UDC 004.4
DOI https://doi.org/10.32782/2663-5941/2025.3.2/17

Vernik M. O.
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"

Oleshchenko L. M.
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"

INTELLIGENT SOFTWARE FRAMEWORK FOR BIG DATA AUDIO PROCESSING AND COGNITIVE ENERGY ESTIMATION

The article presents an innovative software framework for the analysis of human-produced audio aimed at quantifying cognitive energy. A theoretical model has been proposed for representing daily cognitive energy (E), time (T), and actions (D1...Dn), with a distinction between productive and unproductive activities, as well as the identification of a specific "flow state." A new metric, Countable Cognitive Energy (CCE), has been introduced, derived from transcribed audio of daily tasks. The technical system architecture has been described, which includes continuous audio recording (~16 hours/day, ~1.2 GB/day per user) via wearable devices, transcription using Whisper AI, and segmentation and labeling of data for further analysis. The software system comprises a React-based interface for visualization and upload, Azure Blob Storage for scalable data storage, Azure Functions for transcription orchestration and NLP processing, a .NET API for access control, and a Python backend for speech and language processing. The architectural design of developed software has been illustrated with data flow diagrams and wearable device schematics. Key limitations of proposed software system have been examined, such as transcription delays, GPU constraints, and bandwidth limitations. The scalability of the system has been analyzed. The proposed approach has been compared to existing software solutions (LENA, TILES Audio Recorder, Friend wearable), with an emphasis on its ability to visualize user productivity patterns and detect flow states. Relevant literature in lifelogging and productivity research has been reviewed. The prospects for future research include improving NLP-based classification of activities, validating CCE through user studies, and incorporating additional contextual sensors such as computer usage and geolocation. Potential extensions of the framework to distinguish between cognitive and physical energy have been discussed.

Key words: *software framework, big data, audio, mp3, wav, cognitive energy, human-produced data, recorder, NLP, Whisper AI, Python, Azure Blob Storage.*

Statement of the problem. Continuous self-monitoring through lifelogging offers unprecedented outcomes into personal productivity, time use, and mental state. Motivated by the idea of Maximum Productivity, a conceptual framework positing that individuals have finite daily energy (*E*) and time (*T*) which must be optimally allocated among actions $D_{1...n}$ – we propose a holistic system merging theory and practice. Philosophical ideas like energy depletion and "flow state" productivity gains marketingmag.com.au are quantified via continuous audio lifelogging using a wearable device.

Prior works such as the LENA device (Language ENvironment Analysis) [1] have demonstrated the feasibility and research value of all-day audio recording in specific contexts. More recent developments, including wearable "always listening" assistants (Bee AI, Omi), underline emerging commercial interest in passive audio capture for personal analytics [2].

Analysis of recent research and publications. Early lifelogging visions like MyLifeBits and Total Recall advocated capturing one's entire life digitally. Continuous audio recording has been explored via devices like LENA (for child language environments) and research prototypes like the Electronically Activated Recorder (EAR). The TILES Audio Recorder (TAR) introduced a wearable sensor and mobile app for recording audio features in sensitive environments, demonstrating passive capture with privacy-preserving feature extraction [3]. Such systems aimed to minimize intrusiveness and maximize battery life. Recent wearable recorders (e.g., "Friend" pendant on Kickstarter, and devices by Bee AI and Omi) show the transition from lab

prototypes to consumer products. While these devices capture and transcribe conversations to provide summaries and to-do lists, our system uniquely focuses on interpreting transcripts through the lens of productivity theory to derive cognitive energy usage metrics. Psychologist Mihaly Csikszentmihalyi defined flow as a state of complete absorption in an activity where time seems to vanish [4]. Flow is associated with peak performance. Flow and productivity have been linked to optimal energy management, more so than mere time management.

The concept of ego depletion suggests cognitive energy is finite and can be exhausted by tasks requiring self-control. We draw on these ideas to categorize recorded activities and estimate how each consumes or replenishes a person's cognitive energy E [5]. Our work is informed by previous productivity literature, including Deep Work (advocating focused, flow-like work states) and personal analytics tracking efforts (e.g., lifelogging diaries on LessWrong).

