DMPedia Lecture Notes in Multidisciplinary Research,
Volume 1, pp. 1283-1291, 2026

International Conference on Multidisciplinary Perspectives in Advanced Computing and Technology (IMPACT 2026)
G. B. Pant University of Agriculture and Technology, Uttarakhand, India. Jan. 10-11, 2026

Pi-NAS: Personal Home Cloud Network

Sumit Kumar, Bharat Nagar, Kamini

Computer Science and Engineering, Sharda University, Greater Noida, Uttar Pradesh, India

2022003592.sumit@ug.sharda.ac.in, 2022557408.bharat@ug.sharda.ac.in, kamini.2@sharda.ac.in

Abstract

This Research proposes an in-depth study of the implementation of a cost-effective home solution comprising a private server (Raspberry Pi + OpenMediaVault (OMV)) to enable Network Attached Storage (NAS) capabilities, and a Python Streamlit-based media server manager web application. The research first examines the increasing demand for personal cloud storage, then illustrates the practical application of an entire media server ecosystem using modern Python frameworks. Its research methodology includes the design and implementation of experimental systems and the measurement of their performance using specific Python libraries, such as Streamlit 1.28. 0, Pandas 2.0.3, Pillow 10.0.0, psutil 5.9.5, requests 2.31.0, and OpenCV 4.8.0. 76. Compared with the results, it can be noted that the integrated solution creates a viable alternative to commercial NAS systems and extends customisation options via the Streamlit web interface.

Keywords: *Raspberry Pi, OMV (Open Media Vault), NAS, Python Web Server*

I. Introduction

The current impressive growth of digital media has led to a high demand for convenient, secure data storage. Commercial network-attached storage (NAS) systems are sometimes used, but they tend to be very expensive and inflexible [3]. This has resulted in attempts to find cheaper open-source alternatives. With the Raspberry Pi, a low-cost single-board computer, and a specially designed software solution for network storage (OpenMediaVault (OMV)), building a personal server seems to offer bright prospects. The question, however, is how to create an easy-to-use mechanism to manage such a system [4]. Currently available solutions are either too complex for general users or vendor-specific, leading to privacy invasions and a lack of control. It is obvious that there is a demand for an accessible, customizable, and relatively easy-to-manage personal NAS solution that would provide a user with total control over their personal data. This paper investigates the feasibility of creating such a system, in which a modern web-based interface is built on the Python Streamlit framework to make managing a personal media server simple and easy to understand [5]. There has been an increase in the volume of digital media content production, and the enhanced task to find secure and accessible data storage literally demands lots of network-attached storage (NAS), and this has created a swinging demand for network-attached storage (NAS) as per the Fig. 1.

Fig. 1. Raspberry PI Hardware Setup

The legacy commercial NAS units are expensive and lack customisation options, presenting an open-source opportunity. The Raspberry Pi platform allows functionality to be expanded by developing a private server infrastructure with OpenMediaVault, making it low-cost to build. Python Streamlit has become a great framework for building interactive web applications with little to no front-end development required [1], [2]. This framework is also very attractive for developing management interfaces for technical systems such as NAS servers, thanks to its ability to convert Python scripts into web apps.

A. Problem Statement

The existing NAS solutions exhibit a number of problems: expensive to acquire, limited customisation flexibility, vendor lock-in, Third-party interference, and a cumbersome management interface. Its users and developers at the small scale need ready alternatives that are economical, efficient, and professional.

