technologies

MDPI

Article

Optimization of Energy Consumption in Voice Assistants Through AI-Enabled Cache Implementation: Development and Evaluation of a Metric

Alber Oswaldo Montoya Benitez [1,2,*], Álvaro Suárez Sarmiento [2], Elsa María Macías López [2] and Jorge Herrera-Ramirez [3,*]

[1] Faculty of Engineering, Instituto Tecnológico Metropolitano, Medellín 050013, Colombia
[2] Grupo de Arquitectura y Concurrencia (GAC), University Institute of Cybernetics, Business, and Society, Universidad de Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain; alvaro.suarez@ulpgc.es (Á.S.S.); elsa.macias@ulpgc.es (E.M.M.L.)
[3] Faculty of Exact and Applied Sciences, Instituto Tecnológico Metropolitano, Medellín 050013, Colombia
* Correspondence: albermontoya@itm.edu.co or alber.montoya101@alu.ulpgc.es (A.O.M.B.); jorgeherrerar@itm.edu.co (J.H.-R.)

Abstract: Intelligent systems developed under the Internet of Things (IoT) paradigm offer solutions for various social and productive scenarios. Voice assistants (VAs), as part of IoT-based systems, facilitate task execution in a simple and automated manner, from entertainment to critical activities. Lithium batteries often power these devices. However, their energy consumption can be high due to the need to remain in continuous listening mode and the time it takes to search for and deliver responses from the Internet. This work proposes the implementation of a VA through Artificial Intelligence (AI) training and using cache memory to minimize response time and reduce energy consumption. First, the difference in energy consumption between VAs in active and passive states is experimentally verified. Subsequently, a communication architecture and a model representing the behavior of VAs are presented, from which a metric is developed to evaluate the energy consumption of these devices. The cache-enabled prototype shows a reduction in response time and energy expenditure (comparing the results of cloud-based VA and cache-based VA), several times lower according to the developed metric, demonstrating the effectiveness of the proposed system. This development could be a viable solution for areas with limited power sources, low coverage, and mobility situations that affect internet connectivity.

Keywords: voice assistant; metric; artificial intelligence; energy saving; cache; intelligent systems

Academic Editors: Georgios Fotis, Spyridon Nikolaidis and Valeri Mladenov

Received: 9 October 2024
Revised: 16 December 2024
Accepted: 27 December 2024
Published: 2 January 2025

Citation: Montoya Benitez, A.O.; Suárez Sarmiento, Á.; López, E.M.M.; Herrera-Ramirez, J. Optimization of Energy Consumption in Voice Assistants Through AI-Enabled Cache Implementation: Development and Evaluation of a Metric. *Technologies* **2025**, *13*, 19. https://doi.org/10.3390/technologies13010019

1. Introduction

The Internet of Things (IoT) presents a model that moves towards a world integrating the physical with the virtual through the programming and interconnection of objects to solve various problems [1]. Several technologies, such as fog computing, Wireless Sensor Networks (WSN), data mining, context awareness, real-time analytics, virtual reality, and augmented reality, are integrated into IoT, offering solutions across various sectors in industry and at home. Central to these developments is the concept of Artificial Intelligence of Things (AIoT), which enables the processing of large volumes of data generated by IoT systems, enhancing and optimizing these environments [2].

To implement an IoT solution, a control system that interconnects sensors and actuators is required [3]. These interconnections can be established through various options, ranging from wired solutions to low-power wireless technologies such as Near Field Communications (NFC) [4], Zigbee, Bluetooth, or WiFi.

Among the most common devices that serve as sensors and actuators in this ecosystem are Voice Assistants (VAs). These devices, equipped with microphones, speakers, and software capable of receiving audio commands, interpreting them, searching for a response, or executing an action, are integral to user control and interaction systems within IoT. VAs take advantage of the complementarity of all these systems, enabling AI-mediated emotional perception of users [5], making them a focal point for new technological advances and developments in device and network domains [6].

In the past decade, AI integration into mobile devices has begun [7] and, since the end of 2022, there has been a proliferation of new AI-based developments and applications, leading people to rely on information technologies for their daily activities increasingly. One of the main ways to interact with these systems is through VAs. As a result, VAs have become ubiquitous, integrated into mobile devices, computers, and standalone VA devices, making their operation a significant factor in energy resource consumption, whether the VA represents an application using device hardware or functions as an independent hardware device. In either case, the operation of VAs depends on constant Internet connectivity, leading to significant energy consumption due to their passive or active listening states and the processes required to connect to the Internet to process and respond to requests. Adopting new AI-based developments and applications, which have intensified VA usage, makes their energy consumption and response times critical factors.

Conventional VAs rely on an Internet connection to function, which can be problematic in situations or geographical areas where communication is affected by low coverage or unstable connections due to mobility or high network traffic congestion. In these cases, immediate access to information would be highly beneficial, which could be achieved using cache memory. Moreover, repetitive queries to VAs about data that do not frequently change, such as user information, locations, values, and prices, often occur. For instance, when a user asks a VA about a country's territorial size, the VA must access the Internet every time the user requests this information, unnecessarily increasing resource consumption.

This research analyzes the parameters influencing VAs' energy consumption and response times in this context. A VA prototype incorporating cache memory supported by artificial intelligence is proposed to improve energy savings, response times, user experience (UX) [8], and operability in environments with limited or unstable connectivity. Additionally, this solution could reduce data exposure to numerous security threats in networks [9–11], by minimizing the constant transmission of information over external networks.

This proposal can expand the use of VAs to a wide range of remote spaces that require greater operational autonomy or the possibility of offline operation, which could be highly useful in the situations above and for low-income populations with limited Internet access. Reducing the response time for queries made through VAs could significantly impact critical processes and user experience (UX) improvement. In the case of mass adoption, there could be a reduction in the eco-environmental impact, as global annual energy consumption could decrease in theory by the order of Megawatts (MW) or Gigawatts (GW).

In the proposed implementation, several aspects of VA operation need to be considered. An initial aspect is that these require the process of speech-to-text translation using the STT (Speech To Text) protocol. This process can take place either within the local device or on servers located in the cloud provided by the Internet Service Provider (ISP). Thus, traffic in these VAs is captured as plain text in order to identify the queries issued and

the potential storage in the cache within the prototype, allowing subsequent responses to be served from the cache or encapsulated and sent through the channel to the cloud to obtain the response from that point. This alternative of using cache storage has been widely suggested due to its ability to relieve traffic load with the ISP and minimize access latency, as seen in other increasingly popular applications like live peer-to-peer (P2P) streaming or machine-to-machine (M2M) communications [12].

Kurz (2022) mentions that text processing by VAs can present error rates [8]. For this reason, in this study, a parallel process is proposed when capturing traffic in text format, consisting of sending the query to an AI scheme such as GPT-4 via an API, enabling the interpretation of content using natural language processing (NLP) techniques. The results obtained from this source would be used to feed a cache database complementarily. On the other hand, this type of development is implemented using lightweight and low-cost hardware, leading to the use of light techniques and protocols that allow for energy consumption optimization. In the state of the art, four lines of research on AVs were highlighted, which are presented in Figure 1.

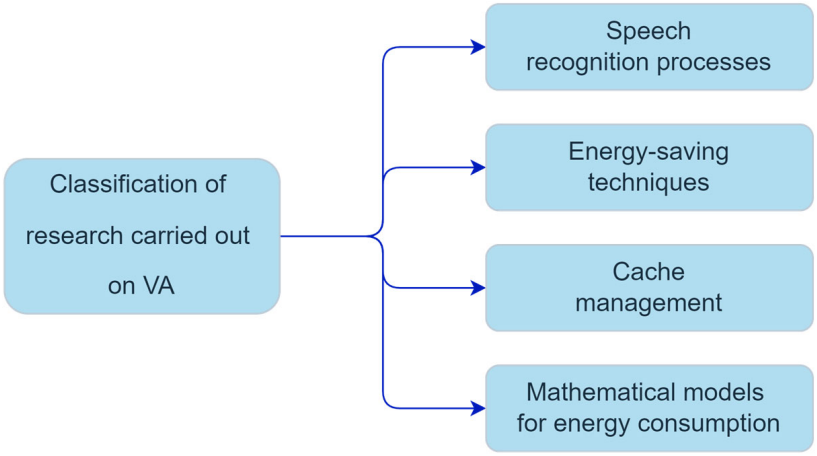

Figure 1. Classification of lines of research on VAs.