Task statement. This research is structured around a set of guiding questions that emerged during the daily activity process.

1. Formalize a mathematical model of daily cognitive energy (E), time (T), and actions ($D_1 \ldots D_n$), including productive vs. unproductive categorization and the flow-state parameter (φ). Define and validate the Countable Cognitive Energy (CCE) formula by assigning energy-rate weights (w_i) to transcribed activity segments. Which features – textual, temporal, and behavioral – can be extracted efficiently and interpreted meaningfully to support future analytical tasks such as narrative detection or engagement modeling?

2. Design a low-power wearable audio recorder microscheme for continuous ~16 h/day capture. Specify an end-to-end cloud pipeline (React front-end, .NET API, Azure Blob Storage, Azure Functions, Python/Whisper NLP) for large-scale ingestion, transcription, analysis, and secure data delivery.

Outline of the main material of the study. We formalize the intuitive notion that each person wakes up with a finite reserve of cognitive energy (E) and a fixed amount of time (T = 24h) for the day. Let E be a quantitative measure (e.g., in arbitrary units in this paper) of cognitive energy available at the start of the day. Actions (tasks, activities) draw from or replenish this energy. We define an action D_i by its duration Δt_i (part of T) and its energy cost or gain ΔE_i. Productive actions are those that move a person towards their goals (work, learning, creative tasks), often requiring significant energy but yielding progress or skill improvements. Unproductive actions might be necessary but do not yield long-term benefits (e.g., idle scrolling or avoidable distractions).

At any time t, the remaining energy $E(t)$ is impacted by preceding actions:

$$E(t) = E_0 + \sum_i \Delta E_i \quad (1)$$

where each productive action typically has

$\Delta E_i < 0$ – energy consumption,

$\Delta E_i \geq 0$ – restful or enjoyable actions might have (energy maintenance or recovery).

Our model aligns with the concept of ego depletion, where tasks requiring discipline deplete a common cognitive energy pool.

By categorizing each action D_i as productive if it contributes to personal or professional objectives (writing code, studying, exercising), or unproductive if it is a time filler or procrastination (excessive social media, aimless web browsing). Importantly, an unproductive action still consumes time T and may consume E (e.g., decision fatigue from context switching). Our later sections describe how transcribed audio can be analyzed and tagged to classify segments into these categories automatically, via NLP keyword spotting or context analysis.

A flow state action is a special productive action wherein the individual experiences heightened focus, enjoyment, and often loses sense of time. In our framework, flow state actions are productive actions with a drastically reduced perceived cost to E per unit time, due to deep intrinsic motivation and focus. In some cases, being in flow can even feel energizing (ΔE_i approaches zero or turns positive, as tasks "fuel" the person). We model flow by a parameter φ_i for each action,

with $\varphi = 1$ indicating normal state,

and $\varphi << 1$ indicating flow.

The effective energy cost in flow is scaled:

$$\Delta E_{i,\ effective} = \varphi_i \cdot \Delta E_i \quad (2)$$

For a flow-state task, φ might be 0.5 or lower (halving energy drain), consistent with studies showing dramatically increased productivity and delayed onset of fatigue.

Flow states are detected post-hoc by analyzing behavioral cues: extended periods of uninterrupted activity on a single task, higher speed of speech or coding, enthusiastic tone, etc. In our audio-based

approach, evidence of flow might include sustained monologues about a work topic, few pauses or distractions, and possibly keywords expressing excitement or deep concentration.

Hypothesis of Maximum Productivity

This hypothesis posits an optimal distribution of actions $D_{1...n}$ that maximizes output given constraints of E and T. It encourages prioritizing high-impact tasks when energy peaks and minimizing energy leakage on low-value tasks.

Our model incorporates the idea that managing energy is more important than managing time. Thus, the goal is not to be busy every minute, but to strategically allocate E and T to maximize productive outcomes and, ideally, achieve flow for the most important activities.

