 14(21),9771. doi:10.3390/app14219771

Olabi, A.G. (2017). Renewable energy and energy storage systems. *Energy,* 136,1-6. doi:10.1016/j.energy.2017.07.054

Sachdev, H.S., Akella, A.K., & Kumar, N. (2015). Analysis and evaluation of small hydropower plants: a bibliographical survey. *Renewab Sustainab Energy Rev,* 51,1013-1022. doi:10.1016/j.rser.2015.06.065

Sarı, V., & Özyiğit, F.Y. (2020). Sivas Cumhuriyet Üniversitesi yerleşkesinde güneş enerjisi santralinin ekonomik analizi. *Dokuz Eylül Üni Müh Fak Fen Müh Derg,* 22(65),517-526. doi:10.21205/deufmd.2020226519

Sayed, E.T., Olabi, A.G., Alami, A.H., Radwan, A., Mdallal, A., Rezk, A., & Abdelkareem, M.A. (2023). Renewable energy and energy storage systems. *Energies,* 16(3),1415. doi:10.3390/en16031415

Shaygan, M., Ehyaei, M.A., Ahmadi, A., Assad, M.E.H., & Silveira, J.L. (2019). Energy, exergy, advanced exergy and economic analyses of hybrid polymer electrolyte membrane (PEM) fuel cell and photovoltaic cells to produce hydrogen and electricity. *J Cleaner Product,* 234,1082-1093. doi:10.1016/j.jclepro.2019.06.298

Suberu, M.Y., Mustafa, M.W., & Bashir, N. (2014). Energy storage systems for renewable energy power sector integration and mitigation of intermittency. *Renewab Sustainab Energy Rev,* 35,499-514. doi:10.1016/j.rser.2014.04.009

Veziroğlu, N.T., & Şahin, S. (2008). 21st Century's energy: hydrogen energy system. Clean Energy Research Institute, University of Miami, Coral Gables, FL 33124, USA, Gazi Universitesi, Teknik Eğitim Fakültesi, Teknikokullar, Ankara, Turkiye.

Wu, W., Christiana, V.I., Chen, S.A., & Hwang, J.J. (2015). Design and techno-economic optimization of a stand-alone PV (photovoltaic)/FC (fuel cell)/battery hybrid power system connected to a wastewater-to-hydrogen processor. *Energy,* 84,462-472. doi:10.1016/j.energy.2015.03.012

Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0025

Real-time laser weld point seam tracking system for robotic welding

Nuri Furkan Koçak[1], Ali Saygın[2]
[1]Department of Electrical and Electronics Engineering, Graduate School of Natural and Applied Sciences, Gazi University, Ankara, Turkiye
[2]Department of Electrical and Electronics Engineering, Faculty of Technology, Gazi University, Ankara, Turkiye

Cite this article as: Koçak, N.F., & Saygın, A. (2025). Real-time laser weld point seam tracking system for robotic welding. *J Comp Electr Electron Eng Sci,* 3(1),29-34.

Received: 10.03.2025 ◆ **Accepted**: 08.04.2025 ◆ **Published**: 17.04.2025

ABSTRACT

This paper presents a real-time vision-based laser weld seam tracking system optimized for robotic welding. The core innovation is a computer vision algorithm that processes video data to detect welding points with high precision, achieving an average absolute error of ±0.23 mm, with varying precision for different joint types (butt joint: ±0.63 mm, lap joint: ±0.04 mm, circular lap joint: ±0.02 mm). Integrated into an NVIDIA Jetson Nano, the system demonstrates robust real-time performance, processing video at 30 FPS for 720p resolution. By leveraging HSV color space analysis, morphological operations, and contour detection, the system effectively isolates and tracks laser points under dynamic conditions. Experimental results across butt, lap, and circular lap joints highlight its adaptability, with errors varying due to joint geometry and environmental factors. Comparative analysis shows superior accuracy over existing methods, such as a 0.31 mm error in laser-structured systems. By integrating directly with robotic welding tools, the system enables precise real-time adjustments, reducing human intervention and improving weld quality. This study provides a significant contribution to advancing automated welding technology.

