ІНФОКОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА ЕЛЕКТРОННА ІНЖЕНЕРІЯ
INFORMATION AND COMMUNICATION TECHNOLOGIES,
ELECTRONIC ENGINEERING

№ 1 (1), 2021

ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

УПРАВЛІННЯ МЕРЕЖАМИ МОБІЛЬНОГО ЗВ'ЯЗКУ 5G ЗА ДОПОМОГОЮ ВИКОРИСТАННЯ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ

Б. Шубин, М. Климаш, А. Масюк, А. Осташевський
Національний університет "Львівська політехніка", вул. С. Бандери, 12, Львів, 79013, Україна

Відповідальний за рукопис: Богдан Шубин (e-mail: boshubin@gmail.com).

(Подано 1 квітня 2021)

Стаття стосується проблеми надмірного навантаження трафіком комірок базових станцій. З метою зменшення впливу цієї проблеми на якість послуг операторів мережі мобільного зв'язку запропоновано використовувати технологію штучного інтелекту (ШІ) для аналізу та прогнозування навантаження у мережі. ШІ чудово підходить для середовищ із безпровідним зв'язком, оскільки в ньому є безліч доступних даних для аналізу та отримання певних шаблонів. В статті запропоновано модель машинного навчання та архітектуру нейронної мережі для прогнозування навантаження на 5G комірки.

Ключові слова: *штучний інтелект; машинне навчання; нейронні мережі; 5G.*
УДК: 621.126

1. Вступ

У наш час розвивається багато нових технологій, та послуг, які вимагають надзвичайно швидкого передавання даних із мінімальними затримками. Також для всіх цих пристроїв дуже важливими параметрами є енергозбереження, безпека та повномасштабне покриття, оскільки, наприклад, підтримка автоматизованого управління автомобілем із розпізнаванням смуг та дорожніх знаків потребує покриття та з'єднання на кожному метрі дороги. Все це створює колосальне навантаження як на комірки 5G мережі, так і на 5G мережу загалом [1]. Треба пам'ятати, що крім перерахованих вище сервісів мережу використовуватимуть і прості користувачі за допомогою своїх мобільних телефонів, ноутбуків, планшетів тощо. Це все спричиняє проблему забезпечення нестабільної швидкості передавання даних, що залежить від розподілу ширини смуги частот між комірками.

Одне із можливих рішень, запропоноване в статті, – це забезпечення вищеперерахованих послуг пліч-о-пліч із використанням штучного інтелекту. Завдяки стрімкому розвитку штучного інтелекту є надзвичайно багато готових моделей та алгоритмів, які можуть виконувати різноманітні завдання із різних сфер життя. Отже, у зв'язку з цим, наше завдання – придумати не алгоритм навчання нейронної моделі, а алгоритм, який зможе вирішити проблему динамічного розподілу

© 2021 Національний університет "Львівська політехніка"

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

ширини смуги частот, та показати, як можна використовувати 5G мережу разом із певною нейронною моделлю.

2. Аналіз та принцип роботи Глибокого навчання

Загалом, штучний інтелект (AI) – це обчислювальна парадигма, яка наділяє машини інтелектом, спрямована на те, щоб навчити їх працювати, реагувати та вчитися, як люди [2]. Робота багатьох методів потрапляє під це визначення (рис. 1), ураховуючи машинне навчання, експертні системи та еволюційні алгоритми. Серед них машинне навчання дає змогу штучним процесам поглинати знання із даних та приймати рішення, не будучи чітко програмованими. Алгоритми машинного навчання (ML), як правило, класифікують на контрольоване, неконтрольоване та допоміжне навчання.

Глибоке навчання – це сім'я методів машинного навчання [3], що імітують біологічні нервові системи та виконують репрезентативне навчання за допомогою багатошарових перетворень, що поширюються на всі три парадигми навчання, згадані вище.