Data Capture and Ethical Considerations

We introduce Countable Cognitive Energy (CCE) as a quantitative estimate of energy expenditure and replenishment derived from the day's audio transcripts. Conceptually, CCE is computed by assigning energy costs (or gains) to each transcribed activity and summing them.

To compute CCE the continuous audio (approx. 16 hours of awake time) is transcribed to text via an automatic speech recognition (ASR) model (OpenAI Whisper in our case). This yields a chronological sequence of utterances with timestamps.

The transcript is segmented into distinct activities or contexts using cues like silence gaps, location changes (if available), or keywords. For example, segments might be "morning routine," "coding at work," "lunch with colleagues," "scrolling Twitter," "evening workout." Techniques such as unsupervised topic modeling or supervised classification with known activity keywords are used to label segments.

We maintain a knowledge base of typical energy costs for activities.

Focused work (coding, writing) – high energy cost per minute.

Social interaction (meeting, conversation) – moderate cost (can vary if the person is extroverted vs. introverted).

Passive leisure (watching TV, browsing) – low to moderate cost, possibly slight energy recovery if relaxing.

Rest (napping, meditation) – energy gain (negative cost).

Flow state instance of any above – reduce cost by factor φi.

CCE is the net sum of energy expenditures over all activity segments. We compute it as:

$$CME = \sum_{i \in segments} w_i \cdot \Delta t_i \quad (3)$$

where:

w_i is the energy cost rate of segment i (units of energy per minute);

Δt_i is the duration of segment i (in minutes).

If w_i is negative (i.e., an energy-replenishing activity), it contributes positively to the overall sum.

A large negative CCE (e.g., $-E_0$) indicates a fully draining day.

A smaller magnitude (e.g., $-0.7 \cdot E_0$) suggests only 70% of the day's energy reserve was used.

A positive CCE would imply net energy gain – rare, but possible on particularly restful days (e.g., meditation).

By converting qualitative transcript segments into a single quantitative metric, CCE enables day-to-day or person-to-person comparisons of cognitive workload. Accuracy depends on reliable transcript segmentation (e.g., via Whisper ASR) and correct assignment of w_i values. Silent but productive activities may be inferred contextually (e.g., long quiet coding sessions), and future extensions could integrate additional sensors (e.g., EEG) for more precise flow detection.

The designed system comprises hardware for audio capture and a cloud-based software pipeline for data processing and analysis. We describe each component and their integration. The wearable device (Fig. 1) is a small module attached to the user's shirt or as a pendant, also will be a custom circuit implementation instead of built-in solution in the II-phase of the research, which should include [6]:

Fig. 1. Wearable dictaphone

– Microphone with omnidirectional electret or MEMS mic to capture ambient sound and speech.

– Microcontroller (MCU) to manage audio sampling (e.g., 16 kHz for speech fidelity), optional compression, and storage writing. A low-power MCU with integrated ADC (analog-to-digital converter) can digitize audio. We use a microSD

or flash chip for ~8–16 GB storage, sufficient for ~1 week of audio if compressed.

– A rechargeable lithium battery (~200–300 mAh) provides power. Based on similar devices (Capture recorder used a 135 mAh battery), our design supports ~16 hours recording by using low-power modes when idle and efficient encoding (e.g., MP3 compression on the fly).

– Storage is offloaded via USB or Bluetooth daily. We opt for manual offload (user uploads nightly) to avoid constant wireless transmission (which would drain battery).

Users interact with their lifelog data through a React-based web application. The front-end serves two main functions.

1. "Upload interface" allows users to upload daily audio files from the wearable (likely after transferring from the device to a computer or phone). The UI shows upload progress and ensures the ~1.2 GB file is successfully sent. Large file upload is handled via chunking and resume support, given potential network issues.

2. "Visualization Dashboard" after processing, the web app displays transcripts, CCE metrics, and interactive visualizations. For example, a timeline highlights productive vs. unproductive segments in different colors, and pie charts summarize time spent in various categories. Flow state occurrences might be marked with a special icon. The React app fetches this data via API calls (discussed below) and uses libraries (like D3.js) for dynamic charts.