Regarding speech recognition processes, ref. [13] presents three tools for Automatic Speech Recognition (ASR) and text translation locally, avoiding the need for an Internet connection. These tools are implemented on Raspberry Pi (Raspberry Pi Foundation, UK) using Python, including validation with a word error rate (WER). The results show Vosk as one of the most reliable applications, which is why it is the tool used in this work. Additionally, Giraldo's work presents a complete mixed-signal chip system capable of directly interacting with an analog microphone, performing keyword detection (KWS) and Speaker Verification (SV) through algorithmic optimization and integrated single-chip design [14]. Although this proposal is interesting, it is not oriented toward cache storage.

Jiawen Li et al. designed and implemented an AIoT system to assist visually impaired individuals, which consists of a smart cane for monitoring various variables and glasses for text and object recognition. This system provides substantial help to people with this disability as it transmits captured information through a VA [15]. While this work bears some relation to the proposal presented here, it is not aimed at optimizing energy consumption, given the number of sensors and elements needed for its operation. A couple of communication platforms created for home automation systems are presented in [16,17], which can operate offline and automatically perform various tasks in a household. However, these solutions need certain functionalities as they do not incorporate cache storage. Furthermore, ref. [18] proposes a communication interface using ASR, applied to a

local network without Internet access. This research differs from ours as it is focused on automating home processes rather than responding to user queries.

Regarding energy-saving techniques, the focus is on optimizing the operational algorithms of devices and the routing they use for traffic transmission. In this regard, Wireless Multimedia Networks (WMN) have been developed due to the growing availability of low-cost hardware and the wide range of wireless device applications. Energy resources are limited in battery-powered wireless devices, leading to significant threats to the Quality of Service (QoS) for WMN. For these reasons, energy-efficient routing techniques are necessary to manage WMN's dynamic topology, which relies on a vital resource like energy [19] or the incorporation of media access control protocols as proposed in [20]. While minimizing and reducing hardware components contributes to energy savings, additional solutions can be implemented, especially for systems where changes to the equipment are not possible but can be made at the software level, as proposed in this study. Interesting contributions regarding minimizing energy consumption in smart spaces are made in studies such as [21–23], although these are not directly aimed at VAs.

Zhu and Luo propose a content caching method based on file value and corresponding algorithms. This method uses Markov Chains to predict a mobile user's trajectory during the next storage time interval and cache replacement. The algorithms are designed to minimize content access latency [24]. This proposal aligns with the present research; however, our project focuses on optimizing consumption by minimizing connection times. In [25], "Neurosurgeon" was developed, a scheduler that automatically divides Deep Neural Network (DNN) computation between mobile devices and cloud data centers by characterizing the type of traffic to define where processing is more efficient (device or cloud). This scheduler adapts to various DNN architectures, intelligently dividing the computation to achieve the best latency or mobile energy usage. It demonstrated an average improvement in latency of 3.1×, a reduction in mobile energy consumption of 59.5%, and a 1.5× improvement in data center performance.

In [26], a new static analysis approach, "Energy-Aware Timing Analysis of Intermittent Programs (ETAP)", is presented, estimating the behavior of intermittently used programs affected by factors such as environmental energy, the energy consumption of target hardware, capacitor size, input space, and program structure. Considering the effects of power outages, ETAP symbolically executes a given program to generate time probability distributions for each function's execution, estimating consumption but not proposing solutions for energy savings. In [27], an Efficient Supervised Machine Learning Network for Non-Intrusive Load Monitoring (ENSML) is presented for smart energy consumption measurement of devices in a household. With 99.9% accuracy, this technique would be very suitable for determining actual consumption values associated with VAs, allowing comparisons, and determining the amount of energy saved using the cache memory proposed in this work.

Regarding cache management, ref. [28] presents a decentralized service migration algorithm that integrates the analysis and selection of service placement decisions and edge base stations to minimize service migration overhead. The need for dynamic content updates and database size is considered in a framework for implementing caching in the front-end and back-end. Due to high resource consumption, this would not be the best solution for VAs with limited memory and processing resources. In [29,30], proposals by Lanyu Xu are highlighted with his CHA caching framework, which offers faster action execution in homes with IoT systems and improves resource efficiency, and the memory reduction solution presented by Mikhail, which uses a Large Language Model (LLM). However, it neither caches user queries nor provides an energy consumption metric.

Latency of Voice (Vo) communication over Internet Protocol (IP) must be considered for analyzing the performance of VAs in the Cloud. In [31], the authors presented a theoretical review of the implementation of VoIP analyzing Session Initiation Protocol (SIP) proxies as well as CODECs and presented the usual values of latency and jitter. In [32], the authors presented a position paper that also reviewed the importance of CODECs using Digital Signal Processing (DSP) and proxies but did not present any experimental results. The authors of [33] presented the performance of video streaming in a Data Center by analyzing (a) Traffic received at the destination node, (b) Traffic sent from the destination node, and (c) Packet Point to Point Delay. All the above papers recognize that latency is an important issue in delivering voice in IP, but none of them presented experimental results in which they observed the influence of proxies (caches) in energy saving as we did. Moreover, we did not consider the influence of CODEC on energy saving in our experimental results.

Another commonly used technique is Acoustic Echo Cancellation (AEC), which eliminates echo in the voice signal [34]. The Echo can be caused by transmitting the signal through different channels, such as a phone line or data network. In general, Noise Reduction (NR) is applied to eliminate such effects in the voice signal, resulting in lower energy consumption to process a clean signal. To implement this type of proxy, audio and video tools such as Squid, Varnish, or Nginx are used, which allow audio and video streams to be cached. However, it is essential to note that implementing these applications may require greater processing power in the server hosting the proxy. While the solution in this study shares similarities with the presented applications, it is carried out in the local memory of VAs, handling plain text data and adding more speed to the process.

Alternatively, caching could be performed using a cloud service provider, such as AWS Storage Gateway Volume in cache mode with SQL Server [35]. This allows recently accessed data or a full-volume copy to be stored locally in cache memory so that applications can benefit from quick access to the data. This solution requires a cloud service for deployment, which could become complex and robust, making it less attractive to VA users described in this work.

Regarding mathematical models for energy consumption, several metrics exist to measure energy consumption in different systems. In [36], a systematic literature review found 41 energy consumption metrics, which are reliable, although they are purely oriented toward software systems, not considering device hardware. Some metrics relate to standby energy consumption, energy consumption during active use, and usage time, which measures how long the device is used during a given period; another is "energy efficiency", which calculates how much energy is consumed per unit of performance or task completed. Geoffry and Kelvin compare three modeling techniques for electrical consumption: linear regression, neural networks, and decision trees [37]. These models are predictive, a characteristic not represented in this work.

The model presented in this study does not correspond to predictive models like those mentioned in the previous paragraph, nor does it depend solely on the software running on the devices; the model and metrics presented in this work derive primarily from an experimental process supported by a series of functional tests, where measurements and calculations are performed based on temporal variables from each phase of the process, ultimately reflecting a value for VA energy consumption.

It is important to note that the precision of these models may vary depending on training data, the complexity of the usage scenario, the device manufacturer, and the quality of the wireless network. Additionally, the precise measurement of VA consumption can be a challenge due to the complexity of the technology and the need to consider multiple factors in the energy calculation. On the other hand, a combination of different models may be required to provide accurate energy consumption predictions in varied situations.

Table 1 abstracts the characteristics of the studies related to this work, identifying those that explored each research line associated with this project.

Table 1. Comparison of topics addressed in related research studies.

Paper (Cite)	Energy Saving	VA/Speech	Cache Management	Metric Development
Setiawan et al., 2022 [13]		√		
Giraldo et al., 2020 [14]	√	√		
J. Errobidart., 2017 [16]		√		
Irugalbandara et al., 2023 [17]	√	√		
S K Satyanarayana, L Nirmala D. . ., 2022 [19]	√			
Li et al., 2021 [22]	√		√	
Kang et al., 2017 [25]	√			√
Erata et al., 2023 [26]	√			√
Hadi et al., 2022 [27]	√	√		√
Abolhassani et al., 2022 [28]			√	√
Ergasheva et al., 2020 [36]	√			√
Tso et al., 2007 [37]	√			√
Xu et al., 2020 [29]	√	√	√	
Rovnyagin et al., 2024 [30]	√	√	√	
This work	√	√	√	√

Thus, the main contributions of this work are as follows:

- Presentation of a connection architecture and a prototype that enables energy savings in VAs through the implementation of cache memory supported by AI, achieving experimental results, demonstrating savings and an additional level of security;
- Development of a metric to determine the energy consumption of VAs when user responses are obtained from local network cache memory;
- Comparison of consumption between passive and active states for two types of VAs;
- Proposal of a mathematical model of the process performed by VAs, from which a metric was developed to compare VA energy consumption;
- Energy-saving solution for using VAs in remote locations with limited electrical resources and low Internet connectivity;
- The proposal minimizes the response time for queries made to VAs, improving UX.

