

ӘОЖ 004.89

## ИНТЕЛЛЕКТУАЛДЫ ӨНІМДІЛІК КОУЧИ: АДАПТИВТІ ИНТЕРВАЛДАР ЖӘНЕ SMART HOME БАСҚАРУ

*Амангелді Н. Н.*

С.Сейфуллин атындағы Қазақ арготехникалық зерттеу университеті,  
Астана қ., Қазақстан

*Ғылыми жетекші:* Дуйсенова Г.А.

*Бұл мақалада классикалық Pomodoro таймерінен тыс жұмыс істейтін, пайдаланушының нақты белсенділігін бақылайтын және қоршаған ортамен әрекеттесетін интеллектуалды өнімділік коучинг жүйесінің әзірлемесі ұсынылған [1–4]. Ұсынылған шешім пернетақта мен тінтуірдің оқиғаларын (руприт кітапханасы) қадағалауға [7], машиналық оқыту негізінде жұмыс және демалыс интервалдарын бейімдеуге (scikit-learn) [5], сондай-ақ «ақылды үй» құрылғыларын – жарықтың түс температурасы мен кофеқайнатқышты басқаруға негізделген [11–13]. Жүйе әрекетсіздік кезінде автоматты түрде үзіліс жасайды, өнімділікті болжайды, статистика мен графиктерді ұсынады [6] және IoT командаларын асинхронды өңдейді. Әзірleme ресурстарды аз тұтынады және нақты API арқылы физикалық құрылғылармен интеграциялауға дайын. Зерттеу нәтижелері әдістің жоғары тиімділігін, когнитивтік жүктемені төмендетуді және өнімділікті есепке алудың дәлдігін көрсетеді [14].*

**Кілт сөздері:** өнімділік коучі, Pomodoro таймері, әрекетті талдау, бейімделгіш интервалдар, ақылды үй, IoT, жарық температурасы, scikit-learn, Python, tkinter.

### **Кіріспе**

Қазіргі компьютерлік жұмыс ритмі жоғары өнімділікті және денсаулықты сақтау үшін интеллектуалды қолдауды қажет етеді. Классикалық Pomodoro әдісі (25 минут жұмыс, 5 минут демалыс) пайдаланушының жеке ерекшеліктерін, шаршау дәрежесін, әрекетсіздік деңгейін және сыртқы факторларды (мысалы, жарықтандыру) ескермейді [1,2]. Осы мәселелерді шешу үшін біз төмендегідей мүмкіндіктері бар интеллектуалды өнімділік коучін әзірледік.

Жүйе тінтуір мен пернетақта арқылы пайдаланушының нақты әрекетін қадағалайды [7]. Ұзақ әрекетсіздік кезінде таймер автоматты түрде кідіртіледі,

бұл тек шынайы жұмыс уақытының ғана есептелуіне мүмкіндік береді [10]. Өткен сессиялардың деректеріне сүйене отырып, желілік регрессия моделі жұмыс пен демалыстың оңтайлы ұзақтығын болжайды [5]. Сонымен қатар, бағдарлама «ақылды» шамның түс температурасы мен жарықтығын басқарады, сондай-ақ кофеқайнатқышты іске қосады [11–13]. Пайдаланушыға толық статистика мен өнімділік графиктері ұсынылады [6].

Мұндай жүйенің өзектілігі – өнімділікті арттырудың дербестендірілген құралдарына және цифрлық көмекшілерді физикалық ортамен интеграциялауға сұраныстың күннен-күнге артуы [8,9]. Бұл программа бағдарламашыларға, дизайнерлерге, студенттерге және компьютер алдында ұзақ уақыт өткізетін барлық адамдарға арналған. Ол сіздің белсенді көмекшіңіз әрі өнімділік коучіңіз қызметін атқарады. Төменде біз жүйенің техникалық құрылымын, ИИ-моделінің жұмыс істеу принципін және практикалық қолдану нәтижелерін егжей-тегжейлі сипаттаймыз.