Keeping the front-end decoupled ensures responsiveness; heavy processing is done server-side. We introduce a .NET (ASP.NET Core) proxy API as a secure middle layer. This service serves as the gateway to Azure resources.

– It accepts file uploads from the React app and streams them to Azure Blob Storage, authenticating using Azure AD or SAS (Shared Access Signatures) tokens. The API can enforce rate limits (one file per day per user) and file size limits, returning appropriate errors for oversized or too frequent uploads.

– It exposes endpoints for retrieving processed results (transcripts, analytics). Instead of the React app hitting storage or databases directly, the .NET layer fetches data (from Blob) and returns it after authorization checks.

– This design addresses security: user authentication tokens are validated by the .NET API, and direct access to cloud data is prevented, reducing the risk of leaks.

Cloud Storage (Azure Blob) for scalable, durable storage of large audio and text files.

– Audio blob container has raw uploaded audio files land here. Each file is named by user and date (e.g., user123/2025-04-01.wav).

– Transcript container saves a text file or JSON with timestamps (e.g., user123/2025-04-01-transcript.json) after transcription.

– Results Container with final analysis outputs, such as a JSON containing segment classifications, CCE values, etc., are stored here (e.g., user123/2025-04-01-analysis.json).

Azure Blob automatically triggers events on new uploads. Specifically, we use Blob storage events to initiate Azure Functions in our pipeline (next subsection). Blob Storage offers virtually infinite scalability and cost-effective storage for large data, crucial for lifelogging where data accumulates quickly.

The core processing is orchestrated via Azure Functions, enabling an event-driven, scalable workflow.

1. Transcription function (triggered by "Audio Upload") is using a blob trigger on the audio container, this function runs when a new file is uploaded. It loads the audio (streaming directly from Blob to avoid local temp storage) and invokes the Whisper ASR model (which we package with the function or call a containerized service with GPU if available). The function generates a transcript and stores it as JSON in the transcript blob container. We include metadata like recognized language and a confidence score. Execution time is a bottleneck: Whisper's speed varies by model (tiny/large) and hardware (GPU vs CPU).

For ~16 h of audio, even a fast model on CPU might take multiple hours. To mitigate, we can use an Azure Batch Transcription service or optimize by splitting audio into chunks processed in parallel (Azure Functions can process chunks concurrently if we split the file and signal multiple triggers).

2. NLP analysis function is triggered by "Transcript Upload" when the transcript JSON is saved, another blob trigger fires. This function performs several NLP tasks:

– Detects segment boundaries via silence timestamps or topic shifts.

– Tags each segment as productive/unproductive using keywords or a classifier. For example, "code", "write", "design" suggest productive, whereas "scroll", "TV", "idle" suggest unproductive. If uncertain, default to unproductive to avoid overstating productivity.

– Flow detection via searches for long monologues or exclamations indicating high

engagement. Perhaps if the user is humming or talking excitedly about work for >30 minutes uninterrupted, mark flow.

– CCE computation using the classifications and known segment durations. Also computes subtotals (e.g., CCE from morning vs afternoon).

– Optionally generate a summary of the day (for user readability).

– Save a structured output (JSON with segments list and metrics) to the results container.

Software architecture diagram on Fig. 2 depicts the end-to-end system flow from wearable to cloud to user interface.

The wearable generates an audio file which the user uploads via the React app to the .NET API. The API stores it in Azure Blob (Audio File). The upload triggers the Whisper transcription function which saves a transcript to Blob (Transcript).

That triggers the NLP analysis function, which produces results saved to Blob (Results). The user can then fetch and visualize those results in the React app via the API. This loosely coupled, serverless architecture is resilient and scalable: each component can be updated independently, and Azure handles the heavy lifting of scaling storage and compute.