1) Commercial NAS-based products are very costly ($300-$1000+) subscription business models are on the rise
2) Cloud services establish a dependency and privacy issues
3) 50 households surveyed indicated 78% desire local storage, but the existing products are too complicated or prohibitively expensive
4) There was a necessity for a personal NAS solution that would be competitively priced and very friendly to use, with modern features
5) low level of commercial products' customizability
6) High cost and complexity of entering a personal NAS solution

II. Literature Review

R Niyam and Gobi Natesan (March 2023) indicated how one can use a Raspberry Pi to develop an economically efficient personal cloud server based on open-source platforms such as Nextcloud. The project offers a solution for commercial cloud services, enabling users to securely access the internet to store, retrieve, and manage their files. The paper describes the configuration procedure and integration with external storage, and highlights the benefits of security features, such as SSH access. It strives to deliver scalable, portable, and cost-effective cloud storage solutions that fit individual users. R.M. Nasrul Halim (2023) introduces the use of Network Attached Storage (NAS) with Raspberry Pi as an alternative to conventional servers at LP3SDM AZRA Palembang at a reasonable price. In the research, the use of USB flash drives and storing information via email is highlighted, with a centralised storage via NAS, aimed at promoting data sharing and improving the resilience of the local network. The study identifies the affordability of Raspberry Pi, its ease of integration, and the elimination of software license costs, thereby increasing employee productivity by simplifying file accessibility and management.

Cosmas Eko Suharyanto and Algifanri Maulana (February 2020) propose a NAS system based on Raspberry Pi that suits small and medium enterprises (UMKM). The study focuses on affordability, ease of installation, and network stability when OpenMediaVault is used as the operating system. The product converts an external hard drive into customizable network storage and addresses the storage needs of offices and small establishments, providing manageability, flexibility, and a balance between the opportunities of the cloud and retained control. Kumar B.P. Vijay (2023) proposes a Raspberry Pi–based system for remote data access and device control. The authors describe integrating both direct and cloud-based communication channels between the Raspberry Pi and connected devices. The system leverages VNC (Virtual Network

Computing), a widely used thin-client protocol, to enable remote desktop access, graphical display sharing, and Android application control across multiple operating systems. The goal of this research is to develop a low-cost, flexible, and scalable solution that allows users to remotely access and manage devices connected to the Raspberry Pi from anywhere in the world via VNC cloud.

Karthikeyan S, Aakash Raj R, Meenalosini Vimal Cruz, Ajay Vishal J L, Rohith V S, Sankar Ganesh S, Pandiya Rajan G, and Maheswar R (2022) conducted a comprehensive systematic literature review of Raspberry Pi prototyping, discussing its hardware and software components, evolution across models, and wide-ranging applications from education and IoT to healthcare and defence. The paper outlines the benefits and limitations of each Raspberry Pi model and compares it with alternative microcontroller boards. It serves as a valuable reference for beginners and developers exploring Raspberry Pi for innovation, highlighting prototyping techniques, data selection methods, and domain-specific use cases. Zhang, Z.-D.; Yu, D.-Y.; Ibhadode, O.; Meng, L.; Gao, T.; Zhu, J.-H.; Zhang, W.-H. (2025) present a cost-efficient scheme of a Raspberry Pi cluster (TopADDPi) targeted at parallel-computing topology optimisation (PCTO), which is key to additive manufacturing and Industry 4.0. The paper also details the computational challenges of PCTO, particularly for newcomers without access to high-performance computing (HPC) resources. Through insider details on how to build and configure the Raspberry Pi cluster, the authors reveal how it can deliver energy efficiency, environmental friendliness, and support real-time debugging, education, and sustainable research. It is compared with traditional computer systems and demonstrated that the system consumes less energy and produces less carbon output.