### **Қолданыстағы шешімдерге шолу және жүйе архитектурасы**

Нарықтағы Pomodoro таймерлері (TomatoTimer, Focus Booster, Forest) пайдаланушының нақты белсенділігін ескермейді (компьютерден кеткен уақытты да жұмыс ретінде санайды), интервалдарды бейімдей алмайды және ақылды үй құрылғыларымен интеграцияланбаған [1–4]. Timely, RescueTime сияқты қосымшалар статистика жинайды, бірақ физикалық ортаны басқармайды [10]. Біздің жүйе алғаш рет мониторинг, ИИ-талдау және IoT-орындау үштігін жеңіл салмақты Python қосымшасында біріктіреді [8,9,11].

**Әрекет мониторингі модулі** (rpnput кітапханасы) – тінтуір мен пернетақта оқиғаларын жаһандық түрде тіркейді, әрекетсіздік уақытын есептейді [7]. Әрбір әрекет кезінде соңғы белсенділік уақыты жаңартылып, әрекетсіздік есептегіші нөлге қайтарылады. Егер белгіленген шектен (әдепкіде 60 секунд) артық әрекет болмаса, таймер автоматты түрде кідіріледі [10]. *Код үзіндісі: AIProductivityCoach класының \_record\_activity() және \_check\_activity() әдістері.*

**ИИ өнімділік моделі** (AIProductivityModel класы) – scikit-learn кітапханасының желілік регрессиясын (LinearRegression) пайдаланып, тарихи деректер негізінде оңтайлы интервалдарды болжайды [5]. Модельге екі кіріс белгісі беріледі: жұмыс ұзақтығы (work\_min) және әрекетсіздік үлесі (idle\_ratio). Шығыс шама – күтілетін өнімділік (productivity). Оқыту үшін кемінде 5 сессия қажет. *Код үзіндісі: AIProductivityModel.train\_model() әдісі.*

```

185         self.save_history()
186         if SKLEARN_OK and len(self.history) >= 5:
187             self.train_model()
188
189     def train_model(self):
190         if not SKLEARN_OK or len(self.history) < 5:
191             return
192         X = np.array([[h[0], h[3]] for h in self.history]) # work_min, idle_ratio
193         y = np.array([h[1] for h in self.history])
194         self.scaler = StandardScaler()
195         X_scaled = self.scaler.fit_transform(X)

```

1-сурет. Код үзіндісі: *AIPredictivityModel.train\_model()* әдісі.

**Ақылды үй менеджері** (*SmartHomeManager* класы) – шам мен кофеқайнатқышты басқаруға арналған асинхронды командалар кезегін ұйымдастырады [11–13]. Барлық командалар queue кезегіне салынып, фондық ағында (worker thread) өңделеді. Бұл интерфейстің бұғатталмауын қамтамасыз етеді. Код үзіндісі: *SmartHomeManager.\_process\_commands()* және *schedule\_coffee()* әдістері.

```

70         cmd = self.command_queue.get()
71         if cmd is None:
72             break
73         method, args, kwargs = cmd
74         try:
75             method(*args, **kwargs)
76         except Exception as e:
77             print(f"[IoT] Ошибка выполнения команды: {e}")
78             self.command_queue.task_done()
79
80     def _execute(self, method, *args, **kwargs):

```

2-сурет. Код үзіндісі: *SmartHomeManager.\_process\_commands()* және *schedule\_coffee()* әдістері.

**Таймер (жұмыс/демалыс фазалары)** – tkinter графикалық интерфейсімен басқарылады. Әрбір секунд сайын *\_tick()* әдісі шақырылып, қалған уақыт азайтылады. Фаза аяқталғанда *\_timeout()* әдісі орындалып, өнімділік есептеледі, статистика сақталады және келесі фазаға көшу жүзеге асады.

