

А. С. КОМІН

аспірант кафедри інформаційних технологій
Сумський державний університет
ORCID: 0000-0003-0328-3934

О. В. БОЙКО

кандидат технічних наук, доцент,
старший викладач кафедри інформаційних технологій
Сумський державний університет
ORCID: 0000-0001-8557-2267

ПРОГНОЗУВАННЯ ЕНЕРГОСПОЖИВАННЯ У МІКРОМЕРЕЖАХ: ОЦІНКА КОНВЕРТОВАНИХ МОДЕЛЕЙ

У статті розглядається можливість застосування моделей машинного навчання на мобільних пристроях для прогнозування рівня споживання електроенергії у мікромережах. Дані пристрої здатні забезпечити додатковий захист при обробці приватних даних, але мають суттєвий перелік обмежень, що вимагає додаткової оцінки як способу підготовки моделей, так і способу їх запуску.

У даній роботі проведено оцінку ефективності моделей прогнозування рівня споживання електроенергії на базі LSTM після їх конвертації у мобільні формати CoreML, Tensorflow Lite для подальшого використання у якості частини підсистеми прогнозування на периферійних мобільних пристроях. Для навчання та оцінки ефективності моделей було використано набори даних двох категорій споживачів: промислового підприємства закритого типу та малого об'єкта цивільної інфраструктури. Для розробки тестових прототипів моделей було використано програмний пакет Tensorflow з подальшою конвертацією даних прототипів у формати, що дозволяють їх запуск на мобільних пристроях Apple. Конвертовані моделі були оцінені за розміром, точністю прогнозування, швидкістю надання прогнозу, кількістю споживаної оперативної пам'яті, впливом на нагрів пристрою та навантаженням на його центральний процесор.

В результаті оцінки було зроблено висновок, що втрата точності прогнозування після конвертації є незначною, а продуктивність конвертованих моделей дозволяє виконувати прогнозування в режимі реального часу з допустимим значенням рівня використання обчислювальних ресурсів пристрою. Даний результат підтверджує можливість використання мобільних пристроїв у якості периферійного засобу обчислення у підсистемі прогнозування рівня споживання енергії.

Ключові слова: периферійні обчислення, машинне навчання, розумні мікромережі, прогнозування споживання електроенергії, Tensorflow Lite, CoreML, мобільний додаток.

A. S. KOMIN

Postgraduate Student at the Department of Information Technologies
Sumy State University
ORCID: 0000-0003-0328-3934

O. V. BOIKO

PhD in Technical Sciences, Associate Professor,
Senior Lecturer at the Department of Information Technologies
Sumy State University
ORCID: 0000-0001-8557-2267

MOBILE ENERGY CONSUMPTION FORECASTING IN MICROGRIDS: EVALUATION OF CONVERTED MODELS

The article examines the feasibility of applying machine learning models on mobile devices for forecasting energy consumption levels in microgrids. These devices can provide additional security when processing private data but have significant limitations that require further evaluation of both the model preparation process and execution methods.

This study assesses the performance of LSTM-based energy consumption forecasting models after their conversion into mobile formats—CoreML and TensorFlow Lite—for further use as part of a forecasting subsystem on edge mobile devices. To train and evaluate model performance, datasets from two categories of consumers were used: a closed-type industrial enterprise and a small civil infrastructure facility. The development of test model prototypes was carried out using the TensorFlow framework, followed by conversion into formats suitable for execution on Apple mobile devices.

The converted models were evaluated based on their size, forecasting accuracy, inference speed, RAM consumption, impact on device heating, and CPU load. The evaluation results indicate that the loss of forecasting accuracy after conversion is minimal, and the performance of the converted models allows real-time forecasting with an acceptable level of computational resource usage. This result confirms the feasibility of using mobile devices as edge computing units in energy consumption forecasting subsystems.

Key words: edge computing, machine learning, microgrids, energy consumption forecasting, TensorFlow Lite, CoreML, mobile application.

Постановка проблеми

Для забезпечення зростаючого попиту на електроенергію важливо проводити модернізацію та вдосконалення існуючих енергетичних мереж. Інтеграція розумних мікромереж (ММ) у великі централізовані мережі дозволить покращити їх стійкість до форс-мажорів, краще балансувати навантаження та залучати генерацію на основі відновлюваних джерел енергії [1] conventional methods are being used, including fossil fuels. This in turn leads to greenhouse gas emissions (e.g., carbon dioxide or CO₂). Однак кожна ММ є унікальною з точки зору географічного розташування, параметрів обладнання, об'єму генерації та споживання, що робить розробку систем управління даними мережами нетривіальною задачею.