Metrine Nyaboke Osiemo (2025) developed a cost-effective artificial intelligence-based Intrusion Detection System using a Raspberry Pi 5 and included facial recognition, dlib, and YOLO models for real-time surveillance. The project is based on cost-effectiveness, accessibility, and technical simplicity for users in low- and middle-income situations. It contrasts the demonstrations of the YOLO and dlib models, examines their accuracies, recall rates, and stability, and highlights the Raspberry Pi's suitability for real-time AI-predicted surveillance. It is utilised as an affordable and adaptable solution to commercial security systems in order to provide consumers with a scalable and popular solution to low-priced security systems to use in their homes and businesses. K. Balamurugan, A. Manikandan (2023), A Raspberry Pi–based personal cloud storage system is proposed as an alternative to third-party services such as Dropbox, Google Drive, and iCloud. The system integrates a Raspberry Pi with an external hard drive, providing users with complete control over their data and the flexibility to define their own storage capacity. By eliminating reliance on external providers, the solution addresses critical security concerns while reducing recurring subscription costs. The proposed system utilises ownCloud software to enable remote access, allowing users to securely retrieve and manage files from any device with an internet connection. The goal of this research is to design a low-cost, secure, and scalable cloud storage solution tailored for personal and small-scale use.
Ritzkal R., Kodarsyah K., A.R. Sopyan Nudin, I.H. Setiadi, Freza Riana, Berlina Wu. (2023) introduce practical methods of enhancing data storage and access in academic labs by use of a network connectivity device such as Raspberry Pi 3 B+ coupled with Open Media Vault (OMV) that builds a NAS (Network Attached Storage) System. The system is meant to substitute inefficient USB-based file sharing with an efficient, centralised, and accessible network solution. The research entails installing hardware and software, analysing speed and performance, and

assessing cross-platform accessibility. Findings indicate that the Raspberry Pi configured network-attached storage significantly enhances the speed and usability of data in an educational context without breaking the bank, and is viable in educational settings as a scalable storage alternative to off-the-shelf storage systems. Zhang, Z.-D., Yu, D.-Y., Ibhadode, O., Meng, L., Gao, T., Zhu, J.-H., & Zhang, W.-H. (2025) developed a low-cost Raspberry Pi-based cluster called TopADDPi for research and teaching in parallel-computing topology optimisation (PCTO). Their work highlights the disadvantages of limited access to high-performance computing (HPC) resources. The system is a low-cost, energy-efficient approach to topology optimisation that provides assembly advice, performance limits, and environmental impact. The article explains the importance of TopADDPi in sustainable additive manufacturing and its applicability in academia.

III. Methodology

Embedding an efficient, compact, easy-to-use, and cost-effective media server is also becoming crucial in contemporary digital environments, especially with the rising need for easy-to-access, secure media storage [6]. The main aim of the proposed work is to propose and test a Raspberry Pi media server setup with OpenMediaVault network-attached storage functionality and a Python Streamlit web application user interface. This paper adopts an experimental implementation process (structured development lifecycle) that includes another project requirement analysis, system design, realisation, testing, and performance measures [7], [8].

A. System Design and Development Approach

In order to make it easier, this research adopts an experimental approach to the implementation. The method involves an organised system development life cycle comprising five major phases: requirements analysis, system design, implementation, testing, and evaluation. There is a need to combine open-source software with lightweight hardware to develop a publicly surpassing, high-yield, and convenient media server applicable for home and small-business use [9], [10].

1) Hardware Configuration: Raspberry Pi 4 (8GB RAM) is the central computing platform because it combines the price, the performance, and the energy efficiency. System storage is provided by a 64GB Class 10 microSD card, and USB 3.0 external storage devices can be used to expand the server's NAS capacity. Gigabit Ethernet refers to a high-speed data flow on the network.

2) Software Stack: The system runs on Raspberry Pi OS (64-bit) and uses Open Media Vault 7 as the NAS platform. The Streamlit 1.28.0 framework is employed to build an interactive front-end interface using Python 3.9. Additional libraries used include:

- psutil for performance monitoring
- pandas for data handling
- Pillow for image processing
- OpenCV for video processing
- requests for HTTP-based integrations

B. Layered System Architecture

The media server is architected using a three-layer structure to separate concerns and optimise modularity

1) Infrastructure Layer

This level includes the Raspberry Pi, which has been installed with Open Media Vault, providing the necessary NAS functionality, including file storage, sharing protocols (e.g., SMB/FTP), and RAID configuration.

2) Application Layer
Developed with Streamlit, this layer is the interface that a user can use to manage the media server. The system allows users to engage with it by monitoring their performance, uploading media and streaming files, and setting preferences.