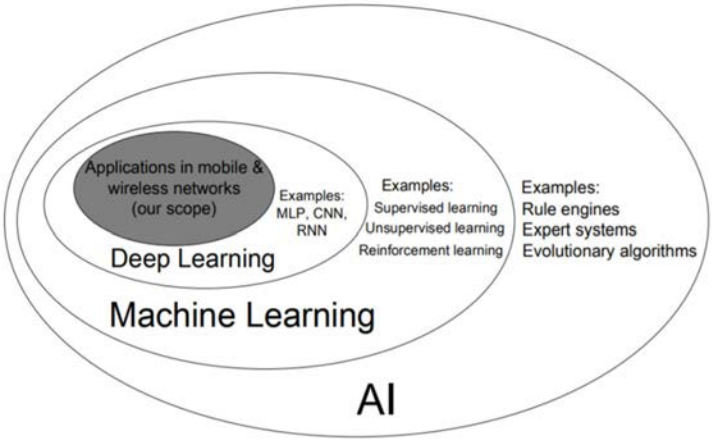

Рис. 1. Технології, які охоплює Штучний Інтелект

На рис. 1 видно, що, на відміну від Machine Learning та Artificial intelligence, Deep Learning поділяється на такі види, як рекурентна і згорткова нейронні мережі. Саме Deep Learning широко застосовують у мобільних та бездротових мережах для уникнення ручного програмування алгоритмів.

Ключовою метою глибоких нейронних мереж є наближення складних функцій за допомогою складання простих і заздалегідь визначених операцій одиниць (або нейронів) [4]. Така цільова функція може бути майже будь-якого типу, наприклад, відображення між зображеннями та їх мітками класів (класифікація), обчислення майбутніх цін акцій на основі історичних значень (регресія) або навіть прийняття рішення про наступний оптимальний шаховий хід з огляду на поточний статус дошки (управління). Виконувані операції зазвичай визначаються зваженою комбінацією певної групи прихованих одиниць із нелінійною функцією активації, залежно від структури моделі. Такі операції разом із вихідними одиницями називають "шарами". Архітектура нейронної мережі нагадує процес сприйняття у мозку, де певний набір одиниць активується з урахуванням поточного середовища, впливаючи на результат роботи нейронної мережі.

У математичному плані архітектура глибинних нейронних мереж, як правило, диференційована, тому ваги (або параметри) моделі можна вивчити, мінімізуючи функцію втрат за допомогою методів градієнтного спуску зі зворотним поширенням, дотримуючись основного правила ланцюга [5]. На рис. 2 проілюстровано процеси навчання та висновків глибокої нейронної мережі, де використовується двовимірна (2D) згорткова нейронна мережа (CNN)

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

чотиришарової CNN. w (·) позначають ваги кожного прихованого шару, σ (·) – це функція активації, λ стосується швидкості навчання, ∗ (·) позначає операцію згортки, а *L (w)* – функція втрат, яку слід оптимізувати.

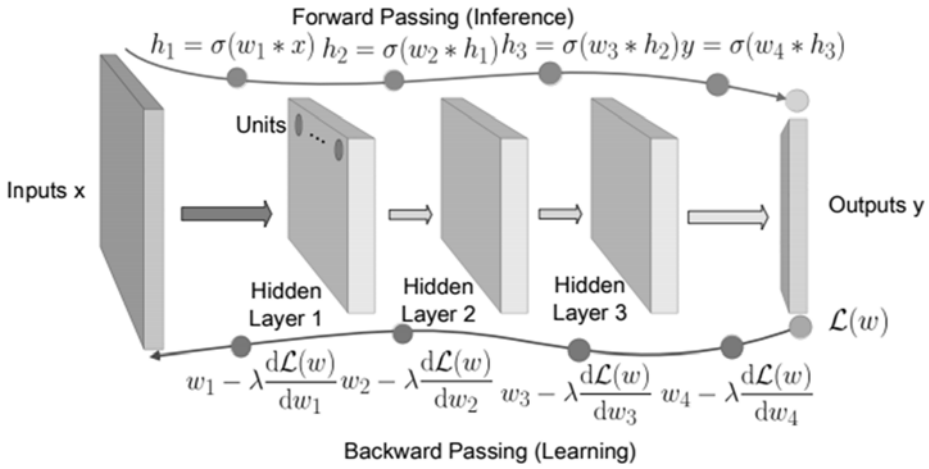

Рис. 2. Ілюстрація процесу навчання та висновків

Процес навчання поділяється на два основні кроки: поширення вперед та зворотне поширення.