Azure Functions provide automatic scaling. If many users upload simultaneously, multiple instances of these functions spawn. ASR is resource

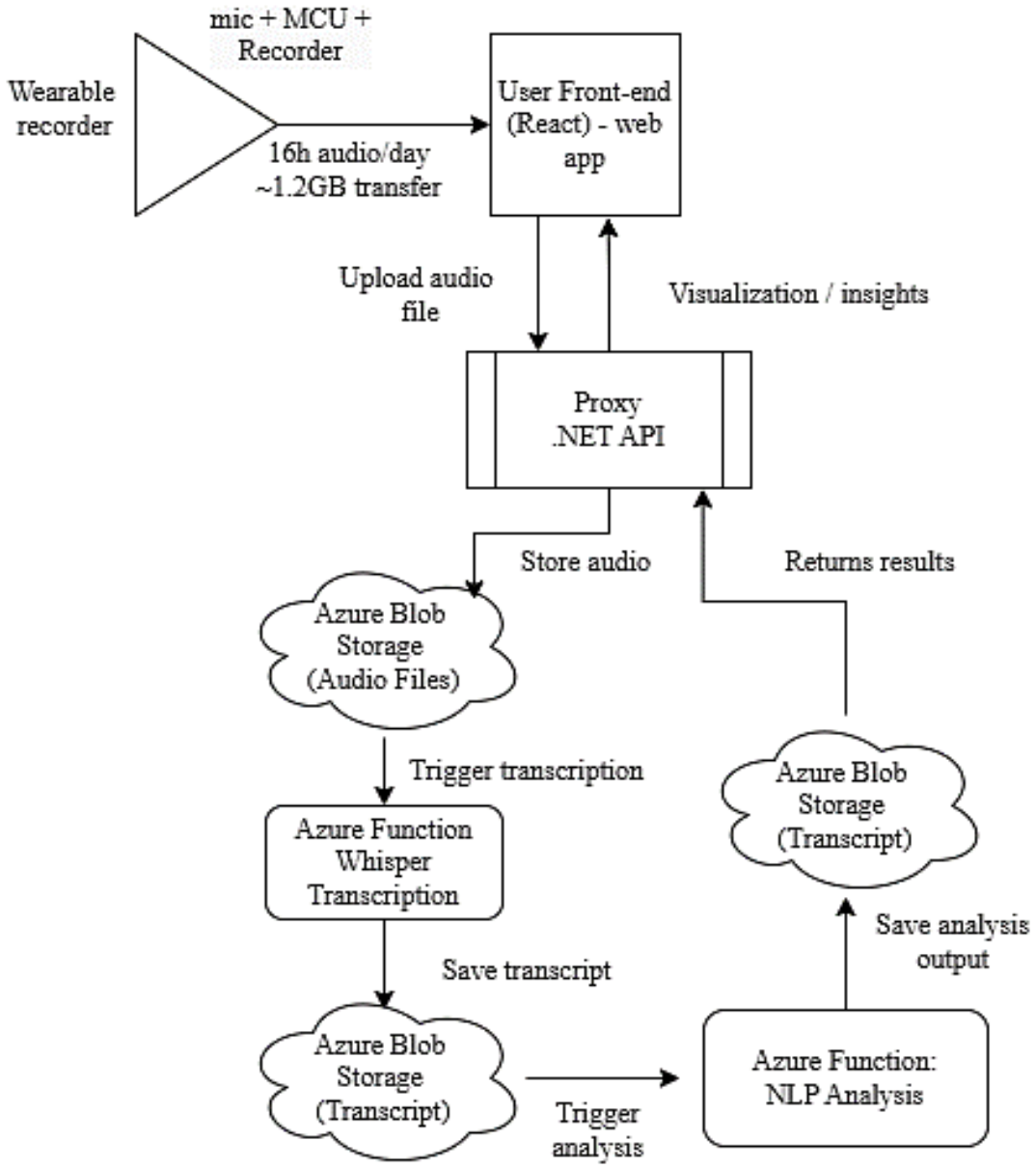

Fig. 2. Software system architecture data flow

heavy; a limitation is the availability of GPUs in Azure Functions.

If needed, we might offload transcription to Azure Cognitive Services for Speech to Text (batch transcription API), which provides a scalable managed service (at cost). Our architecture is flexible to swap in such services without changing the dataflow.