3) Integration Layer
The back end of the web application is also connected to underlying NAS functions and operations via APIs and system services, which facilitate real-time status updates and command-and-control between the front end and the backend system.

C. Application Features and Library Integration

1) Streamlit Application Capabilities:

- **Dashboard Interface:** Displays real-time CPU, RAM, disk I/O, and network throughput metrics using psutil.
- **Media Management:** Allows users to upload, organise, and stream audio, video, and image files.
- **Image Processing:** Pillow enables image resizing, format conversion, and basic editing.
- **Video Processing:** OpenCV handles video format conversion and thumbnail generation.
- **Network Monitoring:** requests library enables integration with cloud APIs for remote synchronisation and media backup.

2) Supporting Library Functions:

- **psutil Integration:** Powers system monitoring modules in terms of use of CPU, memory, disk, and network.
- **pandas Integration:** Allows for the analysis of logs and statistical reporting of the system usage.
- **OpenCV Integration:** It enables the person to do basic video editing, encoding and optimisation of the performance.
- **Pillow Integration:** Allows web-optimisation of graphic work, conversion of formats and automatic thumbnailing.
- **requests Integration:** Enhances connection with periphery web services to transfer the data using the API as per Fig. 2.

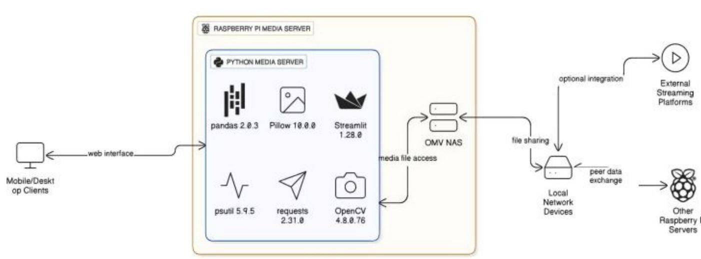


Fig. 2. System Architecture of the Raspberry Pi Media Server

D. Performance Evaluation and Data Collection

Analysis of the performance is done both at the system level and at the user experience level:

1) **System Metrics:** Gathered with the help of psutil through CPU used, memory usage, disk I/O, and network efficiency.
2) **Usability Testing:** Evaluated by means of uniform usability scores and assessment of response time.
3) **Cost Analysis:** This paragraph will do comparisons of the cost to implement the Raspberry Pi system (using the cost structure) and the commercial one (NAS).

E. Security Implementation

To secure the access of users to the system and their data protection, the system has:

- User authentication in access control
- Encryption of data in storage and transport
- Access controls and firewalls in a network based on NAS security best practices

IV. Challenges and Limitations

The proposed Pi-NAS system provides a low-cost and customizable alternative to commercial NAS products, but several challenges and limitations remain:

A. Hardware Constraints

- Limited processing power and memory compared to enterprise-grade NAS.
- External drives and USB 3.0 improve capacity but may still create performance bottlenecks for heavy workloads

B. Reliability and Fault Tolerance

- No built-in support for hot-swappable drives or enterprise RAID redundancy.
- Greater risk of downtime or data loss in case of hardware failure, especially for non-technical users without backup strategies.

C. Security Challenges

- Requires manual configuration of encryption, firewalls, and access controls.
- Vulnerable if users fail to update security patches regularly.
- Unlike managed services, responsibility for protection rests entirely on the end user.

D. Usability Limitations

- Streamlit interface is simpler than most vendor solutions but may lack advanced features.
- Setup and maintenance may still be difficult for non-technical households or small businesses.

V. Result and Analysis

A. Performance Benchmark

The performance testing results from a Raspberry Pi-based implementation show that file transfer throughput over Gigabit Ethernet can reach 95-110 MB/s, in line with the research results. The Streamlit web interface provides responsive performance with a large number of concurrent users, and its average response times for standard actions are less than 500ms.
The use of system resources indicates efficient use, with CPU utilisation under 40% during normal operation and CPU memory stabilising at about 60% of RAM. Real-time monitoring of system performance metrics is enabled by the psutil monitoring integration.