The article's structure is as follows: Initially, a description of the methodology and materials used in the work is presented, as well as the mathematical model and prototype implementation. Subsequently, the results obtained are presented and analyzed, and finally, the discussion and conclusions of this work are shown.

2. Materials and Methods

This section presents in the first part, an analysis of the energy consumption of VAs in their passive state, i.e., when the device is turned on but not using voice functions, and in their active state, i.e., when voice functions are being executed. In the second part, a connection architecture is proposed using a cache device. The third part focuses on creating a model that describes the consumption metric of these assistants. Finally, the implementation of a VA to evaluate its energy efficiency and performance is presented. Each of these phases is described below.

2.1. Comparison of Energy Consumption of Voice Assistants Between Active and Passive States

To verify the behavior of devices using VAs regarding their energy consumption, tests were conducted to gather relevant data on battery consumption when the devices execute functions that respond to voice commands. The tests were performed in two different scenarios. In the first scenario, the devices had the VAs in active mode (or On state), utilizing the VAs' functions. In the second scenario, the VAs were in passive mode (Off state), meaning the devices were not using the assistant's functions. The main objective of these tests is to compare the results obtained between the two states and calculate an average of energy consumption in each situation.

Considering that there are numerous VAs [38], including both speaker-type devices like Google Home or Amazon Echo and virtual assistants integrated into mobile devices such as Google Assistant (GA), Apple's Siri, or Microsoft's Copilot, some with AI capabilities, the two most commonly used voice assistants in mobile and desktop devices were selected: GA and Copilot, which also function on Android and Windows operating systems [39,40]. Basic tests were conducted using these assistants, implementing consumption measurement software to identify significant differences that can be crucial for resource optimization in various processes. Table 2 presents the characteristics of the devices used.

Table 2. Characteristics of devices with VAs.

Characteristics	Mobile Phone	Laptop
Device	Celular Xiaomi, Redmi Note 12S (Xiaomi Communications Co, Ltd., Beijing, China)	Laptop Asus, VivoBook M509DA (ASUSTeK Computer Inc., Taipei, Taiwan)
Processor	Octa Core 2.2 Ghz—MediaTek Helio G96 (MediaTek Inc., Hsinchu, Taiwan)	Octa core de 2.1 Ghz, AMD—Ryzen5 (Advanced Micro Devices, Inc., Santa Clara, CA, USA)
Memory	RAM: 12 GB	RAM: 12 GB
Operating system	Android 14	Windows 11 pro-64 bits

In these tests, the energy consumption of these two VAs was measured over 60 min, During this period, 120 queries were conducted, and the battery level was measured every minute. From these results, the arithmetic mean of battery consumption was calculated. Figure 2 shows the average percentage of battery consumption with standard deviations of 0.033% and 0.007% for Copilot in active and passive states, respectively, and 0.021% and 0.004% for GA in active and passive states, respectively.

From the results obtained, similar discharge behaviors between each test are evident, with a significant graphical difference between passive and active states, showing that devices exhibit higher energy consumption while running the VA compared to when this function is idle. Mobile device tests showed that the battery percentage consumed over 60 min was 8% in the active state and 4% in the passive state, which means that, in the active state, the battery duration is approximately half compared to the operating time in the passive state. Similarly, the difference was 43.5% between the two states on the laptop. The above corroborates a direct relationship between the use of VA functions and the device's energy consumption.

Concerning this, measuring the discharge time of VAs starting at 100% battery until it reaches 0% shows that battery life, both on the laptop and mobile, lasts on average twice as long in the passive state as in the active state. Regarding the consumption of these devices, several power measurements were conducted on the equipment in operation, where values similar to the nominal were obtained. In total, 40 VA queries were conducted and measured using a digital power meter (watts), brand Suraielec (Zhengzhou Suraielec Network Tech Co, Zhengzhou, China), with a resolution of 0.1 W. The average power

obtained was 4.8 W on the mobile and 6.5 W on the laptop, with a standard deviation of 0.013 W and 0.021 W, respectively.

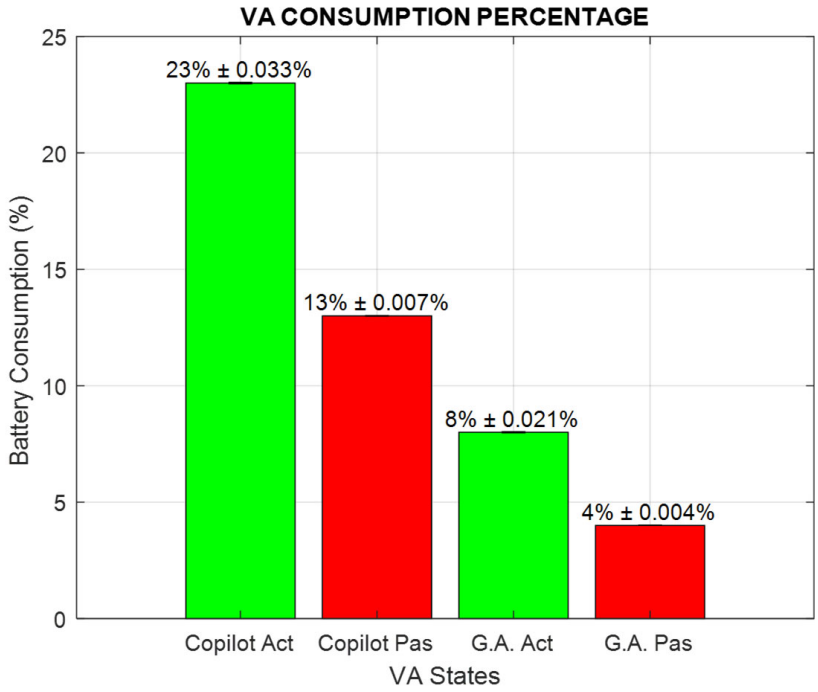

Figure 2. Percentage of battery consumption of VAs in active or passive state.

Although it is typical for VAs to discharge faster in the active state than in the passive state, such a high difference exists, and consequently, a VA user's device battery lasts half the usual time. This work proposes an energy-saving approach for VA-type devices based on these preliminary results.

2.2. Network Connection Architecture with Cache Storage

It is necessary to capture the traffic in plain text and then either respond from the cache or encapsulate it to pass through the channel to the cloud and obtain a response from a remote point to identify the queries made by VAs and store them in the cache. This part of the work proposes a network architecture, Figure 3, which includes several components and functions described below.

1. The first part of the process includes the VA, which must have a microphone, speaker, and communication protocol. This generates the initial data flow (user queries) sent through the network;
2. The second part consists of the cache memory, which stores queries and responses. This can be a local memory, an internal network database, or a cache proxy at the network perimeter. An algorithm then checks if the query is in the cache;
3. The third part involves external network elements, such as cloud servers, routers, and connections with AI systems, where queries not found in the cache will be processed. These responses will be returned to the end user, leaving a copy of the query in local memory.

Thus, this device should respond to future queries, ensuring greater speed in the process and energy savings by not requiring a constant connection to the cloud. It is important to note that the time an object will remain in the cache depends on the specific configuration. Therefore, the Time-To-Live (TTL) of cached objects must be carefully considered to ensure that outdated or incorrect responses are not stored. This point will be addressed in future work. In Section 2.4 of this work, the VA is implemented, and in Section 3 experimental results are presented.

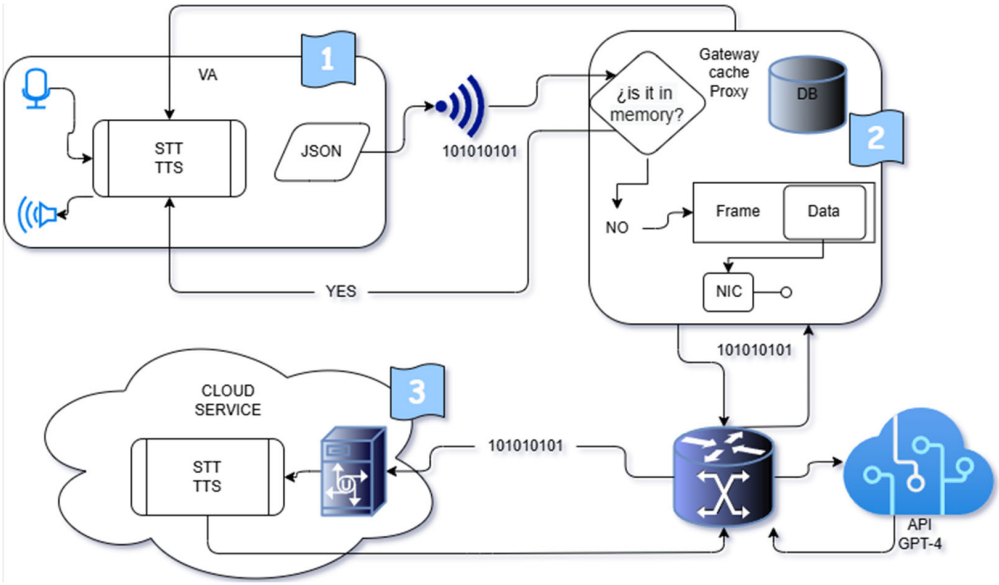

Figure 3. Network architecture for local and remote VA queries.