Keywords: Robotics, seam tracking, laser welding, computer vision, real-time processing, jetson nano

INTRODUCTION

Laser welding is pivotal in modern manufacturing for its precision and efficiency. However, traditional systems often struggle to maintain accuracy under dynamic conditions, leading to inconsistencies in weld quality. To address these challenges, automated vision-based tracking systems have emerged as a promising solution, enabling real-time adjustments to enhance precision. This paper presents a novel system that integrates computer vision with robotic control, utilizing real-time image processing to accurately detect and track welding points, thereby improving the reliability and consistency of the welding process.

What sets laser welding apart is its ability to create deep and strong welds while keeping heat-affected areas to a minimum. This not only reduces distortion but also produces clean, high-quality joints, making it the go-to method for challenging applications (Sivasankaran, 2024). Its speed and seamless compatibility with automated systems make it an essential tool for improving production efficiency, significantly cutting down manufacturing times (Parchegani et al., 2023).

As the manufacturing world embraces digital transformation, laser welding is evolving with the integration of smart technologies. Real-time monitoring, powered by big data analytics, is reshaping how welding is done-optimizing processes, ensuring consistent quality, and reducing errors (Aminzadeh et al., 2023). Advanced monitoring tools now allow manufacturers to make on-the-spot adjustments, leading to superior quality control (Hsu & Salminen, 2023).

However, traditional manual and semi-automatic laser welding systems still face significant challenges. They often struggle to maintain precision in dynamic conditions, leading to inefficiencies and inconsistent results. For example, manual systems can falter when faced with sudden changes in welding paths, resulting in delays and inaccuracies (Li et al., 2022). Additionally, the variability of human operation often leads to uneven weld quality, particularly in unpredictable environments (Yu et al., 2022). These challenges, combined with the need for skilled operators, drive up costs and make scaling difficult (Yu et al., 2022).

To overcome these issues, robotic welding systems are stepping in with innovative solutions. These systems excel at adapting to variations in material shapes and environmental conditions, ensuring consistent weld quality. Real-time processing enables robots to adjust their paths dynamically, even in complex scenarios. Automated joint detection systems further boost efficiency by precisely guiding robotic arms to welding points, addressing labor shortages and enhancing productivity (Kos et al., 2019; Lee et al., 2024).

Recent advancements in deep learning have taken robotic applications to the next level (Karadeniz et al., 2024; Karadeniz & Koçak, 2024). In the study conducted in the Visual Studio Code program, the laser point was detected with an absolute error of 0.8 mm (Kocak & Saygın, 2024). The integration of laser-structured light has significantly improved welding

Corresponding Author: Nuri Furkan Koçak, nfurkan.kocak@gazi.edu.tr

accuracy, reducing errors to as low as 0.31 mm (Nguyen et al., 2024). Deep learning-based key point detection models now achieve over 80% accuracy even in complex environments (Mobaraki et al., 2024). Building on these advancements, our approach further improves welding accuracy by achieving an average absolute error of 0.23 mm, surpassing some conventional techniques while maintaining real-time processing capabilities. Algorithms such as YOLO, used for detecting welding paths, have reached near-perfect precision rates of 99.5% (Reddy et al., 2024). These breakthroughs are redefining robotic welding and setting new benchmarks for accuracy and adaptability in manufacturing environments (Umar et al., 2025).

This paper introduces an advanced system for detecting laser points during welding for various joint types, such as butt joints, lap joints, and circular lap joints, illustrated in **Figure 1**. Using computer vision techniques, the system identifies the laser's position in video data through HSV color space analysis. Noise is reduced with morphological operations like erosion and dilation, while smooth tracking is achieved by minimizing abrupt positional shifts. Built on the NVIDIA Jetson Nano platform, the system processes data quickly and accurately transmits laser coordinates to robotic welding systems. This integration enables precise torch control, reduces human intervention, and lays the groundwork for fully automated welding solutions.

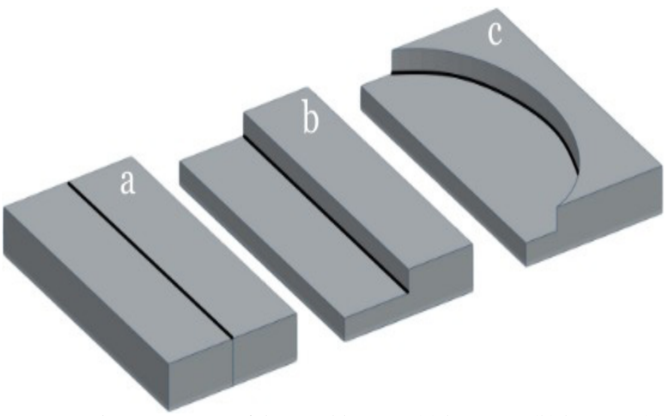

Figure 1. The comparison of three weld pieces: (a) butt joint (b) lap joint (c) circular lap joint

The proposed system has been rigorously tested and has shown significant improvements in both reliability and precision. By combining advanced detection algorithms and real-time processing, it ensures dependable performance, making it highly suitable for industrial use. These improvements streamline the welding process, reduce errors, and represent a significant step forward in automated welding technology.