Залучення джерел генерації на основі відновлюваної енергії, таких як сонячна енергія, енергія вітру тощо, вносить великий рівень невизначеності у показники рівня генерації ММ, що ускладнює завдання вибору режиму їх експлуатації. Відповідно системи управління ММ потребують наявності підсистеми прогнозування споживання та генерації електроенергії для прийняття рішень по забезпеченню змінного попиту на електроенергію із збереженням заданих показників економічної ефективності [2].

Моделі машинного навчання (ML) на базі LSTM здатні надавати точні короткострокові прогнози рівня споживання електроенергії для різного типу споживачів на основі історичних даних про споживання, що може бути використано для проектування підсистеми прогнозування системи управління ММ [3]. Попри високу ефективність моделей ML їх навчання та використання потребує значних обчислювальних ресурсів. Інтенсивне використання моделей ML за останні роки спричинило значний ріст споживання електроенергії [4] та викидів діоксиду вуглецю дата центрами у США: більше 4 % від загального споживання та 2.18 % загальних викидів у 2023 р. відповідно [5]. Враховуючи унікальність кожної ММ моделі, прогнозування також є унікальними, і питання навчання та розміщення таких моделей, що буде економічно та енергетично ефективним, є актуальним.

Обладнання для ММ, таке як розумні лічильники, сенсори, датчики, генерує велику кількість даних, які можуть бути використані для прийняття рішень по управлінню. Однак такі дані можуть містити персональну інформацію про користувачів ММ, наприклад, шаблони та обсяги використання електроенергії. У більшості країн є законодавчі вимоги щодо правил зберігання, передачі та обробки приватної інформації [6], порушення яких може призводити до значних штрафів. Відповідно для зменшення ризику витоку таких даних та зменшення витрат на підтримку інфраструктури для її обробки та зберігання доцільно використовувати периферійні обчислення на рівні самих ММ. Сучасні смартфони мають підтримку шифрування даних, захист доступу кодом-паролем або біометричною автентифікацією, майже постійно знаходяться із власником, а отже є гарними кандидатами для обробки персональних даних. З іншого боку мобільні пристрої є дуже різноманітними як з точки зору апаратної, так і програмної частини, мають обмежену обчислювальну потужність, ємність акумулятора та об'єм пам'яті. При оцінці смартфона як пристрою для периферійних обчислень у підсистемі прогнозування споживання енергії ММ необхідно враховувати ці фактори.

Оскільки короткострокові прогнози споживання енергії використовуються для підтримки операційних рішень по управлінню ММ, критично важливо забезпечити високу точність прогнозів. Для запуску моделей ML на мобільних пристроях їх потрібно конвертувати у мобільні формати, які часто мають обмежену підтримку доступних операторів, шарів та типів даних [7, 8]. Це може призводити до втрати точності та продуктивності моделі, тому конвертовані моделі необхідно оцінювати повторно.

Мобільні операційні системи, такі як iOS, обмежують ресурси (CPU, RAM, акумулятор), що доступні для кожного додатку [9]. Щоб уникнути перевищення лімітів і примусового закриття додатку необхідно оцінити рівень навантаження на пристрій моделлю прогнозування. Також важливо забезпечити високу швидкість надання прогнозів, оскільки підсистема має працювати в режимі реального часу.

Обмежений обсяг пам'яті мобільних пристроїв і специфіка їх використання вимагає мінімізації розміру моделі. Також розмір моделі впливає і на об'єм оперативної пам'яті, необхідний для її запуску. Крім того, мобільні пристрої мають обмежений доступ до мережних ресурсів, а необхідність оновлення моделі з часом роблять її компактність ще важливішою.

Аналіз останніх досліджень і публікацій

У статті [10] розглядається важливість розробки ефективних систем управління ММ, описано оптимізаційні цілі та методи їх досягнення з використанням різних алгоритмів, у тому числі із застосуванням моделей ML

у задачах прогнозування. У роботі [3] розглянуто актуальні методи прогнозування рівня споживання електроенергії ММ для 3 типів споживачів і виділено моделі LSMT як найбільш ефективні.

Розвиток як програмних, так і апаратних технологій зробив можливим запуск моделей ML на малопотужних, енергоефективних пристроях, що відкриває нові можливості для периферійних обчислень, але і вимагає особливої уваги до підготовки та оптимізації даних моделей [11].