Поширення вперед (Forward propagation). На рис. 2 показано CNN з п'ятьма шарами, тобто вхідний шар (сірий), три прихованих шари (синій) і вихідний шар (оранжевий). У разі прямого поширення вхід *x* спочатку обробляється згортковим шаром, який виконує таку згорткову операцію:

$$h_1 = \mathrm{s}\,(w_1 * x), \tag{1}$$

де h_1 – вихід першого прихованого шару; w_1 – згортковий фільтр; а s() – функція активації, спрямована на покращення нелінійності та репрезентативності моделі. Вихід *h*1 надається як вхідні дані й обробляється наступними двома згортковими шарами, що в кінцевому підсумку дає кінцевий вихід *y*. Це може бути, наприклад, вектор ймовірностей для різних можливих візерунків (фігур), виявлених на вході (зображення). Для належної підготовки CNN використовується функція збитків $L(w)$ для вимірювання відстані між вихідним сигналом *y* та істинною базою *y*. Мета тренувань – знайти найкращі ваги *w*, щоб мінімізувати функцію втрат $L(w)$. Цього можна досягти зворотним поширенням через градієнтний спуск.

Зворотне поширення (Backward propagation). Під час зворотного поширення обчислюється градієнт функції втрат *L (w)* над вагою останнього прихованого шару та оновлюється вага за допомогою обчислень:

$$w_4 = w_4 - \mathrm{l}\ \frac{dL(w)}{dw_4}, \tag{2}$$

де λ позначає швидкість навчання, яка контролює розмір кроку руху в напрямку, визначеному градієнтом. Для кожної гирі виконується однакова операція, із дотриманням правила ланцюга. Процес повторюється, і врешті-решт градієнтний спуск приведе до набору *w*, який мінімізує $L(w)$. Для інших структур *NN* процеси підготовки та умовиведення подібні.

3. Принцип роботи рекурентних нейронних мереж

Зазвичай користувачі обмежені багатьма соціальними факторами, такими як робота, сім'я, машини, громадський транспорт тощо. Отже, їхні моделі мобільності подібні одна до одної, тобто кожен користувач має певні звичні траєкторії руху та улюблені місця (бу-

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

динок, робота, ресторан, тренажерний зал тощо). Таку функцію можна легко спроєктувати на велику групу користувачів, так що ми можемо припускати періодичні залежності навантаження трафіку в кожному конкретному макросі та малій комірці. Основне завдання – розробити інтелектуальний метод, який зможе передбачити періодичне навантаження трафіку кожної комірки та відповідно виконати розподіл спектра найефективнішим способом. Загальна ідея полягає у тому, щоб створити "ліквідну мережу", пропускна здатність якої відповідає попиту на трафік, використовуючи довгострокові та короткострокові прогнози періодичними нейронними мережами.

Рекурентні нейронні мережі виявились перспективним рішенням для аналізу та прогнозування часових рядів із періодичними залежностями. Така особливість робить їх придатними для прогнозування майбутніх подій за наявності прихованих періодичних закономірностей. Наприклад, рекурентні нейронні мережі широко використовують для різних застосувань, таких як машинний переклад, прогнозування часових рядів, розпізнавання мови, виявлення аномалій, прогнозування цін на акції та валюти тощо. Тому ми використовуємо їхні можливості для прогнозування трафіку в комірках.

Обмеження звичайних рекурентних нейронних мереж полягає у тому, що вони мають дуже коротку пам'ять. Це обмеження спричинене проблемою градієнта, що зникає під час зворотного поширення. Оскільки градієнт із часом зменшується, швидкість навчання стає дуже малою, тобто мережа майже припиняє навчання і забуває довші залежності. Ця проблема не дає змоги одночасно тренувати повторювану нейронну мережу як для довгострокових залежностей (місяців, тижнів, днів), так і для короткочасних залежностей (годин, хвилин).

Незважаючи на те, що введенням прихованого стану нам вдалося ефективно визначати взаємозв'язок між входами мережі та передавати його між ітераціями, постало запитання: чи можна зробити RNN глибшими? Є чотири способи зробити рекурентну нейронну мережу глибшою:

1. Перший і найочевидніший спосіб – додавати приховані стани один на одного, зв'язуючи їхні входи.
2. Можна також додати нелінійний прихований шар між входами прихованих станів.
3. Можна збільшити глибину переходу між прихованими шарами.
4. І останній варіант – можемо збільшити глибину прихованого та вихідного переходу.