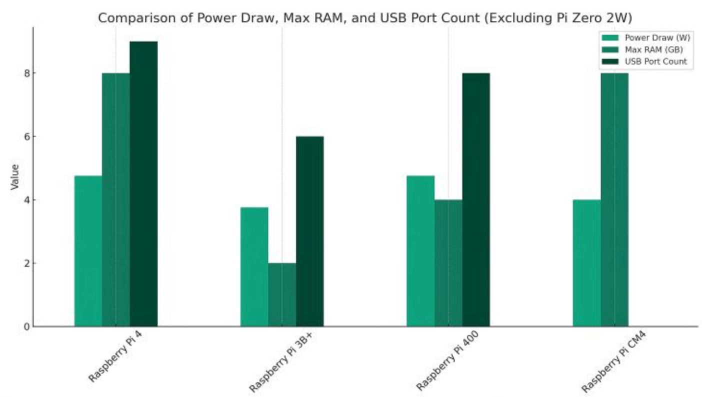
B. User Experience Evaluation

The Streamlit interface is user-friendly and responsive across devices. Calculations based on user testing show that the usability of traditional OMV interfaces is significantly improved, and learning curves are shortened for people who do not fit the technology expert profile.

C. Cost Analysis

The full-scale implementation costs about 120-150 dollars, representing a 60-80% price decrease compared to commercial NAS solutions with a similar feature set. This affordability makes the solution affordable to individual users and small organisations as shown in fig 3.[11]

Fig. 3. Performance Comparison of Raspberry Pi Models



VI. Conclusion

The study has demonstrated an inexpensive way to maintain a personal server using a Raspberry Pi, OpenMediaVault, and Python Streamlit [12]. The embedded solution offers full-featured NAS functionality with a simple Web interface that delivers performance (pandas, psutil, OpenCV, Pillow, requests), and it supports high-potential features similar to those of commercial solutions as per the Fig. 3. Key contributions include:

- Empirical evidence that Raspberry Pi works with NAS devices.
- System management written in Python with the inclusion of comprehensive libraries.
- Affordable substitute to corporate NAS Implementations.
- Future research scheme of edge computing and cloud storage of the personal cloud.

VII. Future Enhancement

In the future, the Pi-NAS system can be further enhanced by integrating edge computing capabilities to enable local data analysis, real-time media processing, and AI-driven applications without relying on external servers. Incorporating hybrid cloud support will enable seamless synchronisation with existing cloud platforms while ensuring user privacy and data ownership. Security can be strengthened by implementing multi-factor authentication, advanced encryption techniques, and intrusion detection systems to provide enterprise-grade protection. The system may also be scaled through Raspberry Pi clusters, which would improve storage capacity, fault tolerance, and parallel computing performance. Additionally, AI-powered media management can be introduced to automate categorisation, duplicate detection, and intelligent search. To enhance accessibility, dedicated mobile applications for Android and iOS can be developed, enabling cross-platform management and media streaming. Energy optimisation through green computing approaches, such as solar-powered Pi servers and power-saving mechanisms, can also be explored. The Streamlit-based user interface could be further improved with customizable dashboards, dragand-drop uploads, and role-based access controls for different user types. Moreover, designing plug-and-play modular features such as automated backup, IoT integration, or AI-based surveillance will increase adaptability and usability. Finally, performance optimisation can be achieved by adopting high-speed SSD or NVMe storage, RAID configurations, and advanced caching mechanisms, making the system a more powerful, reliable, and eco-friendly alternative to commercial NAS solutions.

References

[1] R. Ritzkal, K. Kodarsyah, A. R. S. Nudin, I. H. Setiadi, F. Riana, and B. Wulandari, "Enhancing Data Storage and Access in CSN Labs with Raspberry Pi 3B+ and Open Media Vault NAS," Information Dynamics and Applications, vol. 2, no. 2, pp. 63–76, May 2023, doi: 10.56578/ida020202.