This paper is organized as follows: The first section explains the algorithm development process, detailing steps like converting frames to HSV color space, reducing noise with Median Blur, creating masks, and detecting contours to locate the laser's source point. The second section focuses on real-time integration and testing, including video frame processing, tracking the laser smoothly, and overcoming challenges during implementation. Finally, the results and discussion section evaluate the system's performance in terms of accuracy and efficiency, explores noise management, and highlights potential improvements and broader applications.

METHODOLOGY

In this paper, laser light is projected to the point to be welded, and the point where the laser light is refracted is determined as the welding point. This process involves the real-time detection of source points from video images and the calculation of their coordinates. The analyses and the theoretical foundations applied in the related image processing techniques and mathematical approaches are detailed below.

Laser Light Properties and Detection

The unique properties of laser light, combined with the effectiveness of the HSV color space, have been extensively explored, particularly under varying lighting conditions. Studies highlight the benefits of the HSV model, which separates color information (hue) from intensity (value), thereby enhancing detection accuracy in challenging environments.

The HSV color space has proven effective for filtering fire-like colors, significantly improving detection in vision-based systems (Muhammad et al., 2023). Adaptive thresholding techniques, when combined with morphological processing, have shown that the HSV model is highly effective in detecting objects under dynamically changing lighting conditions (Zhao et al., 2013). Moreover, advanced approaches like QRCP decomposition applied in the HSV color space have improved image processing in low-light environments, leading to enhanced visibility and contrast, which are critical for accurate detection (Ye et al., 2023). The unique characteristics of laser light, including its high brightness and distinct color, make it easily distinguishable from other light sources. This paper utilizes the HSV color space to isolate laser light effectively, with the HSV range for red laser light defined as [172, 175, 175] to [180, 255, 255]. This configuration ensures robust detection across various lighting conditions and backgrounds.

Video Input and Color Conversion

One essential step in detecting laser light involves converting video frames from the RGB color space to HSV using OpenCV. The ability of the HSV model to separate brightness (value) from color (hue and saturation) makes it particularly effective in environments with variable lighting. This advantage has been documented in numerous studies focusing on object tracking and color recognition (Zhao et al., 2013; Cai et al., 2012).

OpenCV provides efficient tools for HSV processing, enabling real-time object tracking with accuracy rates reportedly reaching up to 90% (Gulzar et al., 2022). By applying specific thresholds to the hue channel of HSV-converted video frames, red laser light can be effectively isolated. This method has proven highly beneficial in applications like surveillance and object tracking, where it minimizes noise and enhances detection performance (Gogi & Uttarakumari, 2017).

The video was recorded using an industrial-grade Basler Ace 2 camera at a resolution of 720p and 30 FPS. Each frame, represented as $V = \{v_i \mid i \in N\}$, was first captured in the BGR color space and then converted to HSV using OpenCV's cv2.cvtColor function. A 650 nm red diode laser with a 5 mW power output was used, ensuring consistent visibility across different workpiece surfaces. The pixel values were then processed as shown in equation (1).

$$f(R, G, B) = (H, S, V) \qquad (1)$$

This transformation enabled the application of predefined HSV thresholds for isolating red tones, ensuring accurate detection of the laser light in each frame.

Noise Reduction and Masking

To address speckle noise and specular reflections, which are common in welding environments, the system employed a combination of Median Blur and adaptive thresholding. Median Blur is particularly effective at reducing salt-and-pepper noise while preserving edges, which is crucial for applications involving laser detection (Draz et al., 2023).

Recent advancements in noise reduction, including structure-oriented and space-varying median filtering, have further enhanced accuracy by focusing on specific regions of noise corruption (Oboué et al., 2024). These methods have demonstrated significant improvements in signal-to-noise ratios for real-time systems, including laser detection applications (Kumar et al., 2023).