У цій статті порівняємо дві найпопулярніші рекурентні нейронні мережі на основі LSTM та GRU комірки і виберемо одну з них для проведення дослідження.

Мережі LSTM розроблено для вивчення довгострокових залежностей. На відміну від звичайної рекурентної нейронної мережі, яка складається із одношарових комірок, комірки LSTM містять по чотири шари, що передбачає функціональність довгострокової та короткочасної пам'яті. Ці клітини повторюються в послідовному ланцюгу, подібному до звичайної рекурентної нейронної мережі. Така структура дає змогу мережам LSTM уникнути проблеми зникнення градієнтів.

Опишемо процес оброблення інформації через мережу LSTM. Одна комірка мережі LSTM складається із вхідного входу, рекурентної комірки із прихованим станом та вихідного входу. Кожен шлюз містить функцію активації сигмоподібної форми, яка має межі [0, 1], тобто значення 0 дозволяє комірці забути якусь менш важливу інформацію з попередніх станів, тоді як 1 дає змогу зберігати важливу інформацію та передавати її в наступні стани. Повторна клітина містить гіперболічну функцію активації тангенса, межі якої [–1, 1]. Загальну структуру клітини LSTM подано на рис. 3.

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

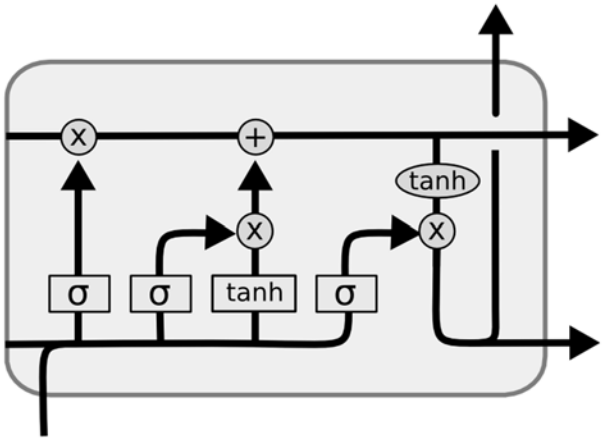

Рис. 3. Структура LSTM комірки

GRU архітектуру (рис. 4) запропоновано в 2014 р. як оптимізацію і спрощення для LSTM, хоча такі оптимізація та спрощення спричинили деякі обмеження, які розглянемо нижче. Модель розв'язує ту саму задачу, що й LSTM, проте, як згадано вище, з деякими перевагами й недоліками. У зв'язку з описаним вище GRU інколи можна розглядати як варіацію LSTM. Отже, розглянемо, як працює GRU модель.

Як зазначено вище, GRU є покращенням стандартної референтної мережі, отже, щоб вирішити проблему зникнення градієнта, використовує так звані ворота оновлення і перезавантаження. Якщо не вдаватись у деталі, є вектори, які приймають рішення, чи інформація може бути передана на вихід. Їх особливістю є те, що впродовж навчання вони можуть зберігати інформацію тривалий час, без її зникнення або затирання через нерелевантність.