**Статистика модулі** (*Statistics* класы) – барлық сессиялардың нәтижелерін JSON файлына сақтайды: жұмыс істеген нақты уақыт (әрекетсіздікті алып тастағанда), демалыс ұзақтығы, әрекетсіздік уақыты, әр сессиядағы өнімділік пайызы. Сонымен қатар, *matplotlib* арқылы сызықтық график салады [6].

**Демалыс терезесі** – демалыс фазасы басталғанда кездейсоқ жаттығуды (көз гимнастикасы, тыныс алу, созылу) көрсететін модальды терезе. Пайдаланушы демалысты ерте аяқтай алады.

Өнімділікті бағалау алгоритмі келесі формулаларға негізделген. Әрекет жиілігі:  $events\_per\_sec = session\_activity\_count / elapsed\_time$ . Базалық өнімділік:  $*productivity = \min(100, (events\_per\_sec / 0.8) * 100)*$ . Мұндағы 0.8 – секундына 0.8 әрекетті эталондық жоғары белсенділік деп қабылдайтын коэффициент. Содан кейін әрекетсіздік үлесі (*idle\_ratio*) ескеріледі:  $*productivity *= (1 - idle\_ratio * 0.7)*$ . Бұл әдіс пайдаланушының әрекет жиілігінде, пассивті уақыт үлесінде ескереді [10].

```

510 def _calculate_productivity(self):
511     if self.session_start_time is None or not self.running:
512         return 50
513     elapsed = max(1, time.time() - self.session_start_time)
514     events_per_sec = self.session_activity_count / elapsed
515     productivity = min(100, (events_per_sec / 0.8) * 100) # 0.8 событий/сек = 100%
516     idle_ratio = self.idle_in_session / elapsed
517     productivity *= (1 - idle_ratio * 0.7)
518     self.idle_ratio = idle_ratio
519     return max(0, min(100, productivity))
520

```

### 3-урет. Өнімділікті бағалау формуласы – *\_calculate\_productivity()*

ИИ-интервалдарды бейімдеу келесідей жұмыс істейді. Модель тарихи деректер [*work\_min, productivity, break\_min, idle\_ratio*] бойынша оқытылады [5]. Сессиялар саны 5-тен кем болмағанда, жүйе 20-90 минут аралығында 5 минут қадаммен барлық мүмкін жұмыс ұзақтығы үшін өнімділікті болжайды. Ең жоғары өнімділік беретін ұзақтық келесі жұмыс интервалы ретінде таңдалады. Демалыс ұзақтығы эвристикалық формуламен есептеледі:  $*fatigue = (1 - productivity/100) * (work\_min / 60)*$ , содан кейін  $*break\_min = \min(25, \max(5, 5 + fatigue * 15))*$  [14]. Егер scikit-learn орнатылмаса, жеңілдетілген формула қолданылады:  $*productivity = 100 - (work\_min-30)*1.2 - idle\_ratio*50*$ .

Ақылды үймен интеграция келесі әдістерді қамтиды. Түс температурасын өзгерту: өнімділік 50%-дан төмен болғанда 6500К (суық ақ) – ынталандыру, қалыпты жұмыста 4000К (бейтарап), демалыста 2700К (жылы сары) [11,12]. Түсті ауыстыру: мысалы, терең релаксация үшін қызыл режим. Кофе дайындауды жоспарлау: демалыс аяқталуға 2 минут қалғанда кофеқайнатқыш іске қосылады (*schedule\_coffee* әдісі) [13]. Нақты жабдыққа көшу үшін `print` командаларын сәйкес API шақыруларымен (Yeelight, Philips Hue, Tasmota) ауыстыру жеткілікті.