2.3. Mathematical Model and Evaluation Metric

Mathematical analysis and calculations are performed to characterize the energy consumption and processing times of the VA in the ON state. The consumption can be presented in two scenarios:

1. When the VA operates offline, retrieve the response directly from the local cache or edge devices, but without connecting to the Internet;
2. When the VA seeks the response by connecting to the Internet, make the respective query to cloud databases or through AI.

In both cases, the initial consumption parameters are the same, including WiFi or Ethernet connection, memory usage, CPU cycles, and activation of voice recognition processes. However, for the second scenario, the consumption required by the connection to the mobile network or external ISP network must be considered.

Accordingly, VAs consume energy during the connection process and when processing queries. This situation can be modeled using the power required in each process:

$$P(t) = \begin{cases} P_l, & t_0 < t \leq t_l \\ P_q, & t_l < t \leq t_q \\ P_c, & t_q < t \leq t_c \\ P_i, & t_c < t \leq t_i \\ P_a. & t_i < t \leq t_a \end{cases} \quad (1)$$

where

$P(t)$, is the power function at every time;
$P_l(t)$, is the power of the VA in listening mode;
$P_q(t)$, is the power consumed during the processing of the query;
$P_c(t)$, is the power of the VA while performing a search in the cache memory;
$P_i(t)$, is the power of the VA while retrieving the response from the Internet;
$P_a(t)$, is the power required to deliver the response to the user.
Here, the energy consumption can be represented by the following integral:

$$C = \int_{t_0}^{t_a} P(t)dt \quad (2)$$

In this model, the energy consumption related to processing while the VA is performing other connection processes is disregarded. Considering the previous function, the consumption can be defined as follows for both the case where the response is obtained locally from the cache (Local), C_L, and when the assistant must query in the cloud (External), C_E, for example, through an AI:

$$C_L = \left[\int_{\Delta t_l} P_l dt + \int_{\Delta t_q} P_q dt + \int_{\Delta t_c} P_c dt + \int_{\Delta t_a} P_a dt\right] \tag{3}$$

$$C_E = \left[\int_{\Delta t_l} P_l dt + \int_{\Delta t_q} P_q dt + \int_{\Delta t_c} P_c dt + \int_{\Delta t_i} P_i dt + \int_{\Delta t_a} P_a dt\right] \tag{4}$$

If average power values are assumed for each section of the process, Equation (2) is equivalent to the following:

$$C = \sum_j P_j \Delta t_j. \tag{5}$$

However, for the local case $j = l, q, c, a$ and for the external case $j = l, q, c, i, a$ which means that the following will always hold:

$$C_L = \sum_j P_j \Delta t_j = P_l \Delta t_l + P_q \Delta t_q + P_c \Delta t_c + P_a \Delta t_a \text{ and} \tag{6}$$

$$C_E = \sum_j P_j \Delta t_j = P_l \Delta t_l + P_q \Delta t_q + P_c \Delta t_c + P_i \Delta t_i + P_a \Delta t_a \tag{7}$$

Considering that in practice the power and time associated with the response process P_a are embedded in the consumption calculated for the processes defined in this work as Local (on cache) and External (on the Internet), respectively, this parameter will be omitted in the following equations. Graphically, the power consumption for the query and response process from the cache memory and the AI on the Internet, as described in Equations (6) and (7), is presented in Figure 4.

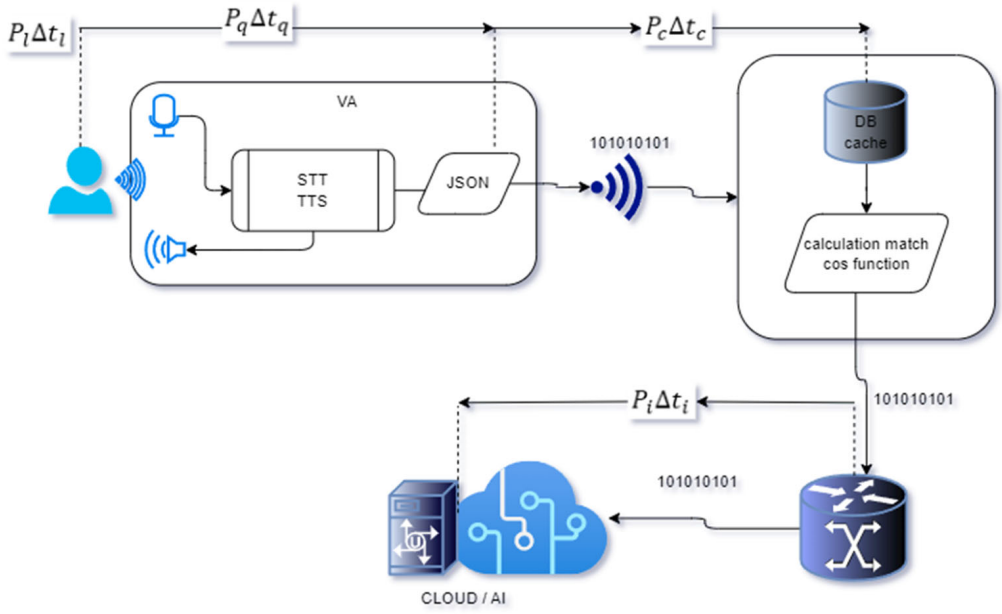

Figure 4. Power consumption according to the proposed architecture.

If the energy consumption while the VA is listening is not of interest for the comparison, i.e., the energy consumption before a query is being detected and answered, then the term

$P_l \Delta t_l$ can be excluded from further consideration. Hence, the energy of interest required in a response process by the VA could be graphically represented as shown in Figure 5.

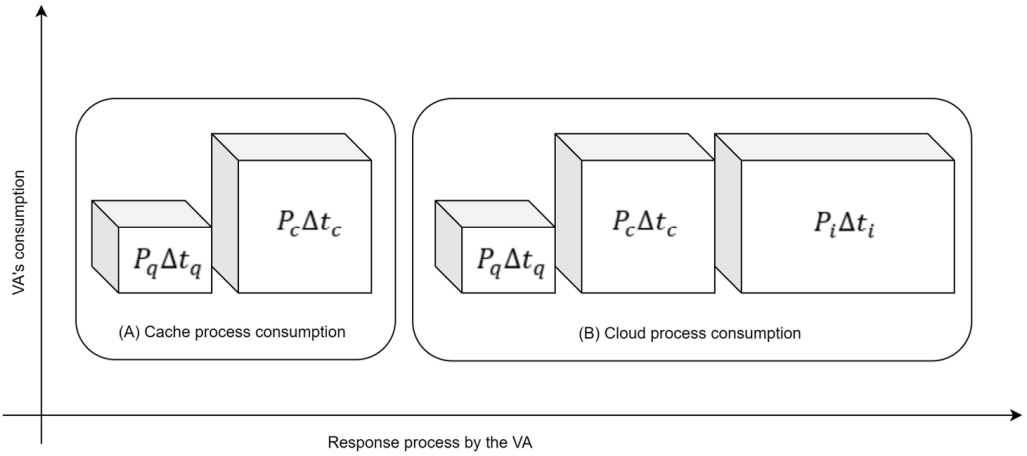

Figure 5. Schematic representation of energy required by the VA depending on the function performed.

The low level indicates the VA's consumption while processing the user's query, and the high level occurs in two moments: one when the VA is searching in the cache and the other when querying the Internet.

Next, the effect of multiple devices within the same network asking the same question Q_j at different times $T_1,\ T_2, \ldots, T_n$ close to each other, is considered. Based on this situation, the following question arises:

Would battery and time be saved if the router or the VA cached the response obtained at time T_1 to handle other queries from other users at later times ($T_n > T_{n-1} > \ldots > T_1$)?