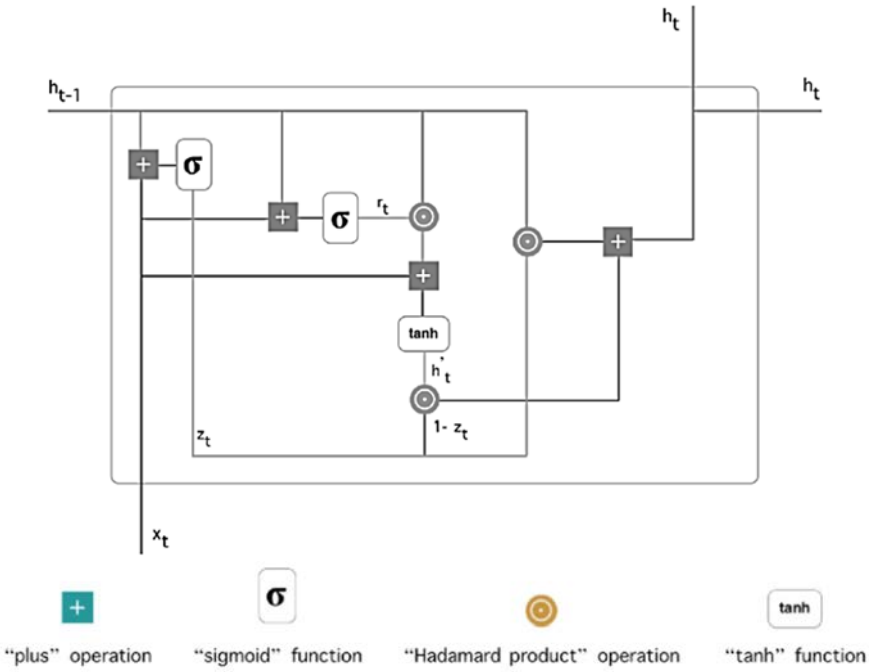

Рис. 4. Умовні позначення операторів блока GRU

Ворота оновлення (Update gates). Субрахунок оновлення воріт починається із використання формули:

$$z_t = \mathrm{s}\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right). \quad (3)$$

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

Коли x_t підключено до нейронних воріт, він множиться на власну вагу $W(z)$. Те саме виконується для h_{t-1}, який зберігає інформацію для попереднього $t-1$ елемента і є перемноженим на власну вагу $U(z)$. Обидва результати додаються, і сигмоподібна функція активації видає результат між нулем та одиницею (рис. 5).

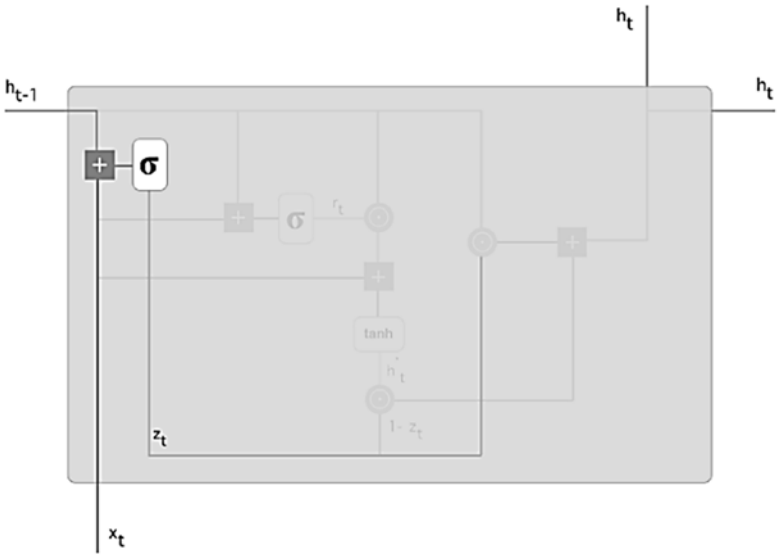

Рис. 5. Ілюстрація оновлення воріт за допомогою сигмоподібної функції

Оновлення воріт допомагає моделі зрозуміти, як часто стара інформація з попередніх ітерацій використовуватиметься в майбутньому. Це надзвичайно потужний інструмент, оскільки модель може зробити копію всієї інформації з минулого й усунути риск зникнення градієнта.

Ворота скидання (Reset gates). Ці ворота вирішують, скільки застарілої інформації необхідно видалити. Щоб це підрахувати, використовують формулу:

$$r_t = \mathrm{s}\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right). \tag{4}$$

Ця формула майже така сама, як і для воріт оновлення, за винятком вагових коефіцієнтів і призначення.