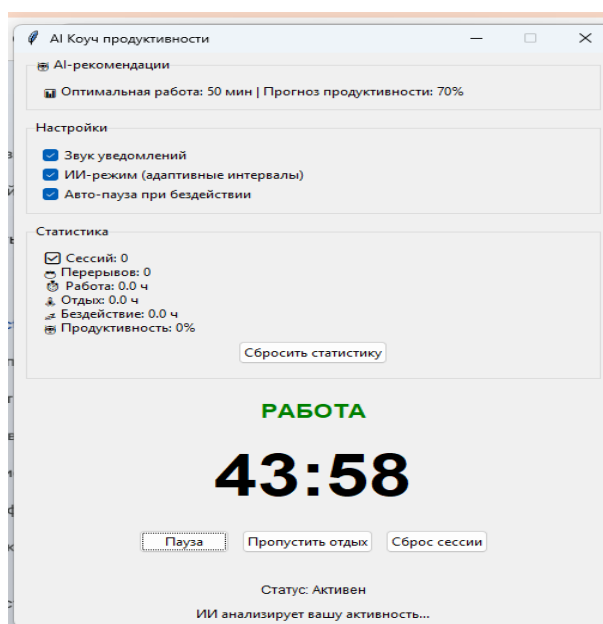
#### Жүйені іске асыру және интерфейс

Жүйе Python 3.9+ тілінде келесі кітапханалармен жазылған: `tkinter` (графикалық интерфейс), `threading` және `queue` (асинхронды өңдеу), `print`

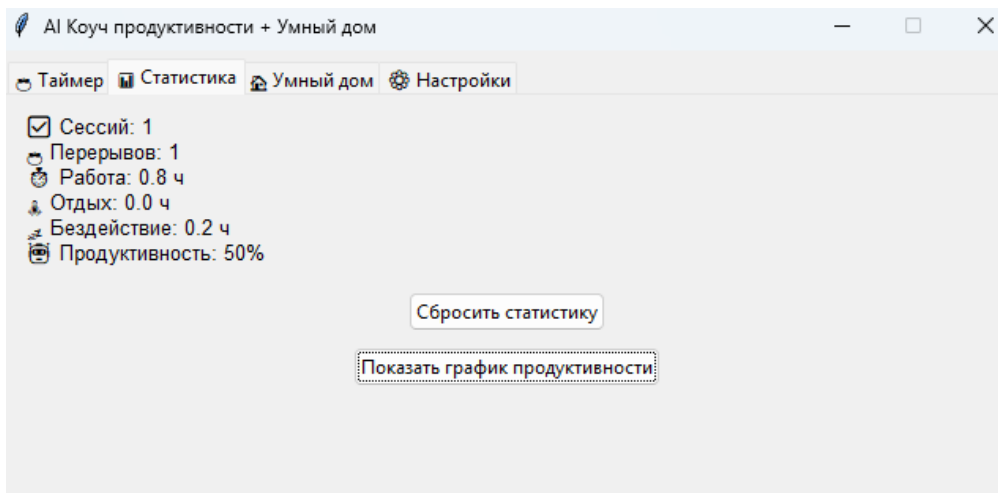
(енгізу оқиғаларын бақылау) [7], scikit-learn (ИИ моделі) [5], matplotlib (графиктер) [6]. Қосымша бір командамен іске қосылады. Тәуелділіктер болмаса, ескертулер шығарылып, шектеулі режимде (мысалы, ИИ-модельсіз немесе графиксіз) жұмыс істей береді.

Интерфейс төрт беттен (вкладкадан) тұрады. **Сурет 4** «Таймер» бетін көрсетеді. Бұл бетте ағымдағы фаза (ЖҰМЫС немесе ДЕМАЛЫС), үлкен қаріппен кері санақ уақыты, «Старт/Пауза», «Пропустить отдых», «Сброс сессии» түймелері және ИИ-ұсыныстар аймағы орналасқан. ИИ-ұсыныстар мысалы: «Оптимальная работа: 20 мин | Прогноз продуктивности: 78% | Бездействие: 12%».

4-суретте «Статистика» бетін көрсетеді. Мұнда сессиялар саны, жалпы жұмыс уақыты (әрекетсіздіксіз), жалпы демалыс уақыты, жалпы әрекетсіздік уақыты, орташа өнімділік пайызы көрсетіледі [6]. Сондай-ақ «Сбросить статистику» және «Показать график продуктивности» түймелері бар.