To resolve Q_j, the VA needs to search locally or access only the router that has cached the response from the previous query made at time T_j. In this way,

$$\forall\ T_j > T_1\ \wedge\ \Delta t_i = 0 \tag{8}$$

$$\Rightarrow\ C_L < C_E \tag{9}$$

Then, we define a consumption metric M_C as a comparison between the energy consumption for processing an inquiry (C) and the nominal consumption of the VA as if it were responding from the Internet (C_{N_i}), that would be the maximum processing time, and so, maximum consumption. This consumption is the product of nominal power (P_N) times the required time required in the external connection:

$$M_C = 1 - \frac{C}{C_{N_i}} \tag{10}$$

$$M_C = 1 - \frac{\sum_j P_j \Delta t_j}{P_N\left(\Delta t_q + \Delta t_c + \Delta t_i\right)} \tag{11}$$

This metric is normalized, meaning, $0 < M_C \leq 1$ where $M_C = 1$ in the listening state, i.e., when $\Delta t_q = \Delta t_c = \Delta t_i = 0$ y $M_C \approx 0$ when the process consumes excessive energy in the other stages, i.e., $\sum_j P_j \Delta t_j \rightarrow P_N\left(\Delta t_q + \Delta t_c + \Delta t_i\right)$.

Optimization Mathematical Model

The mathematical automation model for consumption optimization depends on finding the minimum power and time required for the process. It means minimizing Equation (5) for consumption C:

$$\min[C] = \min\left[\sum\nolimits_j P_j \Delta t_j\right] \tag{12}$$

If the part of consumption associated with the query and cache processes are combined into one part called ΔC_c and the processes involving Internet connection are combined into a part called ΔC_i, then the total consumption C would be given by the following:

$$C = \Delta C_c + \Delta C_i \tag{13}$$

Given the previous parametrization, the goal is to minimize the consumption of the VAs, which in turn depends on the time it takes to perform a query and obtain a response. To optimize this time, it is proposed to minimize or reduce the consumption time for the Internet connection to zero by obtaining responses directly from the cache server. Therefore, if a query is made only locally, then

$$\Delta C_i = P_i \Delta t_i = 0 \tag{14}$$

Thus C_L is the consumption for the query up to the cache device. Hence, we can define C_L as follows:

$$C_L = P_c \Delta t_c \tag{15}$$

Considering that both consumption and time are always zero or positive,

$$P_i \Delta t_i \geq 0 \tag{16}$$

Then,

$$C_L \leq C \tag{17}$$

It follows that if the device does not need to query the Internet, the consumption will always be less than or equal to that of querying over the Internet. Thus, a metric M_{Cc} for queries within the internal network is proposed, eliminating the power consumption of the connection and response from the Internet, which can be expressed from Equation (11) (the initial metric) by removing the external connection consumption:

Summarized, it would be

$$M_{C_L} = 1 - \frac{P_q \Delta t_q + P_c \Delta t_c}{P_N\left(\Delta t_q + \Delta t_c + \Delta t_i\right)} \tag{18}$$

For greater accuracy in the optimization process, the following parameters will be considered, each having a determined value:

- Standard or version supported by the WiFi network devices Institute of Electrical and Electronics Engineers (IEEE). 802.11;
- Processing capacity of the devices (VA and WiFi Router);
- Algorithm for cache storage and response to users.

2.4. Implementation of a Voice Assistant with Cache Storage

In this experimental phase, a device capable of utilizing natural language recognition and processing techniques was created, providing a highly energy-efficient and much faster

VA compared to conventional VAs. It is essential to clarify that the tests conducted in this work are independent of those used for the system's design and creation.

Implementation of a VA on Raspberry Pi Using Python

One of the processes required by VAs is synthesizing voice traffic into text. This requires proprietary or standard protocols to perform this task, either on the local device or the ISP cloud servers. To implement a VA system that allows receiving queries, obtaining responses from an AI system in the cloud, and storing this information in a cache to return it to the user later, a state-of-the-art device with sufficient hardware capabilities was used for the proper functioning of a VA and the implementation of local cache. In this case, a Raspberry Pi 4 with 4 GB of RAM and a 1.5 GHz processor was used, programmed with Python version 3.11 and an omnidirectional USB 2.0 microphone with 1 dB sensitivity.

For this experimental phase, the following configurations and implementations were carried out:

Implementation of an STT system using Vosk (vosk-model-es-0.42) [13]: this tool allows a pre-trained model to recognize multiple languages, interpret voice with its different tones, and convert it into text.

Interaction with a Natural Language Processing (NLP) system [41]: TensorFlow Text library was used to create a model for interpreting user queries. Since humans can ask the same thing in various ways in natural language, an AI, which is capable of calculating the similarity between two texts in percentage, was necessary. This helps determine if a query associated with a response had been previously made. If so, the response is retrieved from the cache. The AI can validate if two requests are equivalent using a probabilistic method based on the cosine function, which defines a similarity value. If the defined threshold is met, the AI compares it with stored responses using the same cosine function to find the appropriate answer for the user.

Use of Generative AI (cloud service) [42]: the text converted from voice represents a question/query. If it is not in the cache, a Generative AI resolves it. OpenAI (gpt-4) was chosen for its reliability, high recognition, and usability in the current world.

Storage of questions and answers in a cache: pairs of questions and their answers are stored in memory, and a file with this information is created, allowing it to be preserved over time.

Translation of text responses to speech (TTS) (pyttsx3) [43]: the response delivered to the user is synthesized into speech using the text-to-speech system.

3. Results

After implementing the VA system, various tests were conducted. The most representative test involved launching a query, searching the Internet for it, and storing the response in a cache. Subsequently, the same or a similar question was posed. The algorithm determines the similarity with the information stored in the cache and verifies if it is the same query. Depending on this, the response is delivered to the user either from cache or by searching the Internet. Additionally, the algorithm calculates the time elapsed in each part of the process performed, achieving millisecond-level accuracy. Figure 6 shows a screenshot of the prototype displaying the time results obtained with the algorithm during a test. This test demonstrates the difference in response times between cloud-based AI and responses retrieved from the local cache.

In this exercise, for the tests (queries to the VAs), a bank of questions was created, classified into 5 categories: Weather (Q1), Places (Q2), History (Q3), Mathematics (Q4), and Characters (Q5).

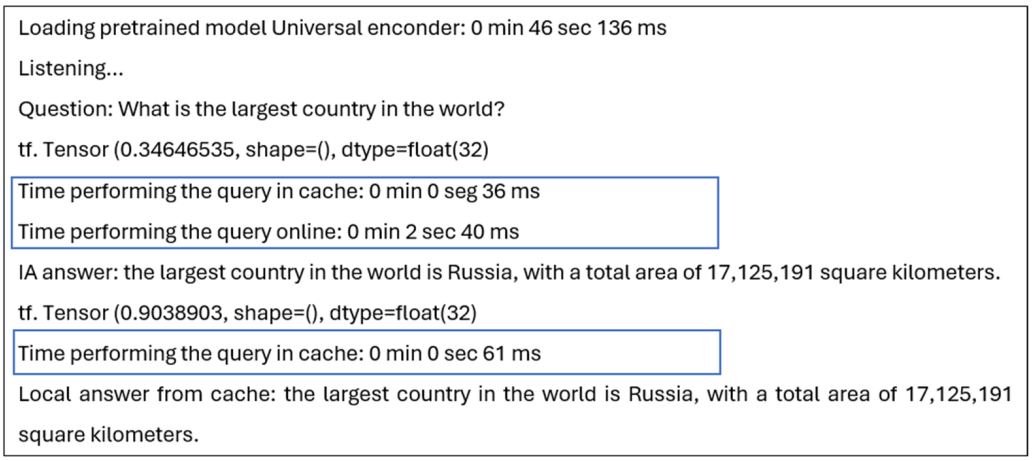

Figure 6. Direct image from the prototype display showing a test of elapsed times for cloud and local cache queries. The blue boxes highlight the performing times.

These queries showed that response times and energy consumption might depend on the type of topic or the size of the query itself. However, the relationship and differences between the results from the cache and those from the cloud remained consistent; in other words, response times were always higher when an internet connection was required. Table 3 presents 10 questions per defined category, each with their respective results in terms of measured time and power. The system's power consumption does not vary significantly across different queries or categories, with the average measured operating power of the virtual assistant remaining around 3.15 W.

Subsequently, the averages for each category are calculated and presented in Table 4. In this case, query time, cache time, and cloud time are distinguished for each query. The standard deviation of response times for cache queries is 0.24 s, and for cloud tests, it is 1.95 s.

Figure 7 illustrates the average times measured for each stage of the process, the time the VA takes to pose a question, the time to retrieve a response from the cache, and finally, the time required to obtain a response from the cloud.