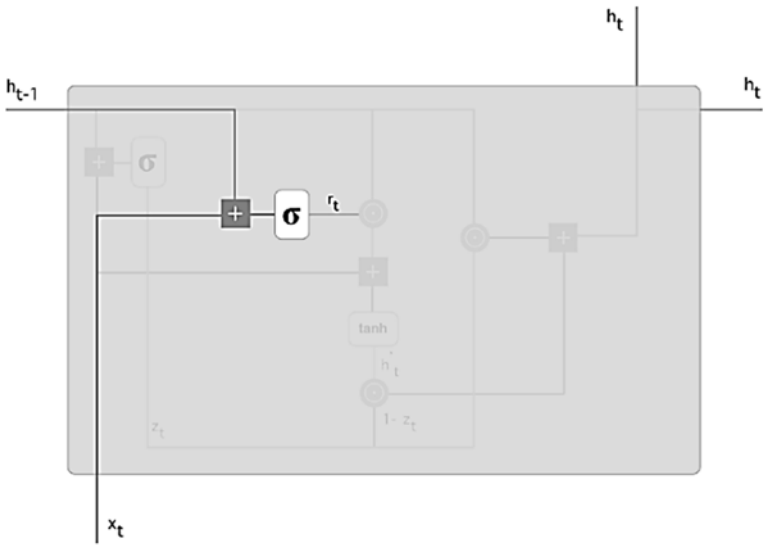

Рис. 6. Ілюстрація процесу регулювання воріт за допомогою сигмоподібної функції

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

Як із воротами оновлення, на вхід ми подаємо *h*_(*t*–1) (голуба лінія) та *x*_*t* (пурпурова лінія), додаємо їх разом, враховуючи їх вагові коефіцієнти, і застосовуємо до них сигмоподібну функцію.

Для нашого дослідження пропонуємо використовувати рекурентну мережу на основі GRU комірки.

4. Передбачення навантаження за допомогою використання рекурентної нейронної мережі на основі GRU комірки

Для навчання нейронної мережі, яка прогнозуватиме абонентське навантаження на 5G комірку, використано GRU модель. Процес навчання моделі та зберігання результатів відбувався у середовищі розроблення Python. Для навчальних і тестових даних було використано масиви абонентського навантаження з часовими інтервалами 5 хв для п'яти 5G комірок. Дані демонструють пересування абонентів між комірками із прив'язкою до часу. Навчання моделі відбувається за допомогою фрагмента коду, наведеного на рис. 7.

```
def train_model(model, X_train, y_train, name, config):
    model.compile(loss="mse", optimizer="rmsprop", metrics=['mape'])
    hist = model.fit(
        X_train, y_train,
        batch_size=config["batch"],
        epochs=config["epochs"],
        validation_split=0.05)
    model.save('model/' + name + '.h5')
    df = pd.DataFrame.from_dict(hist.history)
    df.to_csv('model/' + name + ' loss.csv', encoding='utf-8', index=False)
```

Рис. 7. Фрагмент коду для навчання нейронної мережі

Тут вказуються шлях до моделі (в цьому випадку GRU) і шляхи для збереження метаданих. Також задається початкова конфігурація batch та epochs, які становлять 8000 та 7 відповідно. Це максимально ефективні характеристики для уникнення перенасичення моделі.

Передбачення відбувається на основі навченої моделі й порівнюється з тестовим набором даних, який обмежений однією добою.

```
    predicted = model.predict(X_test)
    predicted = scaler.inverse_transform(predicted.reshape(-1, 1)).reshape(1, -1)[0]
    predication = pd.DataFrame(predicted).to_csv('result.csv')
    y_preds.append(predicted[:288])
    print(name)
    eva_regress(y_test, predicted)

plot_results(y_test[: 288], y_preds, names)
```

Рис. 8. Фрагмент коду, який генерує передбачення на основі натренованої моделі

Запускаємо процес навчання моделі п'ять разів для п'яти наборів даних, які належать до 1 з 5 5G комірок, і генеруємо графіки передбачення для кожної комірки.