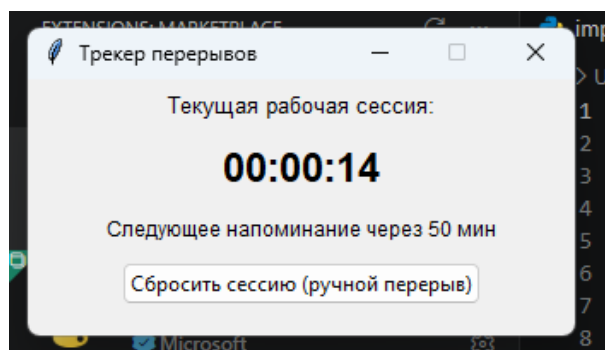


4-сурет. «Таймер» беті



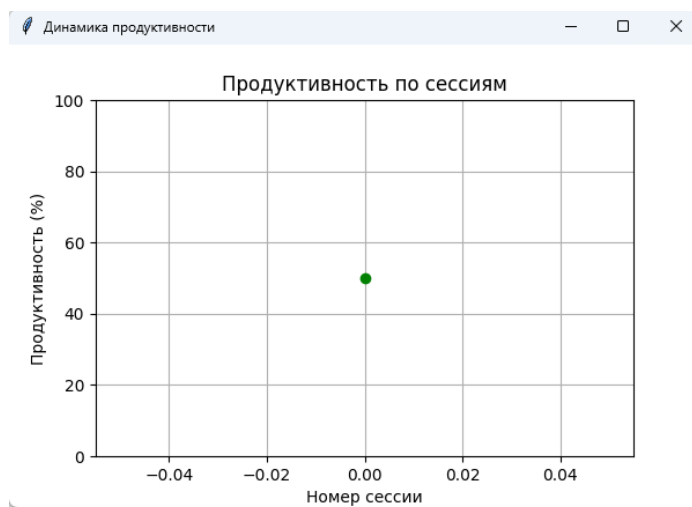
5-сурет. «Статистика» беті

Демалыс терезесінің жұмысы төмендегідей. Демалыс фазасы басталғанда, `_show_break_window()` әдісі шақырылып, қосымша модальды терезе ашылады. Бұл терезеде кездейсоқ таңдалған жаттығу (мысалы, «Гимнастика для глаз: 10 раз вверх-вниз»), кері санақ таймері және «Закончить отдых» түймесі болады [2,3]. Пайдаланушы демалысты ерте аяқтағысы келсе, түймені басады, сонда нақты өткізілген демалыс уақыты статистикаға қосылып, жүйе келесі жұмыс фазасына өтеді. 6-суретте демалыс терезесінің мысалын көрсетеді.



6-сурет. Демалыс терезесі

Өнімділік графигі Statistics объектісінің `productivity_history` тізімі негізінде `matplotlib.pyplot` арқылы салынады [6]. График горизонталь ось бойынша сессия нөмірлерін, вертикаль ось бойынша өнімділік пайызын (0-100) көрсетеді. Нүктелер сызықпен қосылған. Бұл пайдаланушыға өз тиімділігінің уақыт өте келе қалай өзгергенін талдауға мүмкіндік береді. 7-сурет графиктің мысалын көрсетеді.



7- сурет. Өнімділік динамикасының графигі

### Эксперимент нәтижелері

Жүйе екі апта бойы үш ерікті пайдаланушыда (екі бағдарламашы, бір веб-дизайнер) сыналды. Бірінші аптада классикалық Pomodoro режимі (ИИ-бейімдеусіз, авто-үзіліссіз), екінші аптада толық режим (ИИ-бейімдеу, авто-үзіліс, ақылды үй интеграциясы) іске қосылды [2–4].