Формулювання мети дослідження

Метою цього дослідження є оцінка ефективності конвертації у мобільний формат та адаптації моделей короткострокового прогнозування рівня споживання електроенергії для споживчого сектора та промисловості для подальшого запуску на мобільних пристроях. Дослідження аналізує вплив вибору фреймворку конвертації моделі на її продуктивність та ефективність прогнозів, причини відмінностей у продуктивності та труднощі інтеграції цих фреймворків у мобільні додатки.

Викладення основного матеріалу дослідження

Методологія дослідження

Кожен споживач електричної енергії є унікальним з точки зору обсягів і графіків споживання, але загалом їх можна розділити на три основні групи: домогосподарства, об'єкти соціальної інфраструктури та промислові споживачі [3]. Щоб забезпечити ефективність моделі короткострокового прогнозування споживання енергії для всіх цих типів споживачів, необхідно навчати та оцінювати її на основі наборів даних від кожної групи. У цій роботі для цього використовуються набори даних щогодинного споживання енергії автозаправної станції та молокозаводу, що відповідають другій та третій групі споживачів з раніше зазначених.

Платформа iOS обрана для проведення експерименту, оскільки пристрої iOS є менш різноманітними з точки зору апаратних компонентів порівняно з пристроями на Android, що забезпечує стабільніші показники продуктивності та точності прогнозів. Мова програмування Swift та бібліотека графічного інтерфейсу SwiftUI дозволяють швидко розробити прототип мобільного додатку для тестування, а бібліотека Charts дозволяє візуалізувати результати прогнозу у вигляді графіків.

Одними з найефективніших моделей для прогнозування споживання енергії є моделі на основі LSTM [3]. Платформа TensorFlow дозволяє швидко виконати прототипування та навчання комплексних моделей ML, що складаються з декількох шарів, і можуть включати шари на основі рекурентних нейронних мереж (RNN), таких як LSTM. Також ця платформа дозволяє конвертувати повноформатні ML-моделі у мобільний формат TensorFlow Lite за допомогою вбудованого конвертера. Цей формат може використовуватися в мобільних додатках для iOS, Android, а також на малопотужних пристроях, побудованих на мікроконтролерах, таких як ARM Cortex та ESP32 [12], що робить його гарним кандидатом для використання у якості формату ML-моделей для периферійних обчислень.

TensorFlow Lite дозволяє виконувати обчислення на CPU, GPU та NPU (Neural Processing Unit) мобільних пристроїв, які доступні в деяких просунутих моделях пристроїв Android та iOS, однак конвертовані моделі підтримують лише обмежений набір операцій TensorFlow [7], що може призводити до зниження точності моделей порівняно з оригінальними. TensorFlow Lite був обраний як перший формат мобільних ML-моделей для тестування.

Apple має власний формат ML-моделей під назвою CoreML, який працює на пристроях з MacOS та iOS. Цей формат оптимізований для пристроїв екосистеми Apple і може використовувати CPU, GPU та Neural Engine [13] (доступний у мобільних чіпах починаючи з A11 Bionic) для обчислень, мінімізуючи використання пам'яті та споживання енергії. Apple надає бібліотеку CoreMLTools для конвертації сторонніх моделей у формат CoreML [14]. Цей формат був обраний як другий мобільний формат ML-моделей для тестування.

Для проведення експерименту було розроблено 2 прототипи моделей прогнозування споживання енергії на основі LSTM за допомогою TensorFlow Keras, що аналогічні до моделей представлених у роботі [3]. Навчання моделей було проведено з використанням двох наборів даних, які представляють дві групи споживачів: цивільну інфраструктуру та промислове підприємство. Моделі приймають щогодинні показники рівня споживання електроенергії за попередні 24 години як вхідні дані для прогнозування рівня споживання на наступну годину. Навчені моделі були конвертовані у формат TensorFlow Lite за допомогою вбудованого в Tensorflow конвертору та у формат CoreML за допомогою CoreMLTools. Вхідний та вихідний формати даних для конвертованих моделей були встановлені як Float32, щоб зберегти точність обчислень якомога ближчою до оригінальної моделі (Float64). Обидва конвертери використовували рівень оптимізації моделей за замовчуванням.