In this paper, a Median Blur filter was applied to each frame, denoted as M (vi), as presented in equation (2).

$$M(v_i) = \text{median\_blur}\,[\text{HSV}(v_i), k = 5] \tag{2}$$

After applying Median Blur, a bilateral filter was selectively used to preserve edges while reducing high-frequency noise. This dual approach effectively mitigated artifacts caused by spatter and reflections. Subsequently, a binary mask L(vi) was created using OpenCV's cv2.inRange function, which isolates pixels within the defined HSV range for red laser light, as presented in Equation (3).

$$\mathbf{L(v_i)} = \begin{cases} \mathbf{1} & \textbf{if } [\mathbf{172, 175, 175}] \le (\mathbf{H, S, V}) \le [\mathbf{180, 255, 255}] \\ \mathbf{0} & \textbf{otherwise} \end{cases} \tag{3}$$

Morphological operations, including erosion and dilation, were subsequently applied to refine the mask. While erosion eliminates small artifacts, dilation emphasizes the laser light, enhancing its visibility against the background.

Contour Detection and Determination of Source Points

Contour analysis is a critical step for accurately tracking and detecting laser light. By applying contour extraction algorithms, laser systems can effectively determine the shape and position of detected objects (An et al., 2019). These techniques are particularly valuable in industrial applications such as laser cutting and welding, where precision is paramount (Franz et al., 2011).

Contours were extracted from the binary mask using OpenCV's cv2.findContours function. Among the identified contours {C1, C2, … , Cn}, the largest contour C_{max} was selected based on area, as presented in equation (4).

$$C_{max} = \max\,[\{\text{area}\,(C_i) \mid i \in \{1, \dots, N\}] \tag{4}$$

The source point of the laser, (cx, cy), was determined as the leftmost point within C_{max}, as presented in equation (5).

$$c_x = \min\,[\{x_j \mid (x_j, y_j) \in C_{max}\}],\ c_y = \text{corresponding } y \tag{5}$$

To ensure smooth tracking, the distance d between consecutive source points was calculated, as presented in equation (6).

$$\mathbf{d} = \sqrt{(\mathbf{c_x} - \mathbf{c_{x_{previous}}})^2 + (\mathbf{c_y} - \mathbf{c_{y_{previous}}})^2} \tag{6}$$

If d exceeded a predefined threshold of 75 pixels, the current point was reset to the previous point, preventing abrupt shifts and ensuring smoother tracking.

Visualization and Display

Real-time visualization of laser tracking is essential for various applications, from alignment tasks to UAV target tracking. OpenCV has emerged as a key tool, supporting advanced tracking systems with its comprehensive library of algorithms and deep learning integration (Yang, 2023).

Recent innovations, such as parallel multi-target detection algorithms, have improved accuracy and efficiency, achieving detection rates of up to 90% in complex environments (Cao et al., 2023). In robotic and UAV applications, laser spot detection systems have demonstrated low error rates and high execution speeds, significantly advancing practical use cases (García-Cardenas et al., 2020).

In this work, each frame's detected laser position (cx, cy) was highlighted using OpenCV's cv2.circle function for real-time visualization. Both the binary mask and the processed frame were displayed with cv2.imshow, allowing precise tracking of the laser's movement across frames.

EXPERIMENTAL STEPS

The experimental procedure in this paper comprises several stages of image processing, including identifying laser tracking points, extracting their coordinates, and marking these coordinates for visualization. The algorithm is then implemented in real-time on the NVIDIA Jetson Nano developer kit. Video input was captured using an industrial-grade Basler Ace 2 camera operating at 720p resolution (1280×720 pixels) and 30 frames per second (FPS), ensuring high-quality data acquisition under dynamic welding conditions. The camera was mounted 30 cm above the workpiece with a fixed focal length to ensure consistent imaging. A 650 nm red diode laser with a 5 mW output power was utilized to project the welding points. To optimize the reflection visibility, the laser was reflected at a 30° angle.