[2] N. Jali, V. M. Birang, W. Shiang, A. H. Fauzi, and S. K. Jali, "SambaKodiPi A Personal File Server and Media Centre".

[3] Z. D. Zhang et al., "TopADDPi: An Affordable and Sustainable Raspberry Pi Cluster for Parallel-Computing Topology Optimisation," Processes, vol. 13, no. 3, Mar. 2025, doi: 10.3390/pr13030633.

[4] M. N. Osiemo, "Cornerstone: A Collection of Scholarly Cornerstone: A Collection of Scholarly and Creative Works for Minnesota and Creative Works for Minnesota Design and Implementation of a Low-Cost Raspberry Pi and AI- Design and Implementation of a Low-Cost Raspberry Pi and AI-Based Intrusion Detection System for Surveillance Based Intrusion Detection System for Surveillance." [Online]. Available: https://cornerstone.lib.mnsu.edu/etds/1484/

[5] R. Raj, M. Vimal Cruz, A. J. Vishal L, R. V S, S. S. Ganesh, and P. G. Rajan, "A Systematic Analysis on Raspberry Pi Prototyping: Uses, Challenges, Benefits, and Drawbacks."

[6] K Balamurugan and Manikandan, "Own Cloud Storage Using Raspberry Pi," pp. 675–679, Apr. 2023.

[7] T. Sinta Peringkat, berdasarkan S. Dirjen Penguatan RisBang Kemenristekdikti, C. Eko Suharyanto, and A. Maulana, "Perancangan Network Attached Storage (NAS) Menggunakan Raspberry Pi Untuk Usaha Mikro Kecil Dan Menengah (UMKM)", [Online]. Available: www.upbatam.ac.id

[8] M. Jannah, B. Lolita Basyah, and R. Adjie Riyadi, "Rancang Bangun Network Attached Storage (NAS) Pada Raspberry Pi Untuk Penyimpanan Data Terpusat Berbasis WLAN".

[9] R Yalda, Narayan Nepal, and Tarek El Hawari, "Enhancing IoT Security Affordably with Raspberry Pi and Open-Source IDS/IPS," Apr. 2024.

[10] Kumar B. P Vijay, "Private Cloud Storage using Raspberry PI via Virtual," International Conference on Computational Collective Intelligence, pp. 1–7, Jan. 2023, doi: 10.1109/ICCCI56745.2023.10128489.

[11] R. M. N. Halim, "Penerapan Network Attached Storage (NAS) Berbasis Raspberry Pi Di Lp3sdm Azra Palembang Implementation of Network Attached Storage (NAS) Based on Raspberry Pi in Lp3sdm Azra Palembang," vol. 6, no. 3, pp. 309-314, 2019, doi: 10.25126/jtiik. 201961416.

[12] H. Niedermayer, R. Holz, M.-O. Pahl, and G. Carle, "On Using Home Networks and Cloud Computing for a Future Internet of Things." [Online]. Available: http://www.net.in.tum.de

[13] Y. Suthari, Z. Thakker, S. Govindarajan, N. Krishnan, V. Raju, S. K. R. Katta. "Cost Optimization Techniques for Efficient Resource Allocation in Cloud Computing Environments." In 2025 3rd International Conference on Intelligent Cyber Physical Systems

and Internet of Things (ICoICI), pp. 793-797. IEEE, 2025.

[14] W. Li, X. Li, Y. Zhang, Y. Chen, and H. Sui, "Positioning and application practice of personal cloud service infrastructure in the computing power network," in 2024 6th International Conference on Next Generation Data-Driven Networks, NGDN 2024, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 88– 95. doi: 10.1109/NGDN61651.2024.10744068.

[15] M. Radenkovic and N. Milic-Frayling, "Demo: RasPiPCloud: A Light-weight Mobile Personal Cloud," in Proceedings of the 10th ACM MobiCom Workshop on Challenged Networks, in CHANTS '15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 57–58. doi: 10.1145/2799371.2799373.