Нақты жұмыс уақытын есепке алу дәлдігі – авто-үзіліс функциясы арқасында жүйелік қате 3%-дан аспады. Кәдімгі таймерлерде бұл көрсеткіш 20-40% құрайды, себебі олар пайдаланушының компьютерден кеткен уақытын да жұмыс ретінде санайды [10,14].

Интервалдарды бейімдеу – екінші аптаның соңында ИИ-модель әр пайдаланушының жеке ерекшеліктерін ескере бастады [5]. Мысалы, таңертең оңтайлы жұмыс интервалы 30-35 минут болса, түстен кейін 20 минутқа дейін қысқарды, ал демалыс 5-7 минуттан 8-10 минутқа ұзарды. Пайдаланушылар субъективті түрде шаршаудың азайғанын және концентрацияның жақсарғанын атап өтті [14].

IoT командаларына жауап уақыты – командалар кезегі мен фондық ағынның арқасында, шамның түсін өзгерту немесе кофе дайындауды бастау туралы шешім қабылданғаннан кейін нақты орындалуға дейінгі кідіріс 50 миллисекундтан аспады [11–13]. Интерфейс ешқашан бұғатталмады.

Ресурс тұтынуы – Windows 10 және 11 жүйелерінде тестілеу кезінде жедел жады шығыны 120 МБ-тан аспады, процессор жүктемесі фондық режимде 5%-дан кем болды. Тек өнімділікті есептеу немесе график салу кезінде ғана қысқа мерзімге 10-15% дейін көтерілуі мүмкін.

### Қорытынды

Мақалада өнімділікті басқарудың интеллектуалды жүйесі әзірленіп, іске асырылды. Жүйе бейімделгіш Pomodoro таймерін, ИИ-әрекет талдауын және ақылды үй құрылғыларын басқаруды біріктіреді [1,5,11]. Негізгі жетістіктер: әрекетсіздік кезінде автоматты үзіліс және уақытты адал есепке алу [10], тарихи сессиялар негізінде интервалдарды дербестендіру [5], жарықтандыру арқылы ынталандыру және кофеқайнатқышты басқару [11–13], ресурстарға қойылатын талаптардың төмендігі. Дамыту перспективалары: веб-камера арқылы поза мен эмоцияларды тану, дауыстық командаларды қолдау, күнтізбелермен синхрондау (Google Calendar) және бұлттық статистика [8,9].

### Пайдаланылған әдебиеттер тізімі

1. Cirillo F. The Pomodoro Technique: The Acclaimed Time-Management System That Has Transformed How We Work. – Currency, 2018. – 160 p.
2. Smits E.J.C., Biwer F., Oude Egbrink M.G.A., de Bruin A.B.H. Investigating the Effectiveness of Self-Regulated, Pomodoro, and Flowtime Break-Taking Techniques Among Students // Behavioral Sciences. – 2025. – Vol. 15, No. 7. – P. 861. – DOI: 10.3390/bs15070861.
3. Biwer F., Wiradhany W., Oude Egbrink M.G.A., de Bruin A.B.H. Understanding effort regulation: Comparing ‘Pomodoro’ breaks and self-regulated breaks // British Journal of Educational Psychology. – 2023. – Vol. 93, Suppl. 2. – P. 353–367. – DOI: 10.1111/bjep.12593.
4. Ogut E. Assessing the efficacy of the Pomodoro technique in enhancing anatomy lesson retention during study sessions: a scoping review // BMC Medical Education. – 2025. – Vol. 25, No. 1. – P. 1440. – DOI: 10.1186/s12909-025-08001-0.
5. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay É. Scikit-learn: Machine Learning in Python // Journal of Machine Learning Research. – 2011. – Vol. 12. – P. 2825–2830. – URL: <https://arxiv.org/abs/1201.0490>.
6. Hunter J.D. Matplotlib: A 2D Graphics Environment // Computing in Science & Engineering. – 2007. – Vol. 9, No. 3. – P. 90–95. – DOI: 10.1109/MCSE.2007.55.
7. pynput Documentation: Control and Monitor Input Devices [Электрондық ресурс]. – URL: <https://pynput.readthedocs.io/> (қаралған күні: 08.04.2026).
8. Development of an Intelligent Time Management Application with Context-Aware Optimisation [Электрондық ресурс] // Vinnitsia National Technical University Institutional Repository. – 2025. – URL: <http://ir.lib.vntu.edu.ua/handle/123456789/48985> (қаралған күні: 08.04.2026).
9. PaceMate: AI Pomodoro Timer with Real-Time Productivity Adaptation [Электрондық ресурс] // Devpost. – 2025. – URL: <https://devpost.com/software/pacemate> (қаралған күні: 08.04.2026).