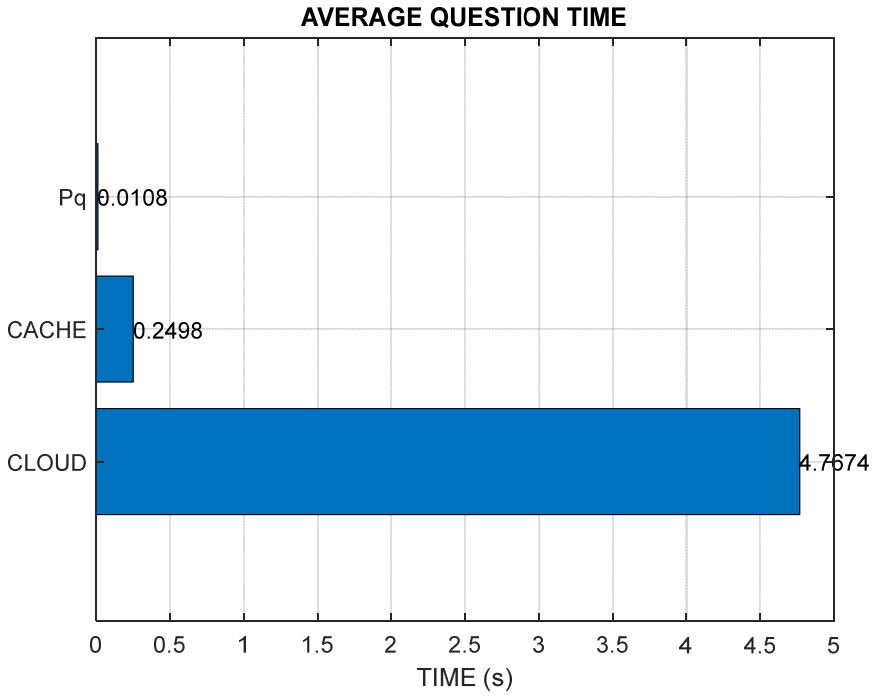

Figure 7. Average time for query/cache/cloud.

Table 3. Comparison of response times from cache and cloud per category.

Category	Test	Time [s]		Power [W]
		Cache	Cloud	
Weather	1	0.048	7.903	3.15
	2	0.031	8.717	
	3	0.013	6.850	
	4	0.070	7.994	
	5	0.011	8.556	
	6	0.098	5.980	
	7	0.049	7.911	
	8	0.052	8.087	
	9	0.120	7.949	
	10	0.018	8.890	
Places	1	0.061	2.762	3.21
	2	0.028	4.838	
	3	0.014	2.918	
	4	0.027	3.606	
	5	0.006	4.784	
	6	0.018	3.870	
	7	0.045	2.097	
	8	0.040	3.501	
	9	0.086	4.569	
	10	0.029	2.990	
Hystory	1	0.011	3.789	3.13
	2	0.019	4.881	
	3	0.053	5.059	
	4	0.030	4.941	
	5	0.041	5.058	
	6	0.004	3.902	
	7	0.057	6.784	
	8	0.016	4.197	
	9	0.025	5.629	
	10	0.044	3.479	
Math	1	0.012	5.780	3.15
	2	0.053	7.018	
	3	0.045	6.192	
	4	0.026	6.118	
	5	0.043	6.197	
	6	0.020	5.601	
	7	0.047	5.866	
	8	0.034	6.514	
	9	0.049	7.303	
	10	0.036	6.734	
Characters	1	0.553	2.797	3.09
	2	0.642	3.025	
	3	0.721	2.122	
	4	0.517	3.288	
	5	0.740	3.121	
	6	0.597	2.614	
	7	0.608	3.580	
	8	0.396	4.319	
	9	0.679	2.743	
	10	0.705	3.316	

Table 4. Comparison of response times from cache and cloud.

Question Number	Response from	Question Processing Time (Pq) [s]	Cache Time (Pc) [s]	Cloud Time (Pi) [s]	Total Query Response Time [s]
Q1	Cloud	0.016	1.621	6.250	7.887
	Cache	0.006	0.045		0.051
Q2	Cloud	0.022	0.031	3.539	3.592
	Cache	0.006	0.029		0.035
Q3	Cloud	0.012	0.046	4.714	4.772
	Cache	0.006	0.024		0.030
Q4	Cloud	0.009	0.029	6.294	6.332
	Cache	0.009	0.027		0.036
Q5	Cloud	0.016	0.036	3.040	3.092
	Cache	0.006	0.610		0.616

As seen in Table 4, the response times for each test in cache and from the Internet are different. Furthermore, when analyzing the total time of the tests with responses from the cache compared to the total time from the cloud, a significant difference is identified: 0.768 s vs. 25.675 s. This leads to a substantial difference in energy consumption, which will ultimately be reflected in monetary costs. Thus, based on the response times obtained from each part of the process and considering the power measured during the query tests using the digital meter, which averaged 3.15 W, and the nominal power, meaning the maximum power consumed by the Raspberry Pi, which would be 4 W, the energy consumption of the VA in its ON state was calculated.

$$E[jules] = P[W] \times t[seconds] \tag{19}$$

Different results were obtained in terms of energy consumption for both local and remote (Internet) queries. Additionally, these times were used to predict the initial VAs (GA and Copilot) for which the experimentally measured power was 3.8 W and 5.5 W for each device, and the power at maximum load was 4.8 W and 6.5 W, respectively. With these data, the energy consumption levels for such VAs are presented in Table 5. The energy is shown in joules for both the Raspberry Pi and the initial VAs.

Table 5. Energy consumption on prototype vs. expected energy consumption on initials VAs.

Test	Joules (Energy Consumption) Raspberry Pi		Joules (Energy Consumption) VA1		Joules (Energy Consumption) VA2	
	Cache	Cloud	Cache	Cloud	Cache	Cloud
Q1	0.16	24.84	0.19	29.97	0.28	43.37
Q2	0.11	11.53	0.13	13.64	0.19	19.75
Q3	0.094	14.93	0.11	18.13	0.16	26.24
Q4	0.11	19.93	0.13	24.06	0.19	34.82
Q5	1.9	9.55	2.34	11.74	3.38	17.00
Total	2.38	80.78	2.9	97.54	4.2	141.18

The calculations for the proposed prototype indicate a difference of 78.4 J in energy consumption between cache-based and cloud-based queries. This means that a cache query consumes only 2.95% of the energy required for a cloud-based query, clearly highlighting a significant reduction in energy usage.

Subsequently, the metrics of the proposed mathematical model were verified using the average values measured in the prototype; P_p is defined as the average power used in the query, cache, Internet, and response processes. These metrics yield the following numerical results:

The metric for the energy consumption of queries performed in the cloud (External query):

$$M_C = 1 - \frac{P_p\left(\Delta t_q + \Delta t_c + \Delta t_i\right)}{P_N\left(\Delta t_q + \Delta t_c + \Delta t_i\right)}$$
$$M_C = 0.21 \tag{20}$$

The metric for the energy consumption of queries performed in the cache (Local query):

$$M_{C_L} = 1 - \frac{P_p\left(\Delta t_q + \Delta t_c\right)}{P_N\left(\Delta t_q + \Delta t_c + \Delta t_i\right)}$$
$$M_{C_L} = 0.99 \tag{21}$$

This result shows that the difference between the consumption metrics for the cache tests and the cloud tests is 0.78.

4. Discussion

In the previous studies reviewed and analyzed, key studies include Lanyu Xu's CHA cache storage framework, Mikhail's memory reduction proposal, V.P. Sriram's VA system, and Boubakr Nour's edge learning architecture for privacy enhancement [29,30,44,45]. These solutions focus on automated systems like home automation, where interconnected devices respond to various events, commands, or voice orders. However, while the first two studies implement a cache memory, they do not store data or responses obtained from the cloud associated with the different user queries.

On the other hand, research on energy optimization through the implementation of VAs primarily focuses on energy management in homes or some industrial settings. Notable studies include those by Sarah et al., 2023, and M. U. Hadi, 2022 [27,46], and the minimization of energy consumption in smart buildings through real-time data processing presented by Amal Zouhri, 2024 [23]. Regarding energy measurement and optimization in sensor networks or systems involving IoT devices, some notable studies include those by Sibel Tombaz, 2014, and Bin Li et al., 2012 [21,22]. However, these studies do not involve VAs in their applications.

Additionally, it is possible to affirm that the metrics found, such as those by Ergasheva et al., 2020, and Tso et al., 2017 [36,37], are not aligned with the same conditions associated with the system proposed in this work.

It is important to highlight that none of the studies listed in Table 1 comprehensively address all the research areas identified in this study, such as energy savings, voice recognition, cache memory management, and metric development. Additionally, we did not find models that implement cache usage in VAs with a similar purpose to ours. Some cited studies that consider cache usage [29,30] do so with different objectives, such as task automation in homes or specific locations, memory management, or improving processing efficiency. However, they do not focus on reducing energy consumption in VAs performing any type of query to a local server or the cloud. In our case, we propose an approach that, to the best of our knowledge, is novel, specifically aimed at optimizing energy consumption through the use of cache for voice assistants. Since the evaluation objectives of these studies differ from ours, making a direct comparison with these models would not be appropriate or representative.