For the initial PoC dataset, the device was used for a 10-day period, ending up in 12 GB of raw audio data *.wav* and compressed into *.mp3* format respectively reducing size to ~1.6 GB and in total of 160 hours, and activity data was analyzed based on that.

The most used activities are described in Table 1, with rounded timing data for simplicity of the subjective analysis. Table presents various daily activities categorized by their duration, cognitive cost, and contribution to a cumulative cognitive effectiveness metric.

A positive CCE indicates net cognitive gain – reflecting the significant flow and recovery periods outweighing costly segments.

In a typical workday without flow or recovery breaks, we would expect a large negative CCE, signaling energy depletion.

Each row represents a specific activity segment, such as Morning Routine, Focused Coding, Deep Work, or Lunch Break.

Duration Δt_i *(min)* is time spent on the activity in minutes.

Base cost w_i *(units/min)* is a qualitative measure of cognitive load or benefit per minute, ranging from negative values (indicating recovery or relaxation) to high positive values (indicating high mental effort).

Flow φ_i is a multiplier representing how well the activity promotes a state of flow or engagement (1.0 being full flow).

Effective cost $\varphi_i \cdot w_i$ *(units/min)* is the product of flow and base cost, representing the actual mental effort or benefit during the activity.

CCE contribution (units) is the total contribution to cognitive effectiveness over the duration of the activity, calculated as duration × effective cost.

Conclusions. The importance of this research lies in its innovative integration of AI, software engineering, and human-computer interaction to create a practical and theoretical foundation for understanding and quantifying cognitive energy in daily life. By introducing Countable Cognitive Energy (CCE) as a measurable metric derived from transcribed audio, the study offers a novel lens to analyze productivity and mental effort across different activities.

The intelligent software framework enables continuous, passive data collection via wearable devices, combined with powerful backend processing pipelines using Whisper AI and natural language techniques. These components allow not only for the identification of high- and low-energy segments but also the detection of psychologically optimal «flow states.» This capability to visualize and assess daily cognitive patterns has implications for improving time management, optimizing work-rest cycles, and enhancing well-being. Furthermore, the system's modular architecture supports scalability and future extensions, such as integration with physiological data, making it a valuable foundation for both academic research and real-world productivity tools.

This research enables a form of productivity lifelogging, where not just events are recorded, but their qualitative impact on one's productive life is assessed.

The ability to visualize how one's energy ebbs and flows throughout the day, and to pinpoint pshycological flow state periods, could help users

Table 1

Activity segments

Segment	Duration Δt_i (min)	Base cost w_i (units/min)	Flow φ_i	Effective cost $\varphi_i \cdot w_i$ (units/min)	CCE contribution (units)
Morning Routine	30	–0.5 (recovery)	1.0	–0.5	–15
Focused Coding	120	1.0 (high cost)	0.6	0.6	72
Team Meeting	45	0.8 (moderate cost)	1.0	0.8	36
Email and Administrative	30	0.5 (low–moderate)	1.0	0.5	15
Social Media Break	15	0.3 (low cost)	1.0	0.3	4.5
Deep Work (Flow)	90	1.2 (very high)	0.5	0.6	54
Lunch Break	60	–0.2 (light recovery)	1.0	–0.2	–12
Commute (Podcasts)	45	0.2 (low engagement)	1.0	0.2	9
Reading (Leisure)	30	–0.1 (relaxing)	1.0	–0.1	–3

restructure their routines for maximal effectiveness and well-being.

It also opens research avenues: for example, correlating CCE with physiological signals (heart rate, cortisol) to validate the energy model, or determine how much energy is expended during an argument.

We plan to refine the NLP models for activity classification (possibly fine-tune a transformer on annotated lifelog data), implement a pilot with users to gather feedback on CCE's usefulness, and explore integration of additional sensors (like detecting computer usage or location to supplement context).

On the theory side, the framework could be expanded to account for different types of energy (cognitive vs physical), and how actions trade off between them. We believe this interdisciplinary approach – mixing theory, AI, and hardware – illustrates a path forward in personal productivity tools and lifelogging research.