The initial stage involves identifying laser tracking points using computer vision techniques. Video frames are converted from the RGB color space to HSV to effectively isolate the red laser light, a process that leverages the HSV model's ability to separate brightness from color. This conversion proved particularly beneficial in environments with variable lighting, such as industrial settings with intermittent arc glare. To enhance the precision of laser detection, noise reduction techniques such as median Blur (kernel size=5) were applied. This technique effectively reduces salt-and-pepper noise while preserving edges, crucial for maintaining the integrity of laser point detection. After noise reduction, a binary mask was created to isolate pixels within the defined HSV range for red laser light (H: 172-180, S: 175-255, V: 175-255). Morphological operations, including erosion (3×3 kernel) and dilation (5×5 kernel), were then applied to refine this mask, eliminating spurious artifacts caused by welding spatter.

The next stage involves contour analysis, a critical step for accurately tracking and detecting laser light. Contour extraction algorithms were applied to the binary mask to determine the shape and position of detected objects effectively.

Experiments were conducted on three joint types-butt, lap, and circular lap joints-using 5 mm thick ST37 steel plates, with 15 samples per joint type tested under controlled conditions. The largest contour was selected based on area, and the source point of the laser was determined as the leftmost point within this contour. The distance between consecutive source points was calculated to ensure smooth tracking, with a threshold of 75 pixels applied to suppress abrupt positional shifts caused by transient noise.

In the final stage, real-time visualization of laser tracking was performed on the NVIDIA Jetson Nano developer kit. The detected weld point in each frame was highlighted, and the x, y coordinates of the welding location were determined. Both the binary mask and the processed frame were displayed, enabling precise tracking of the weld location across frames. The experimental steps followed in this study are illustrated in **Figure 2**.

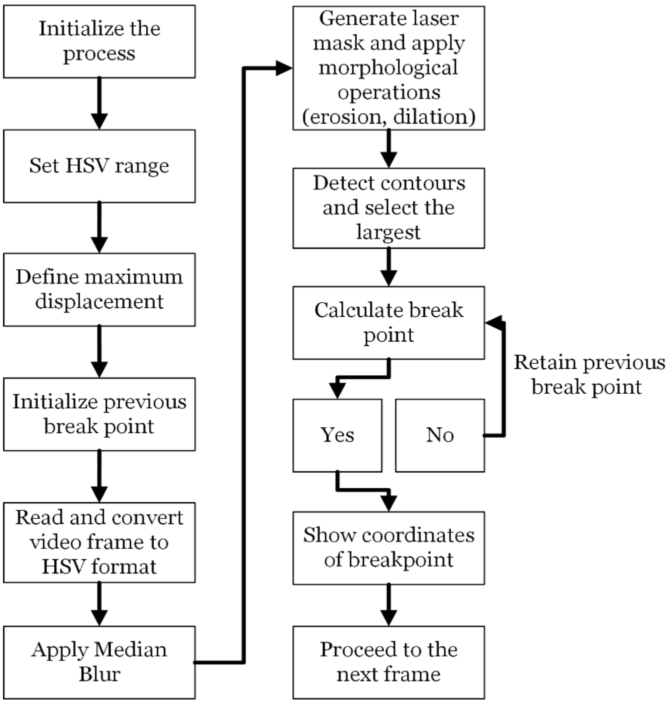

Figure 2. Flow diagram of the experimental steps

RESULTS

Through real-time video processing, the algorithm effectively isolated the laser light in complex and dynamic environments by leveraging the HSV color space and advanced noise reduction techniques. Contour analysis and morphological operations enhanced detection robustness, ensuring the laser points were accurately identified and smoothly tracked.

The proposed algorithm for detecting and tracking laser welding points using computer vision was successfully implemented and tested on the NVIDIA Jetson Nano in real-time applications. The system demonstrated high reliability, with impressive accuracy in determining laser source points. As shown in **Table**, the absolute errors varied across different joint types: a ±0.63 mm error was observed in the butt joint, ±0.04 mm in the lap joint, and ±0.02 mm in the circular lap joint. The average absolute error was calculated to be ±0.23 mm, reflecting the system's overall precision.

Table. Comparison of the models

Joint type	Average absolute error (mm)	Key observations
Butt joint	±0.63	A smaller refraction angle leads to higher error rates
Lap joint	±0.04	An overlapping structure increases the refraction angle
Circular lap joint	±0.02	A broader reflection area allows for clearer identification of the refraction point
Average	±0.23	Lap and circular lap joints exhibit significantly lower errors due to clearer reflections

In butt joints, where the two pieces are adjacent to each other, the refraction angle of the laser light is smaller, causing the laser light to scatter over a narrower area. In contrast, lap and circular lap joints, where the pieces overlap, result in a larger refraction angle, allowing for a broader reflection area and clearer identification of the laser dot. As a result, fewer errors are observed in lap and circular lap joints, while the error rate is higher in butt joints.