In terms of consumption behavior for cloud queries, it can be determined that communication technology influences the energy consumption of VAs, as connection bandwidth directly impacts the response time for queries, which, in turn, generates variations in energy consumption. The tests conducted on our model were performed using a fiber optic network with a 700 Mbps internet connection channel. The results of these tests are presented in the measurement tables for cloud query times.

In this context, and considering related studies [47], if tests were performed using a 4 G mobile network with a bandwidth of 100 Mbps, nominal response times would be approximately seven times slower. In contrast, using a 5 G network with a nominal capacity of 10 Gbps, response times could be up to 14 times faster.

Based on the results obtained and in line with the proposal of this work, the primary reason for the significant difference in the metrics is the use of cache memory, which significantly reduces response times. Specifically, cloud responses are influenced by factors such as the type of language model used [48] and the conditions of the cloud connection, which directly affect response times and, consequently, the differences in metrics.

The results show that the time spent on cache operations is negligible or has minimal impact compared to the time required to obtain a response from the cloud. This suggests that this solution could be implemented in most systems without incurring significant costs compared to systems without this optimization, and it could potentially help reduce energy consumption. Additionally, the findings also indicate that in scenarios with poor connectivity, mobility, or other adverse conditions, cache memory serves as an efficient and valuable resource.

5. Conclusions

In this work, a prototype of a VA was designed and implemented. It was trained with AI capable of performing natural language recognition and comparisons between different user queries, storing them in cache memory for faster responses in future similar queries. This system also adds a layer of security by keeping the information within a private network.

The prototype successfully reduces energy consumption by an average of 2.95% of the total energy when using the cache, as opposed to what is required for cloud-based queries. Additionally, this implementation achieves significantly lower response times, on average 33 s faster than VAs that query the cloud, improving response times in critical processes and enhancing user experience.

Another contribution of this work is a mathematical model based on the system's own attributes. This model allows representation of the elements and parts of the process that VAs perform and their relationship with energy consumption when connected to the cloud versus local connections.

The presented metric allows for determining the energy consumption of voice assistants as a function of time. This was verified through mathematical calculations of power and energy, along with experimental tests on the prototype created. The results showed a high difference in metric (0.21 vs. 0.99) between queries to the cache memory and those made to the cloud. The large-scale implementation of this solution could significantly reduce the environmental impact by achieving considerable global energy savings.

The proposed solution has several potential applications, particularly in situations where mobility affects communication, in remote or isolated areas with no signal coverage, in populations without Internet access, in areas with high concentrations of people and devices, and for scenarios where VAs are required to handle concurrent similar queries.

This work opens up new lines of research, allowing for further exploration into energy-saving techniques by implementing cache memory for encrypted information outside the local device.

Author Contributions: Conceptualization, A.O.M.B., Á.S.S., E.M.M.L. and J.H.-R.; methodology, A.O.M.B., Á.S.S. and E.M.M.L.; software, J.H.-R.; validation, A.O.M.B. and Á.S.S.; formal analysis, A.O.M.B.; investigation, A.O.M.B. and J.H.-R.; resources, A.O.M.B. and J.H.-R.; data curation, E.M.M.L. and J.H.-R.; writing—original draft preparation, A.O.M.B.; writing—review and editing, E.M.M.L. and Á.S.S.; visualization, J.H.-R.; supervision, Á.S.S. and E.M.M.L.; project administration, A.O.M.B., E.M.M.L. and Á.S.S.; funding acquisition, A.O.M.B. and J.H.-R. All authors have read and agreed to the published version of the manuscript.

Funding: The development of this research was supported by the ULPGC—University of Las Palmas de Gran Canaria in the discount voucher received for the APC payment of this article.

Data Availability Statement: The data may be provided free of charge to interested readers by requesting via the corresponding author's email.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ud Din, I.; Guizani, M.; Hassan, S.; Kim, B.S.; Khurram Khan, M.; Atiquzzaman, M.; Ahmed, S.H. The Internet of Things: A Review of Enabled Technologies and Future Challenges. *IEEE Access* **2019**, *7*, 7606–7640. [CrossRef]
2. de Freitas, M.P.; Piai, V.A.; Farias, R.H.; Fernandes, A.M.R.; de Moraes Rossetto, A.G.; Leithardt, V.R.Q. Artificial Intelligence of Things Applied to Assistive Technology: A Systematic Literature Review. *Sensors* **2022**, *22*, 8531. [CrossRef] [PubMed]
3. Dian, F.J. *Fundamentals of Internet of Things: For Students and Professionals 1st Edición*, 1st ed.; Wiley-IEEE Press: Piscataway, NJ, USA, 2022; ISBN 111984729X.
4. Lazaro, A.; Villarino, R.; Girbau, D. A Survey of NFC Sensors Based on Energy Harvesting for IoT Applications. *Sensors* **2018**, *18*, 3746. [CrossRef] [PubMed]
5. Liu, S.; Lee, J.-Y.; Cheon, Y.; Wang, M. A Study of the Interaction between User Psychology and Perceived Value of AI Voice Assistants from a Sustainability Perspective. *Sustainability* **2023**, *15*, 11396. [CrossRef]
6. Ospina Cifuentes, B.J.; Suárez, Á.; García Pineda, V.; Alvarado Jaimes, R.; Montoya Benitez, A.O.; Grajales Bustamante, J.D. Analysis of the Use of Artificial Intelligence in Software-Defined Intelligent Networks: A Survey. *Technologies* **2024**, *12*, 99. [CrossRef]
7. Siemers, W.; Sallou, J.; Cruz, L. The Two Faces of AI in Green Mobile Computing: A Literature Review. In Proceedings of the 2023 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Durres, Albania, 6–8 September 2023; pp. 301–309.
8. Kurz, M.; Brüggemeier, B.; Breiter, M. Success Is Not Final; Failure Is Not Fatal—Task Success and User Experience in Interactions with Alexa, Google Assistant and Siri. *Lect. Notes Comput. Sci. (Incl. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinform.)* **2021**, *12764 LNCS*, 351–369. [CrossRef]
9. Rawat, D.B.; Doku, R.; Garuba, M. Cybersecurity in Big Data Era: From Securing Big Data to Data-Driven Security. *IEEE Trans. Serv. Comput.* **2021**, *14*, 2055–2072. [CrossRef]
10. Alchekov, S.S.; Al-Absi, M.A.; Al-Absi, A.A.; Lee, H.J. Inaudible Attack on AI Speakers. *Electronics* **2023**, *12*, 1928. [CrossRef]
11. Montoya, B.A.O.; Munoz, G.M.A.; Kofuji, S.T. Performance Analysis of Encryption Algorithms on Mobile Devices. In Proceedings of the 2013 47th International Carnahan Conference on Security Technology (ICCST), Medellin, Colombia, 8–11 October 2013; pp. 1–6.
12. Xu, K.; Zhang, M.; Liu, J.; Qin, Z.; Ye, M. Proxy Caching for Peer-to-Peer Live Streaming. *Comput. Netw.* **2010**, *54*, 1229–1241. [CrossRef]
13. Setiawan, P.; Yusuf, R. IoT Device Control with Offline Automatic Speech Recognition on Edge Device. In Proceedings of the 12th International Conference on System Engineering and Technology, ICSET 2022—Proceeding, Bandung, Indonesia, 3–4 October 2022; pp. 111–115. [CrossRef]
14. Giraldo, J.S.P.; Lauwereins, S.; Badami, K.; Verhelst, M. Vocell: A 65-Nm Speech-Triggered Wake-Up SoC for 10-MW Keyword Spotting and Speaker Verification. *IEEE J. Solid-State Circuits* **2020**, *55*, 868–878. [CrossRef]
15. Li, J.; Xie, L.; Chen, Z.; Shi, L.; Chen, R.; Ren, Y.; Wang, L.; Lu, X. An AIoT-Based Assistance System for Visually Impaired People. *Electronics* **2023**, *12*, 3760. [CrossRef]