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

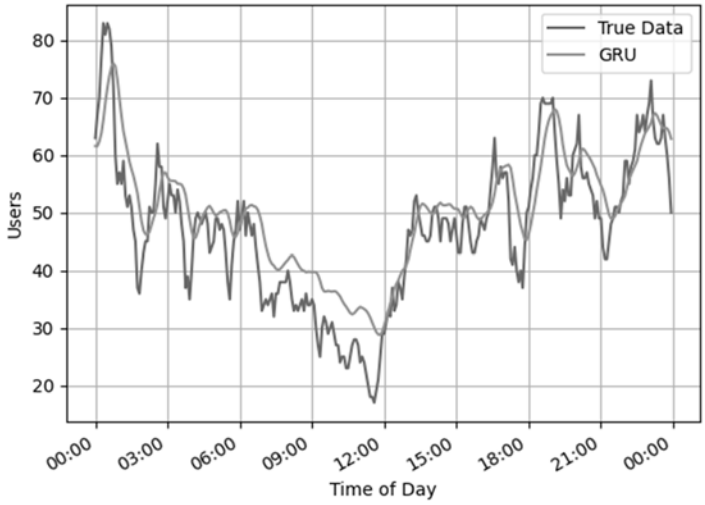

Рис. 9. Співвідношення передбачення до реальних даних для комірки 1

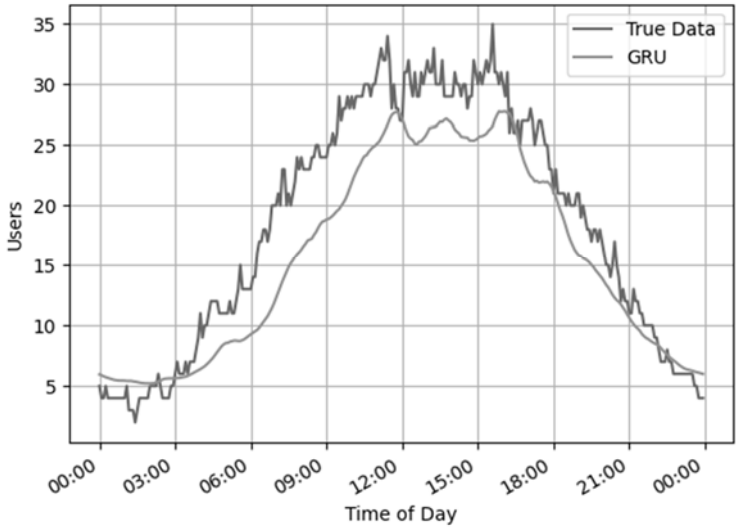

Рис. 10. Співвідношення передбачення до реальних даних для комірки 2

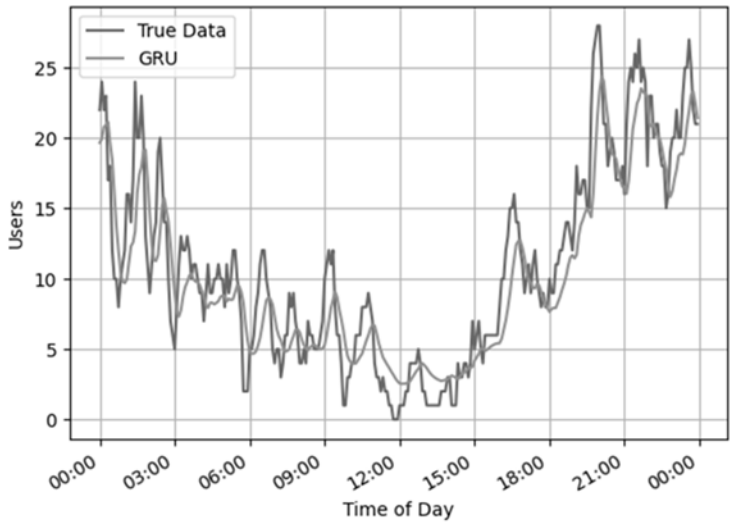

Рис. 11. Співвідношення передбачення до реальних даних для комірки 3

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

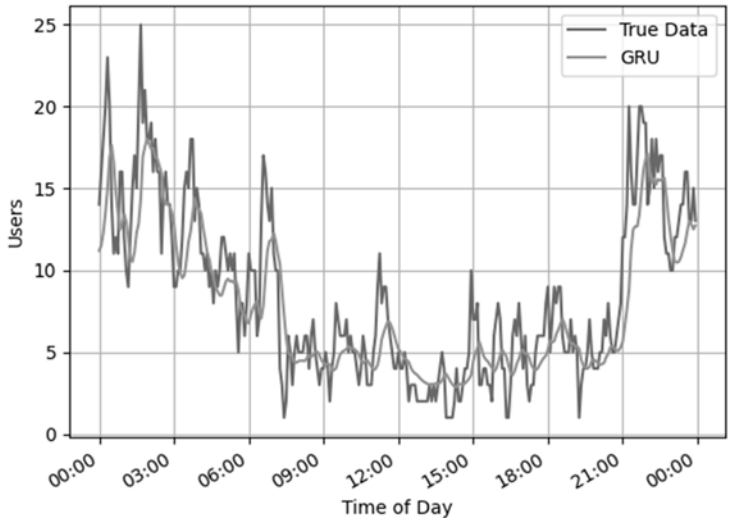

Рис. 12. Співвідношення передбачення до реальних даних для комірки 4

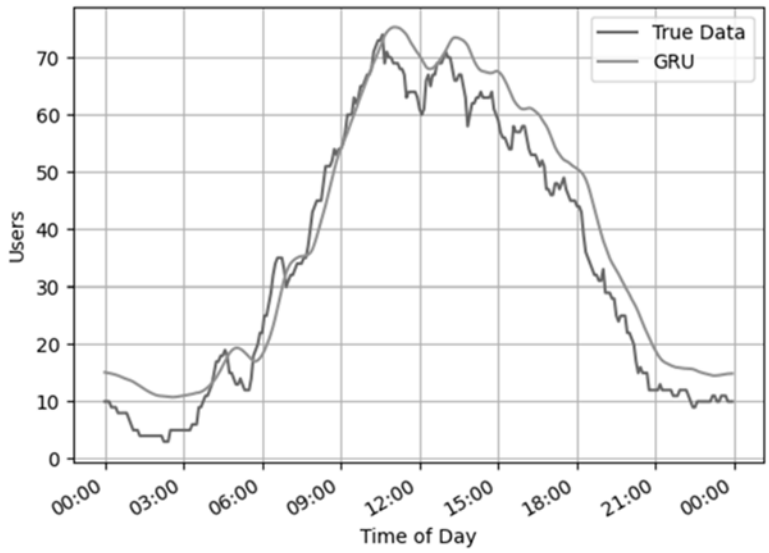

Рис. 13. Співвідношення передбачення до реальних даних для комірки 5

Проаналізувавши графіки, можна зробити два очікувані висновки:

- Пікова кількість користувачів відрізняється для кожної комірки, що відповідає реальній ситуації.
- Передбачення нейронної мережі відтворює картину реального навантаження, хоча слід зазначити, що передбачення таких даних доволі складне для нейронної мережі, оскільки є зони, які змінюються дуже стрімко і непередбачувано.

Одна із найістотніших переваг такого передбачення – це можливість аналізувати майбутнє, потенційне навантаження і забезпечувати ефективний динамічний перерозподіл ширини смуги частот для забезпечення порівняно стабільної швидкості для абонентів.

Висновки

У цій статті проаналізовано роботу рекурентних нейронних мереж на основі LSTM та GRU комірок. Для проведення моделювання вибрано рекурентну нейронну мережу на основі GRU комірки. Отримані результати демонструють, що можна передбачити навантаження у комірці з високою точністю, що дасть змогу аналізувати майбутнє, потенційне навантаження і забезпечувати