The seamless integration of this system with robotic welding tools enabled precise communication of welding coordinates, allowing for automated control of the welding torch. This advancement not only improved operational precision but also reduced the need for manual intervention, paving the way for more efficient and consistent welding processes.

As shown in **Figures 3-5**, the system successfully pinpointed laser locations in real-time scenarios, demonstrating its potential to enhance robotic welding technologies. In Figures 3-5, results belong to three different samples and are labeled as "a", "b", and "c".

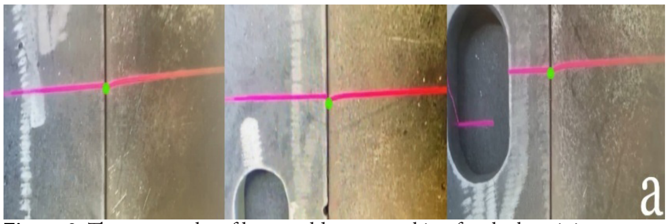

Figure 3. The test results of laser weld seam tracking for the butt joint

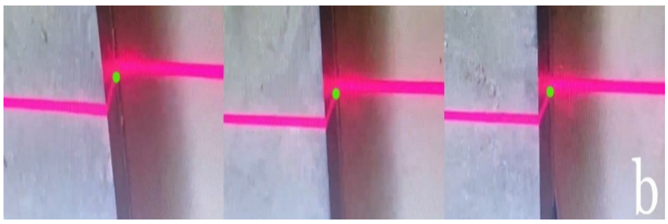

Figure 4. The test results of laser weld seam tracking for the lap joint

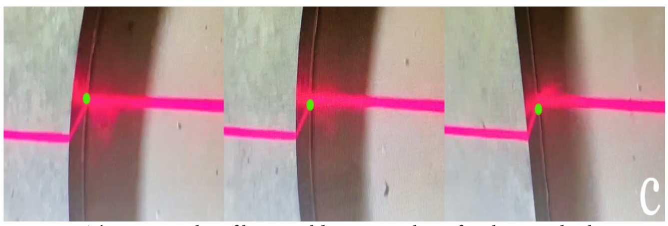

Figure 5. The test results of laser weld seam tracking for the circular lap joint

Future Work

The results of this research underscore the potential of this system to drive advances in automated welding technology with the flexibility to adapt to various industrial applications. Looking ahead, the approach will be expanded to cover a broader range of welding types, including MIG/MAG, TIG and laser spot welding, as well as applied to different types of welding parts. Although these achievements are promising, challenges remain, particularly in ensuring the system's reliability under a variety of environmental conditions and expanding its use in more complex welding scenarios. Overcoming these hurdles will be key to enabling wider adoption and further development. Furthermore, autonomous welding can be achieved by identifying the X and Y coordinates of the welding point and communicating them to the robot. The accuracy of this process can be enhanced using adaptive filtering techniques alongside customized methods for representing the robot's environment, specifically designed for its limited-scale visual perception (Karadeniz et al., 2021).

Future efforts will focus on improving the algorithm's adaptability to a range of welding conditions, including different materials and joint geometries. We also plan to explore the use of advanced machine learning techniques, such as deep learning, to enhance feature detection and prediction accuracy, as well as investigate deep learning-based enhancements to further improve tracking robustness under varying operational constraints. Additionally, testing the system in large-scale industrial settings and exploring its compatibility with other robotic systems will be critical for delivering more comprehensive automated solutions in manufacturing. By addressing these challenges and refining the integration of adaptive control strategies, this research aims to contribute to the next generation of intelligent and efficient robotic welding technologies. The combination of these advancements will not only bolster precision but also strengthen the system's real-time performance, paving the way for broader adoption in high-demand industrial environments (Cao et al., 2023).

CONCLUSION

This paper introduces a powerful, computer vision-based algorithm designed to detect and track laser welding points in real time, all integrated into an NVIDIA Jetson Nano platform. The system has shown exceptional accuracy, with a minimal error of just 0.23 in pinpointing the laser source points. By using the HSV color space and advanced image processing techniques, like morphological operations and contour analysis, the algorithm successfully isolates and tracks laser points in dynamic, real-world environments. The seamless integration with robotic welding tools allows for precise control of welding torches, minimizing manual intervention and significantly improving the efficiency and consistency of the welding process.