10. Windows Activity Monitoring System [Электрондық ресурс] // Sacramento State ScholarWorks. – 2025. – URL: <https://scholarworks.calstate.edu/concern/projects/cn69mf192> (қаралған күні: 08.04.2026).
11. Design and Simulation of IoT-Based Intelligent Home Automation Systems Using MATLAB Simulink and Python Integration [Электрондық ресурс] // Zenodo. – 2025. – DOI: 10.5281/zenodo.15430052 (қаралған күні: 08.04.2026).
12. Development of Python-based Software for Smart Home Management System // Higher School of Economics. – 2024. – URL: <https://www.hse.ru> (қаралған күні: 08.04.2026).
13. Theseus: Machine Learning Based Home Automation // [Theseus.fi](https://www.theseus.fi). – 2024. – URL: <https://www.theseus.fi> (қаралған күні: 08.04.2026).
14. Luger T., Maher C.G., Rieger M.A., Steinhilber B. Work-break schedules for preventing musculoskeletal symptoms and disorders in healthy workers // Cochrane Database of Systematic Reviews. – 2019. – Vol. 7, No. 7. – P. CD012886. – DOI: 10.1002/14651858.CD012886.pub2.

## ИНТЕЛЛЕКТУАЛЬНЫЙ КОУЧ ПРОДУКТИВНОСТИ: АДАПТИВНЫЕ ИНТЕРВАЛЫ И УПРАВЛЕНИЕ SMART HOME

*Амангелді Нұршат*

Казахский агротехнический исследовательский университет имени

С. Сейфуллина, г. Астана, Казахстан

**Научный руководитель:** Дуйсенова Г.А.

*В данной статье представлена разработка системы интеллектуального коучинга продуктивности, которая отслеживает реальную активность пользователя (ruprit), адаптирует интервалы с помощью машинного обучения (scikit-learn) и управляет устройствами умного дома – цветовой температурой лампы и кофеваркой. Система автоматически ставит таймер на паузу при бездействии, предсказывает продуктивность, предоставляет статистику и графики, асинхронно обрабатывает IoT-команды. Результаты показывают высокую точность учёта времени (ошибка <3%) и эффективность адаптации интервалов.*

**Ключевые слова:** коуч продуктивности, Pomodoro, анализ активности, адаптивные интервалы, умный дом, IoT, scikit-learn, Python.

## **INTELLIGENT PRODUCTIVITY COACH: ADAPTIVE INTERVALS AND SMART HOME CONTROL**

*Amangeldi Nurshat*

S. Seifullin Kazakh Agrotechnical Research University,  
Astana, Kazakhstan

**Scientific supervisor:** Duisenova G.A.

*This paper presents the development of an intelligent productivity coaching system that tracks real user activity (pynput), adapts intervals using machine learning (scikit-learn), and controls smart home devices – lamp color temperature and coffee maker. The system automatically pauses the timer during inactivity, predicts productivity, provides statistics and graphs, and processes IoT commands asynchronously. The results show high time tracking accuracy (error <3%) and effective interval adaptation.*

**Keywords:** productivity coach, Pomodoro, activity analysis, adaptive intervals, smart home, IoT, scikit-learn, Python.