Bibliography:

1. LENA Research Foundation. LENA Technology. URL: https://www.lena.org/ (date of access: 20.04.2025).
2. Shah M., Mears B., Chakrabarti C., Spanias A. Lifelogging: archival and retrieval of continuously recorded audio using wearable devices. *2012 IEEE International Conference on Emerging Signal Processing Applications*. 2012. P. 99–102. DOI: https://doi.org/10.1109/ESPA.2012.6152455.
3. Feng T., Nadarajan A., Vaz C., Booth B., Narayanan S. TILES audio recorder: an unobtrusive wearable solution to track audio activity. *Proceedings of the 4th ACM Workshop on Wearable Systems and Applications (WearSys '18)*. 2018. P. 33–38. DOI: https://doi.org/10.1145/3211960.3211975.
4. Csikszentmihalyi M. Flow: The Psychology of Optimal Experience. URL: https://surl.lu/wecjzl (date of access: 20.04.2025).
5. Baumeister R., Bratslavsky E., Muraven M., Tice D. Ego Depletion: Is the Active Self a Limited Resource? *Journal of Personality and Social Psychology*. 1998. № 74 (5). P. 1252–1265. DOI: https://pubmed.ncbi.nlm.nih.gov/9599441/.
6. AN278: Voice Recorder Reference Design. URL: https://www.silabs.com/documents/public/application-notes/AN278.pdf (date of access: 20.04.2025).

Вернік М.О., Олещенко Л.М. ІНТЕЛЕКТУАЛЬНА ПРОГРАМНА ПЛАТФОРМА ДЛЯ ОБРОБКИ ВЕЛИКИХ АУДІО ДАНИХ ТА ОЦІНЮВАННЯ КОГНІТИВНОЇ ЕНЕРГІЇ

Стаття присвячена представленню інноваційної програмної платформи для аналізу аудіо, згенерованого людиною, з метою кількісної оцінки її когнітивної енергії. Розроблено теоретичну модель, яка описує щоденну когнітивну енергію (E), час (T) та дії (D1...Dn), з виокремленням продуктивної та непродуктивної діяльності та визначенням особливого стану потоку. Запропоновано нову кількісну метрику – Countable Cognitive Energy (CCE), що обчислюється на основі транскрипцій аудіо щоденних завдань. Описано технічну архітектуру системи, яка передбачає безперервний запис аудіо за допомогою носимих пристроїв (~16 год/день, ~1,2 ГБ/день на користувача), транскрибування за допомогою Whisper AI, сегментацію та позначення для подальшого аналізу. Програмну систему реалізовано з використанням інтерфейсу на основі React для візуалізації та завантаження даних, Azure Blob Storage для масштабованого зберігання, Azure Functions для керування транскрипцією та обробкою мови, .NET API для керування доступом, а також серверної частини, написаної на Python для розпізнавання мовлення та обробки природної мови. Архітектуру розробленого програмного забезпечення зображено схемами потоків даних та мікросхемою пристрою. З'ясовано ключові обмеження запропонованої програмної системи, зокрема, тривалість транскрипції, обмежену доступність GPU і пропускну здатність мережі. Проаналізовано можливості масштабування системи у майбутньому. Проведено порівняння розробленої платформи із наявними програмними рішеннями (LENA, TILES Audio Recorder, Friend wearable) з акцентом на можливості візуалізації шаблонів продуктивності користувача та виявлення станів потоку. Проаналізовано відповідну літературу з тематики лайфлогінгу та продуктивності. У межах подальших досліджень передбачено вдосконалення NLP-класифікації дій, експериментальну перевірку корисності CCE за участі користувачів та інтеграцію додаткових контекстних сенсорів, таких як використання комп'ютера або геолокації. Також окреслено можливість розширення платформи з розділенням когнітивної та фізичної енергії для подальшого розвитку засобів продуктивності та лайфлогінгу.

Ключові слова: *програмна платформа, великі дані, аудіо, mp3, wav, когнітивна енергія, згенеровані людиною дані, диктофон, NLP, Whisper AI, Python, Azure Blob Storage.*