ETHICAL DECLARATIONS

Referee Evaluation Process

Externally peer-reviewed.

Conflict of Interest Statement

The authors have no conflicts of interest to declare.

Financial Disclosure

The authors declared that this study has received no financial support.

Author Contributions

All of the authors declare that they have all participated in the design, execution, and analysis of the paper, and that they have approved the final version.

REFERENCES

Aminzadeh, A., Rahmatabadi, D., Pahlavani, M., Moradi, M., & Lawrence, J. (2023). Smart laser welding: a strategic roadmap toward sustainable manufacturing in industry 4.0. In Sustainable Manufacturing in Industry 4.0: Pathways and Practices (pp. 41-56). Singapore: Springer Nature Singapore. doi:10.1007/978-981-19-7218-8_3

An, J., Choi, B., Kim, H., & Kim, E. (2019). A new contour-based approach to moving object detection and tracking using a low-end three-dimensional laser scanner. *Transact Vehic Technol,* 68(8),7392-7405. doi:10.1109/TVT.2019.2924268

Cai, Z.Q., Luo, W., Ren, Z.N., & Huang, H. (2012). Color recognition of video object based on HSV model. *Appl Mechan Mater,* 143,721-725. doi:10.4028/www.scientific.net/AMM.143-144.721

Cao, J., Chen, Y., Yu, D., Xu, Z., Hu, X., Liang, Y., ... & Wu, D. (2023). Real-time laser spot detection and tracking system based on parallel multi-target detection and determination algorithm. *Rev Scient Instrumen,* 94,9. doi:10.1063/5.0157141 doi:10.1016/j.jmapro.2024.03.090

Draz, H.H., Elashker, N.E., & Mahmoud, M.M. (2023). Optimized algorithms and hardware implementation of median filter for image processing. *Circ Syst Signal Process,* 42(9),5545-5558. doi:10.1007/s00034-023-02370-x

Franz, C., Singpiel, H., & Trein, J. (2011). Tracking the contour: setup assistance, machine monitoring and process control for robot and scanner applications. *Laser Technik J,* 8(5),41-44. doi:10.1002/latj.2011 90058

García-Cardenas, F., Soberón, N., Amaya, C., & Murray, V. (2020). Design of a Laser Pointer Follower Robot. In 2020 IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON), 1-4. doi:10.1109/INTERCON50315.2020.9220239

Gogi, K., & Uttarakumari, M. (2017). Detection of moving cast shadow and removal for video surveillance using HSV color space. *Int J Advanc Res Comput Commun Eng,* 6(4),733-737. doi:10.17148/IJARCCE.2017. 64138

Gulzar, M.M., Singh, R.P., & Mehra, M. (2022). HSV values and OpenCV for object tracking. *Int J Innovat Res Comput Sci Technol,* 10(1),43-48. doi:10.55524/ijircst.2022.10.1.8

Hsu, L.W., & Salminen, A. (2023). Laser welding monitoring with multisensory data fusion: a brief review. In IOP Conference Series: Materials Science and Engineering (Vol. 1296, No. 1, p. 012014). doi:10. 1088/1757-899X/1296/1/012014

Karadeniz, A.M., & Kocak, N.F. (2024). Fuzzy systems-based voltage control of buck converter for vehicles with 48V E/E architecture. *J Electric Electronic Eng,* 17(2),17-22.

Karadeniz, A.M., & Koçak, N.F. (2024). Steering angle prediction in autonomous vehicles: a deep learning approach combining VGG16 and LSTM. *J Comput Electric Electron Eng Sci,* 2(2),46-51. doi:10.51271/JCEEES-0017

Karadeniz, A.M., Ballagi, Á., & Kóczy, L.T. (2024). Transfer learning-based steering angle prediction and control with fuzzy signatures-enhanced fuzzy systems for autonomous vehicles. *Symmetry,* 16(9),1180. doi:10.3390/sym16091180

Karadeniz, A.M., Hajdu, C., Kóczy, L.T., & Ballagi, Á. (2021). Robot environment representation based on Quadtree organization of Fuzzy Signatures. In 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI), 509-514. doi:10. 1109/SACI51354.2021.9465566