16. Errobidart, J.; Uriz, A.J.; Gonzalez, E.; Gelosi, I.E.; Etcheverry, J.A. Offline Domotic System Using Voice Comands. In Proceedings of the 2017 Eight Argentine Symposium and Conference on Embedded Systems (CASE), Buenos Aires, Argentina, 9–11 August 2017; pp. 1–6.
17. Irugalbandara, C.; Naseem, A.S.; Perera, S.; Kiruthikan, S.; Logeeshan, V. A Secure and Smart Home Automation System with Speech Recognition and Power Measurement Capabilities. *Sensors* **2023**, *23*, 5784. [CrossRef]
18. Froiz-Míguez, I.; Fraga-Lamas, P.; Fernández-CaraméS, T.M. Design, Implementation, and Practical Evaluation of a Voice Recognition Based IoT Home Automation System for Low-Resource Languages and Resource-Constrained Edge IoT Devices: A System for Galician and Mobile Opportunistic Scenarios. *IEEE Access* **2023**, *11*, 63623–63649. [CrossRef]
19. Satyanarayana, S.K.; Devi, L.N.; Rao, A.N. MPIGA—Multipath Selection Using Improved Genetic Algorithm. *Int. J. Commun. Netw. Inf. Secur. (IJCNIS)* **2022**, *14*, 67–78. [CrossRef]
20. Kumar, S. An Energy-Efficient Multi-Channel Design for Distributed Wireless Sensor Networks. *Int. J. Grid High Perform. Comput.* **2023**, *15*, 1–18. [CrossRef]
21. Tombaz, S.; Sung, K.W.; Zander, J. On Metrics and Models for Energy-Efficient Design of Wireless Access Networks. *IEEE Wirel. Commun. Lett.* **2014**, *3*, 649–652. [CrossRef]
22. Li, B.; Wang, W.; Yin, Q.; Yang, R.; Li, Y.; Wang, C. A New Cooperative Transmission Metric in Wireless Sensor Networks to Minimize Energy Consumption per Unit Transmit Distance. *IEEE Commun. Lett.* **2012**, *16*, 626–629. [CrossRef]
23. Zouhri, A.; Ez-Zahout, A.; Chakouk, S.; EL Mallahi, M. A Numerical Analysis Based Internet of Things (IOT) and Big Data Analytics to Minimize Energy Consumption in Smart Buildings. *J. Autom. Mob. Robot. Intell. Syst.* **2024**, *18*, 46–56. [CrossRef]
24. Li, C.; Zhu, L.; Luo, Y. Latency-Aware Content Caching and Cost-Aware Migration in SDN Based on MEC. *Wirel. Netw.* **2021**, *27*, 5329–5349. [CrossRef]
25. Kang, Y.; Hauswald, J.; Gao, C.; Rovinski, A.; Mudge, T.; Mars, J.; Tang, L. Neurosurgeon: Collaborative Intelligence between the Cloud and Mobile Edge. *ACM Sigplan Not.* **2017**, *52*, 615–629. [CrossRef]
26. Erata, F.; Yildiz, E.; Goknil, A.; Yildirim, K.S.; Szefer, J.; Piskac, R.; Sezgin, G. ETAP: Energy-Aware Timing Analysis of Intermittent Programs. *ACM Trans. Embed. Comput. Syst.* **2023**, *22*, 23. [CrossRef]
27. Hadi, M.U.; Suhaimi, N.H.N.; Basit, A. Efficient Supervised Machine Learning Network for Non-Intrusive Load Monitoring. *Technologies* **2022**, *10*, 85. [CrossRef]
28. Abolhassani, B.; Tadrous, J.; Eryilmaz, A.; Yeh, E. Fresh Caching of Dynamic Content Over the Wireless Edge. *IEEE/ACM Trans. Netw.* **2022**, *30*, 2315–2327. [CrossRef]
29. Xu, L.; Iyengar, A.; Shi, W. CHA: A Caching Framework for Home-Based Voice Assistant Systems. In Proceedings of the 2020 IEEE/ACM Symposium on Edge Computing (SEC), San Jose, CA, USA, 12–14 November 2020; pp. 293–306.
30. Rovnyagin, M.M.; Sinelnikov, D.M.; Eroshev, A.A.; Rovnyagina, T.A.; Tikhomirov, A.V. Optimizing Cache Memory Usage Methods for Chat LLM-Models in PaaS Installations. In Proceedings of the 2024 Conference of Young Researchers in Electrical and Electronic Engineering (ElCon), Saint Petersburg, Russian Federation, 29–31 January 2024; pp. 277–280.
31. Abhishek, R.B.; Abhishek, S.V.; Akash, A.; Amruth, P.S. Voice over Internet Protocol (VoIP)-A Review. *Int. J. Innov. Sci. Res. Technol.* **2022**, *7*, 850–853. [CrossRef]
32. Singh, H.; Singh, J.; Bhatti, S. Speech Communication Using DSP in VoIP. In Proceedings of the 5th Conference in the Series International Conference on Intelligent Systems & Networks (ISN-2008), Hong Kong, China, 28–31 July 2008.
33. Kumar, A. Yash Performance Evaluation of Video Streaming Traffic in Data Centre Servers Using Real- Time Transport Protocol (RTP). *Int. J. Emerg. Technol. Innov. Eng.* **2020**, *6*, 472–477.
34. Fazel, A.; El-Khamy, M.; Lee, J. CAD-AEC: Context-Aware Deep Acoustic Echo Cancellation. In Proceedings of the ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing-Proceedings 2020, Barcelona, Spain, 4–8 May 2020; pp. 6919–6923. [CrossRef]
35. Miguel Soto; AWS Editorial Team. Uso de Un Volumen de AWS Storage Gateway En Modo Caché Con SQL Server. Available online: https://aws.amazon.com/es/blogs/aws-spanish/uso-de-un-volumen-de-aws-storage-gateway-en-modo-cache-con-sql-server/ (accessed on 25 May 2024).
36. Ergasheva, S.; Khomyakov, I.; Kruglov, A.; Succil, G. Metrics of Energy Consumption in Software Systems: A Systematic Literature Review. In Proceedings of the IOP Conference Series: Earth and Environmental Science; Institute of Physics Publishing: London, UK, 2020; Volume 431.
37. Tso, G.K.F.; Yau, K.K.W. Predicting Electricity Energy Consumption: A Comparison of Regression Analysis, Decision Tree and Neural Networks. *Energy* **2007**, *32*, 1761–1768. [CrossRef]
38. Sati, B.; Kumar, S.; Rana, K.; Saikia, K.; Sahana, S.; Das, S. An Intelligent Virtual System Using Machine Learning. In Proceedings of the 2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET), Arad, Romania, 20–22 May 2022; pp. 1123–1129.
39. Jamison, N. From Agent Assist to Employee Assist: Copilot Apps Are Proliferating, and They Mean Business. *Speech Technol. Mag.* **2024**, *29*, 6.

40. Sivapriyan, R.; Sakshi, N.; Vishnu Priya, T. Comparative Analysis of Smart Voice Assistants. In Proceedings of the 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Bangalore, India, 16–18 December 2021; pp. 1–6.
41. Mengle, S.S.R.; Maximo, G. *Mastering Machine Learning on AWS: Advanced Machine Learning in Python Using SageMaker, Apache Spark, and TensorFlow*; Packt Publishing, Ed.; Packt Publishing: Birmingham, UK, 2019.
42. Duval, B. *The Definitive ChatGPT Handbook: Techniques, Prompts and AI Strategies for Business, Marketing, Creative Writing AND MORE*; AI DEVOURER; Independently Published: Chicago, IL, USA, 2024; ISBN 979-8880224197.
43. Croyden, A. *Python Programming for Beginners*; Independently Published: Chicago, IL, USA, 2024; ISBN 979-8323169207.
44. Sriram, V.P.; Kamalakkannnan, D.; Archana, T.; Gopatoti, A.; Swapna, B.; Yadav, A.S. Design of Voice Based Virtual Assistant Using Internet of Things. In Proceedings of the 2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India, 24–26 November 2022; pp. 1152–1155.
45. Nour, B.; Cherkaoui, S.; Mlika, Z. Federated Learning and Proactive Computation Reuse at the Edge of Smart Homes. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 3045–3056. [CrossRef]
46. Himer, S.E.; Ouaissa, M.; Ouaissa, M.; Krichen, M.; Alswailim, M.; Almutiq, M. Energy Consumption Monitoring System Based on IoT for Residential Rooftops. *Computation* **2023**, *11*, 78. [CrossRef]
47. Lopes, C.H.d.S.; Lima, E.S.; Pereira, L.A.M.; Borges, R.M.; Ferreira, A.C.; Abreu, M.; Dias, W.D.; Spadoti, D.H.; Mendes, L.L.; Junior, A.C.S. Non-Standalone 5G NR Fiber-Wireless System Using FSO and Fiber-Optics Fronthauls. *J. Light. Technol.* **2020**, *39*, 406–417. [CrossRef]
48. Oralbekova, D.; Mamyrbayev, O.; Othman, M.; Kassymova, D.; Mukhsina, K. Contemporary Approaches in Evolving Language Models. *Appl. Sci.* **2023**, *13*, 12901. [CrossRef]