Для порівняння точності результатів прогнозування початкових та конвертованих моделей було використано оцінку середньоквадратичної помилки (RMSE):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|}, \quad (1)$$

де y_i – реальне значення часової серії у момент i ; $\hat{y}_i$ – прогнозоване значення часової серії у момент i ; n – загальна кількість прогнозованих значень

Відносна зміна точності прогнозування буде розрахована як:

$$RMSE_{\Delta} = \frac{RMSE_{\text{mod}} - RMSE_0}{RMSE_0}, \quad (2)$$

де $RMSE_0$ – значення середньоквадратичної помилки до конвертації моделі; $RMSE_{\text{mod}}$ – значення середньоквадратичної помилки після конвертації моделі

Розміри конвертованих моделей вимірювалися як загальний обсяг дискового простору, який займає модель разом із включеними коефіцієнтами ваги. Для TensorFlow Lite це розмір файлу.tflite, а для CoreML – розмір пакету файлів.mlpackage.

Для вимірювання використання оперативної пам'яті (RAM) конвертованими ML-моделями було використано інструменти профілювання Xcode. Вимірювання проводилися після завантаження моделі мобільним додатком та виконання прогнозів на тестових даних.

Для вимірювання рівня використання центрального процесору (CPU) було використано інструмент профілювання CPU Usage з набору інструментів профілювання Xcode. Діапазон використання 0–100 % визначимо як низький, 100–200 % як середній і більше 200 % як високий.

Для вимірювання впливу використання моделей на нагрів пристрою (TS) було використано інструмент профілювання Thermal State з набору інструментів профілювання Xcode. Інструмент показує значення від найменшого до найбільшого – nominal, fair, serious, critical. Значення critical може призводити до примусового завершення роботи додатку системою iOS.

Швидкість прогнозування конвертованих моделей вимірювалася як середня тривалість прогнозу на тестових даних:

$$t_{\text{avg}} = \frac{\sum_{i=1}^n t_i}{n}, \quad (3)$$

де t_i – час на виконання прогнозу для елемента i тестової часової серії; n – загальна кількість прогнозованих значень

Оцінка ефективності конвертації моделей

Для розробки та оцінки ефективності тестових моделей після їх конвертації у мобільні формати було використано набори даних.

Перший набір даних містить щогодинні дані про споживання енергії станцією АЗС протягом 26 місяців. У наборі даних відсутні 0,02 % значень, а також присутні аномалії, що були виявлені за допомогою моделі Isolation Forest [15]. Відсутні та аномальні значення були заповнені або замінені з використання функції лінійної інтерполяції з бібліотеки Pandas. Приклад даних із цього набору наведено у таблиці 1, візуалізацію набору даних наведено на рисунку 1.

Таблиця 1

Приклад даних з набору даних АЗС

Datetime	Consumption, кВт\год	length_of_day, сек	hour
2010-11-01 00:00:00	29	36190	0
2006-12-16 01:00:00	60	36190	1
2006-12-16 02:00:00	61	36190	2
2006-12-16 03:00:00	61	36190	3
2006-12-16 04:00:00	61	36190	4

Другий набір даних містить інформацію про споживання електроенергії молокозаводом з інтервалом у 30 хвилин протягом 20 місяців. У наборі даних відсутні 2,42 % значень, а також присутні аномалії, що були виявлені за допомогою моделі Isolation Forest. Відсутні та аномальні значення були заповнені або замінені з використання функції лінійної інтерполяції з бібліотеки Pandas. Приклад даних із цього набору наведено у таблиці 2, візуалізацію набору даних наведено на рисунку 2.

Опис версій програмного забезпечення та перелік апаратного забезпечення експерименту наведено у таблиці 3.

В якості архітектури експериментальної моделі для конвертації було використано послідовну (sequential) модель Keras, що містить 5 шарів: 1 допоміжний та 4 функціональних. Типи шарів та їх параметри наведено у таблиці 4.

У якості гіперпараметрів навчання було обрано розмір вибірки – 16, кількість епох навчання – 20. Оптимізатор моделі – «adam», loss функція – середньоквадратична похибка (MSE). Параметри шарів моделі та гіперпараметри навчання були підібрані експериментально для забезпечення кращої точності прогнозів із прийнятним часом навчання моделі – до 10 хвилин на безкоштовному пакеті обладнання servisy Collaboratory.



Рис. 1. Візуалізація набору даних АЗС

Таблиця 2

Приклад даних з набору даних промислового підприємства

Datetime	Consumption, кВт\год
2012-09-30 00:00:00	223
2012-09-30 01:00:00	215
2012-09-30 02:00:00	218
2012-09-30 03:00:00	210
2012-09-30 04:00:00	214



Рис. 2. Візуалізація набору даних молокозаводу

Конвертація експериментальних моделей у формат CoreML проводилася у формат.mlpackage, який підтримується версією iOS 15 і вище. Конвертація експериментальних моделей у формат Tensorflow Lite проводилася з використанням бібліотеки додаткових операцій Tensorflow Lite Selected OPS, що в теорії має покращити якість конвертації шарів моделі у мобільний формат. Використання додаткових операцій потребує залучення додаткової бібліотеки TensorFlowLiteSelectTfOps у мобільному додатку і на даний момент це робить неможливим запуск цього додатку на симуляторі мобільного пристрою iPhone, так як вона не має його підтримки архітектури ios-arm64-simulator [16] і потрібно використовувати реальний пристрій.