Kocak, N.F. & Saygın, A. (2024). Tracking of the Laser Source Point by Image Processing. International Conference on Energy and Environmental Technologies in Engineering and Architecture (ICETEA 2024) (pp.54-57). Baku, Azerbaijan. doi:10.58225/icetea.2024.54-57

Kos, M., Arko, E., Kosler, H., & Jezeršek, M. (2019). Remote laser welding with in-line adaptive 3D seam tracking. *Int J Advan Manufact Technol,* 103,4577-4586. doi:10.1007/s00170-019-03875-z

Kumar, P.N., Budhe, S., Fathima, A.A., & Christudhas, C. (2023). VLSI implementation for noise suppression using parallel median filtering technique. In Futuristic Communication and Network Technologies: Select Proceedings of VICFCNT 2021, 1,251-259. doi:10.1007/978-981-19-8338-2_20

Lee, D., Nie, G.Y., & Han, K. (2024). Automatic and real-time joint tracking and three-dimensional scanning for a construction welding robot. *J Construct Eng Manag,* 150(3),04023165. doi:10.1061/JCEMD4.COENG-14135

Li, D., Wang, M., Wang, S., & Zhao, H. (2022). Research and development of weld tracking system based on laser vision. *Measur Control,* 55(9-10), 1124-1133. doi:10.1177/00202940221092027

Mobaraki, M., Ahani, S., Gonzalez, R., Yi, K.M., Van Heusden, K., & Dumont, G.A. (2024). Vision-based seam tracking for GMAW fillet welding based on keypoint detection deep learning model. *J Manufact Process,* 117,315-328. doi:10.1016/j.jmapro.2024.03.006

Muhammad, I., Sthevanie, F., & Ramadhani, K.N. (2023). Fire detection using combined approach of HSV-based harris corner region extraction and vision transformer classification. In 2023 International Conference on Data Science and Its Applications (ICoDSA), 117-122. doi:10.1109/ICoDSA58501.2023.10277216

Nguyen, Q.C., Hua, H.Q.B., & Pham, P.T. (2024). Development of a vision system integrated with industrial robots for online weld seam tracking. *J Manufact Process,* 119,414-424.

Oboué, Y.A.S.I., Chen, Y., Fomel, S., Zhong, W., & Chen, Y. (2024). An advanced median filter for improving the signal-to-noise ratio of seismological datasets. *Comput Geosci,* 182,105464. doi:10.1016/j.cageo.2023.105464

Parchegani, S., Piili, H., Ganvir, A., & Salminen, A. (2023). Laser welding of additively manufactured parts-A review. In IOP Conference Series: Materials Science and Engineering (Vol. 1296, No. 1, p. 012030). doi:10.1088/1757-899X/1296/1/012030

Reddy, J., Dutta, A., Mukherjee, A., & Pal, S.K. (2024). A low-cost vision-based weld-line detection and measurement technique for robotic welding. *Int J Comput Integrat Manufact,* 37(12),1538-1558. doi:10.1080/0951192X.2024.2314784

Sivasankaran, P. (2024). Laser Welding in Manufacturing Applications. Laser Assist Machin Process Appl, 17-25. doi:10.1002/9781394214655.ch2

Umar, M., Gupta, M., Verma, R., & Dhanda, N. (2025). Role of computer vision in manufacturing industry. In Machine Vision and Industrial Robotics in Manufacturing, 14-35. CRC Press. doi:10.1201/9781003438137-2

Yang, J. (2023). Real time object tracking using OpenCV. In 2023 IEEE 3rd International Conference on Data Science and Computer Application (ICDSCA), 1472-1475. doi:10.1109/ICDSCA59871.2023.10392831

Ye, M., Hu, C., Jiao, B., Liu, W., Zhang, G., & Ye, T. (2023). Low-light image enhancement using QRCP decomposition in HSV space. In 2023 8th International Conference on Signal and Image Processing (ICSIP), 313-317. doi:10.1109/ICSIP57908.2023.10270924

Yu, S., Guan, Y., Yang, Z., Liu, C., Hu, J., Hong, J., ... & Zhang, T. (2022). Multiseam tracking with a portable robotic welding system in unstructured environments. *Int J Advan Manufact Technol,* 122(3),2077-2094. doi:10.1007/s00170-022-10019-3

Zhao, H., Wu, L., Shi, Y., & Wang, Z.Z. (2013). Moving target detection method based on HSV color space. *Modern Electr Techniq,* 36(12),45-48.