При тестуванні обидва формати моделі виконували обчислення на CPU, виконати запуск моделей на процесорі Neural Engine попри коректні налаштування не вдалося. Дане питання варто розглянути в наступних публікаціях, так як це може допомогти розвантажити CPU та покращити продуктивність.

Мобільний додаток для тестування моделей використовує завчасно експортовані у формат.csv тестові дані для прогнозів та виводить візуалізацію прогнозованих та фактичних даних у вигляді лінійної діаграми. Для оцінки продуктивності моделі CoreML використовувався вбудований Xcode пакет інструментів Instruments, що дозволяє отримати показники використаної пам'яті, дані про швидкість надання прогнозу. Для оцінки продуктивності

Таблиця 3

Опис програмного забезпечення та пристроїв експерименту

Інструменти навчання та конвертації моделей		
Назва	Версія	Опис
Python	3.7	Основна мова програмування середовища Colaboratory для взаємодії з Tensorflow
Tensorflow	2.12.0	Фреймворк навчання моделей
CoreMLTools	7.2	Бібліотека для конвертації моделей у формат CoreML
Colaboratory	1.2.0	Хмарний сервіс для навчання моделей машинного навчання
Інструменти розробки мобільного додатку		
Назва	Версія	Опис
Xcode	15.4	Середовище розробки мобільних додатків iOS
Swift	5	Мова програмування
SwiftUI	5.0	Бібліотека розробки графічного інтерфейсу для пристроїв Apple
Tensorflow Lite	2.13.0	Бібліотека запуску моделей Tensorflow Lite
TensorFlowLiteSelectTfOps	2.17.0	Допоміжна бібліотека до Tensorflow Lite для підтримки операцій, що не підтримуються основною бібліотекою
Charts	5.0	Бібліотека створення графіків у SwiftUI
CoreML	7.0	Бібліотека запуску моделей CoreML на пристроях Apple
iOS	18.2.1	Операційна система мобільного пристрою
Тестовий пристрій	15 Pro Max	Пристрій для запуску моделей ML

Таблиця 4

Структура експериментальної моделі

Шар	Параметри
InputLayer	(24, 1)
LSTM	200, activation='relu'
Dense	100, activation='relu'
Dropout	0.1
Dense	1

Таблиця 5

	АЗС		Молокозавод	
	CoreML	TFLite	CoreML	TFLite
$RMSE_0$ (кВт\год)	3.386	3.386	28.439	28.439
$RMSE_{mod}$ (кВт\год)	3.386	3.386	28.439	28.439
$RMSE_A$ (%)	0	0	0	0
RAM (Мб)	20-22	15-17	20-21	15-16
CPU	високий	високий	високий	високий
TS	nominal	nominal	nominal	nominal
t_{avg} (мс)	0,0256	0,0476	0.0265	0.0503
S (кб)	752	738	750	738

моделі Tensorflow використовувався комбінований підхід з використанням Xcode Instruments та додаткових функцій у коді з викликами `DispatchTime.now()` для отримання швидкості надання прогнозу, адже ці дані для моделей Tensorflow у Instruments відсутні. Для підрахунку $RMSE$ була використана самостійно реалізована функція, що аналогічна функції з пакету Tensorflow. Результати вимірів наведені у таблиці 5.

У результаті розрахунку $RMSE_A$ було виявлено, що конвертовані моделі не втратили точність прогнозів у порівнянні з їх оригіналами.

Візуалізації наданих прогнозів з фактичними тестовими даними з набору даних АЗС показують, що прогнозовані дані правильно відтворюють реальну циклічність споживання електроенергії. Приклади візуалізації для набору даних АЗС наведені на рисунках 3 та 4, для набору даних молокозаводу на рисунках 5 та 6. Візуалізації надані у масштабах однієї доби та одного тижня для демонстрації циклічності. Графіки прогнозів Tensorflow Lite та CoreML співпадають, адже мають однакову точність.

Висновки

В результаті оцінки конвертованих у мобільні формати моделей короткострокового прогнозування рівня споживання електроенергії для малого об'єкта інфраструктури та підприємства з закритим типом виробництва було отримано результати, що підтверджують можливість використання таких моделей як частини підсистеми прогнозування для системи управління ММ з використанням периферійних обчислень на мобільних пристроях. Для

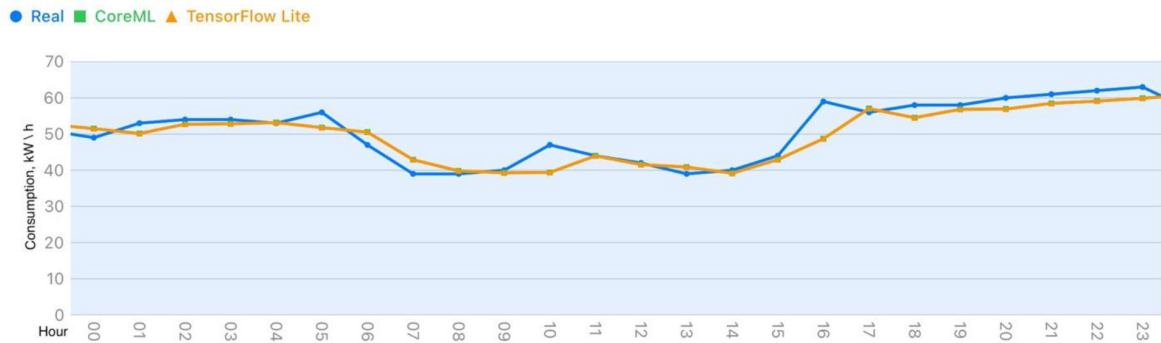


Рис. 3. Приклад прогнозованих та фактичних показників споживання з набору даних АЗС.
Масштаб 1 доба



Рис. 4. Приклад прогнозованих та фактичних показників споживання з набору даних АЗС.
Масштаб 1 тиждень

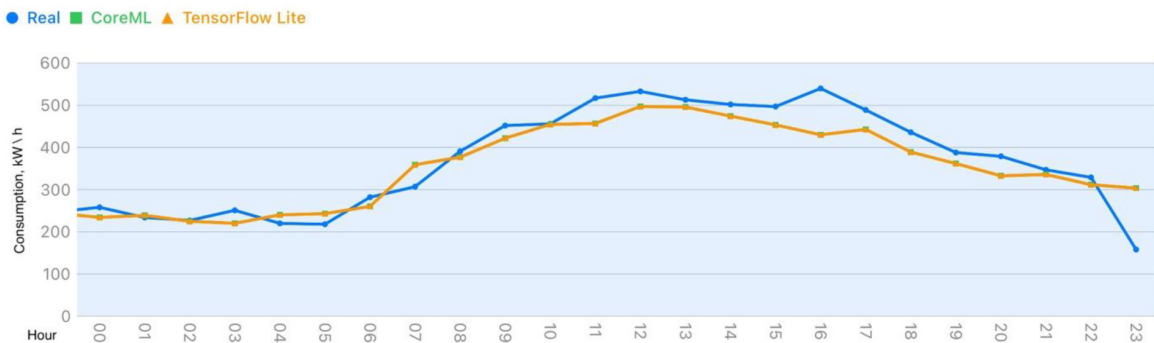


Рис. 5. Приклад прогнозованих та фактичних показників споживання з набору даних молокозаводу.
Масштаб 1 доба

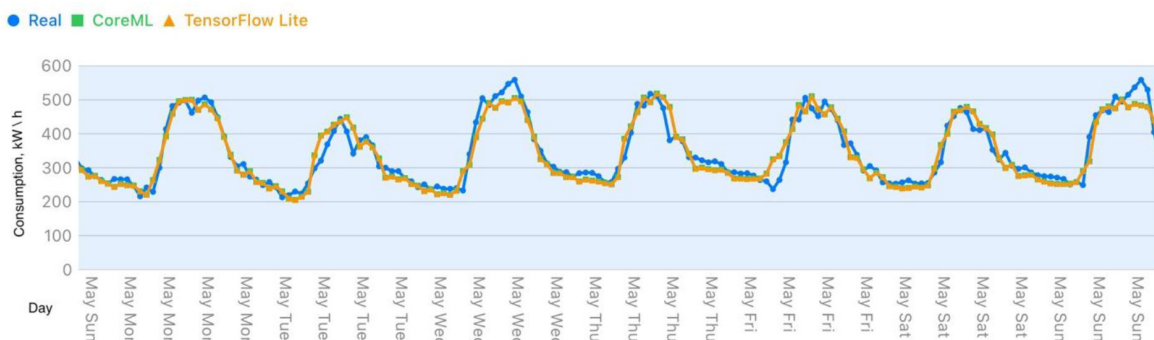


Рис. 6. Приклад прогнозованих та фактичних показників споживання з набору даних молокозаводу.
Масштаб 1 тиждень

набору даних АЗС та набору даних молокозаводу зниження точності конвертованої моделі у порівнянні з оригінальними моделями зафіксовано не було як для Tensorflow Lite, так і CoreML. Це доводить можливість застосування мобільних ML-моделей у підсистемі прогнозування з їх розміщенням на периферійних мобільних пристроях без використання хмарних ресурсів для надання прогнозів. Всі моделі після конвертації мали відносно невеликий розмір – від 700 до 800 кілобайт, що дозволяє їх використання на будь-яких сумісних мобільних пристроях. Рівень споживання оперативної пам'яті при завантаженні моделі та виконанні прогнозів коливався від 16 до 22 мегабайт, що є незначним числом у порівнянні з загальною кількістю оперативної пам'яті, що доступна на сучасних мобільних пристроях. Під час виконання прогнозу обидва типи моделей створювали короточасне високе навантаження на центральний процесор пристрою, що не спричинило нагріву пристрою і є допустимим. Швидкість надання прогнозів моделями складала від 0.256 до 0.503 мс на 1 прогноз, що дозволяє використовувати дані моделі у режимі реального часу. Моделі CoreML показали кращу продуктивність – в середньому на 50 % швидше надання прогнозів, в той же час моделі Tensorflow Lite можуть бути використані як на Android, так і на iOS і показали менший рівень споживання оперативної пам'яті. На жаль, вбудовані функції Xcode не дозволяють точно оцінити вплив запуску моделей на рівень використання заряду акумулятора, але за показниками використання CPU та рівнем нагріву пристрою можна зробити припущення, що вплив знаходиться на рівні звичайних мобільних додатків.

Список використаної літератури

1. Biegańska M. IoT-Based Decentralized Energy Systems. *Energies*. 2022. Вип. 15, № 21. С. 7830. DOI: <https://doi.org/10.3390/en15217830>
2. Wynn S. L. L., Boonraksa T., Boonraksa P., Pinthurat W., Marungsri B. Decentralized Energy Management System in Microgrid Considering Uncertainty and Demand Response. *Electronics*. 2023. Вип. 12, № 1. С. 237. DOI: <https://doi.org/10.3390/electronics12010237>
3. Parfenenko Y. V., Shendryk V. V., Kholiavka Y. P., Pavlenko P. M. COMPARISON OF SHORT-TERM FORECASTING METHODS OF ELECTRICITY CONSUMPTION IN MICROGRIDS. *Radio Electronics, Computer Science, Control*. 2023. № 1. С. 14–14. DOI: <https://doi.org/10.15588/1607-3274-2023-1-2>
4. Vries A. de. The growing energy footprint of artificial intelligence. *Joule*. 2023. Вип. 7, № 10. С. 2191–2194. DOI: <https://doi.org/10.1016/j.joule.2023.09.004>
5. Guidi G., Dominici F., Gilmour J., Butler K., Bell E., Delaney S., Bargagli-Stoffi F. J. Environmental Burden of United States Data Centers in the Artificial Intelligence Era. arXiv, 2024.
6. Data Privacy Laws and Regulations Around the World, URL: <https://securiti.ai/privacy-laws/> (дата звернення: 28.12.2024).
7. LiteRT and TensorFlow operator compatibility, URL: https://ai.google.dev/edge/litert/models/ops_compatibility (дата звернення: 02.02.2025).
8. CoreMLTools Source and Conversion Formats, URL: <https://apple.github.io/coremltools/docs-guides/source/target-conversion-formats.html> (дата звернення: 02.02.2025).
9. Reducing terminations in your app, URL: <https://developer.apple.com/documentation/xcode/reduce-terminations-in-your-app> (дата звернення: 02.02.2025).
10. Tajjour S., Singh Chandel S. A comprehensive review on sustainable energy management systems for optimal operation of future-generation of solar microgrids. *Sustainable Energy Technologies and Assessments*. 2023. Вип. 58. С. 103377. DOI: <https://doi.org/10.1016/j.seta.2023.103377>
11. Elhanashi A., Dini P., Saponara S., Zheng Q. Advancements in TinyML: Applications, Limitations, and Impact on IoT Devices. *Electronics*. 2024. Вип. 13, № 17. С. 3562. DOI: <https://doi.org/10.3390/electronics13173562>
12. LiteRT for Microcontrollers, URL: <https://ai.google.dev/edge/litert/microcontrollers/overview> (дата звернення: 03.02.2025).
13. Experience more with Core ML, URL: <https://developer.apple.com/machine-learning/core-ml/> (дата звернення: 02.02.2025).
14. What Is Core ML Tools?, URL: <https://apple.github.io/coremltools/docs-guides/source/overview-coremltools.html> (дата звернення: 02.02.2025).
15. IsolationForest Algorithm implementation in sklearn, URL: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html#sklearn.ensemble.IsolationForest> (дата звернення: 09.10.2024).
16. Tensorflow official repository, URL: <https://github.com/tensorflow/tensorflow/issues/61857> (дата звернення: 23.10.2024).

References

1. Biegańska, M. (2022). IoT-Based Decentralized Energy Systems. *Energies*, 15(21), Article 21. <https://doi.org/10.3390/en15217830>

2. Wynn, S.L.L., Boonraksa, T., Boonraksa, P., Pinthurat, W., & Marungsri, B. (2023). Decentralized Energy Management System in Microgrid Considering Uncertainty and Demand Response. *Electronics*, 12(1), Article 1. <https://doi.org/10.3390/electronics12010237>
3. Parfenenko, Y.V., Shendryk, V.V., Kholiavka, Y.P., & Pavlenko, P.M. (2023). COMPARISON OF SHORT-TERM FORECASTING METHODS OF ELECTRICITY CONSUMPTION IN MICROGRIDS. *Radio Electronics, Computer Science, Control*, 1, Article 1. <https://doi.org/10.15588/1607-3274-2023-1-2>
4. Vries, A. de. (2023). The growing energy footprint of artificial intelligence. *Joule*, 7(10), 2191–2194. <https://doi.org/10.1016/j.joule.2023.09.004>
5. Guidi, G., Dominici, F., Gilmour, J., Butler, K., Bell, E., Delaney, S., & Bargagli-Stoffi, F. J. (2024). *Environmental Burden of United States Data Centers in the Artificial Intelligence Era* (No. arXiv:2411.09786). arXiv. <https://doi.org/10.48550/arXiv.2411.09786>
6. *Data Privacy Laws and Regulations Around the World*. (n.d.). Retrieved 28 December 2024, from <https://securiti.ai/privacy-laws/>
7. *LiteRT and TensorFlow operator compatibility*. (n.d.). [Documentation]. Google AI for Developers. Retrieved 2 February 2025, from https://ai.google.dev/edge/litert/models/ops_compatibility
8. *CoreMLTools Source and Conversion Formats*. (n.d.). [Documentation]. CoreMLTools Official GitHub. Retrieved 2 February 2025, from <https://apple.github.io/coremltools/docs-guides/source/target-conversion-formats.html>
9. *Reducing terminations in your app*. (n.d.). [Documentation]. Apple Developer Portal. Retrieved 2 February 2025, from <https://developer.apple.com/documentation/xcode/reduce-terminations-in-your-app>
10. Tadjour, S., & Singh Chandel, S. (2023). A comprehensive review on sustainable energy management systems for optimal operation of future-generation of solar microgrids. *Sustainable Energy Technologies and Assessments*, 58, 103377. <https://doi.org/10.1016/j.seta.2023.103377>
11. Elhanashi, A., Dini, P., Saponara, S., & Zheng, Q. (2024). Advancements in TinyML: Applications, Limitations, and Impact on IoT Devices. *Electronics*, 13(17), Article 17. <https://doi.org/10.3390/electronics13173562>
12. *LiteRTforMicrocontrollers*. (n.d.). LiteRTforMicrocontrollers. Retrieved 3 February 2025, from <https://ai.google.dev/edge/litert/microcontrollers/overview>
13. *Experience more with Core ML*. (n.d.). [Documentation]. Apple Developer Portal. Retrieved 2 February 2025, from <https://developer.apple.com/machine-learning/core-ml/>
14. *What Is Core ML Tools?* (n.d.). [Documentation]. Guide to Core ML Tools. Retrieved 2 February 2025, from <https://apple.github.io/coremltools/docs-guides/source/overview-coremltools.html>
15. *IsolationForest Algorithm implementation in sklearn*. (n.d.). [Documentation]. Scikit-Learn, Machine Learning in Python. Retrieved 9 October 2024, from <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html#sklearn.ensemble.IsolationForest>
16. *Tensorflow official repository*. (n.d.). Retrieved 23 October 2024, from <https://github.com/tensorflow/tensorflow/issues/61857>