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

ефективний динамічний перерозподіл ширини смуги частот для забезпечення порівняно стабільної швидкості для абонентів.

Список використаних джерел

[1] *R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, and H. Zhang, "Intelligent 5G: When cellular networks meet artificial intelligence", IEEE Wireless Commun., vol. 24, no. 5, pp. 175–183, Oct. 2017.*

[2] *Koutník, J., Gomez, F., &Schmidhuber, J. Evolving neural networks in compressed weight space, 2010.*

[3] *B. Bojović, E. Meshkova, N. Baldo, J. Riihijärvi, and M. Petrova, "Machine learning-based dynamic frequency and bandwidth allocation in self-organized LTE dense small cell deployments", EURASIP J. Wireless Commun. Netw., vol. 2016, no. 1, 2016, Art. no. 183.*

[4] *A. Rosebrock, Deep Learning for Computer Vision, 1st ed. Baltimore, MD, USA: PyImageSearch, 2017. [Online]. Available: https://www.pyimagesearch.com/deep-learning-computer-vision- python-book/*

[5] *H. Sun et al., "Learning to optimize: Training deep neural networks for interference management", IEEE Trans. Signal Process., vol. 66, no. 20, pp. 5438–5453, Oct. 2018.*

5G NETWORKS MANAGEMENT METHODS USING ARTIFICIAL INTELLIGENCE INSTRUMENTALITY

B. Shubyn, M. Klymash, T. Maksymyuk, A. Ostashevskiy

Lviv Polytechnik National University, 12, S. Bandery Str., Lviv, 79013, Ukraine

The article is devoted to the problem of excessive traffic of base station cells. In order to reduce the impact of this problem on the quality of services of mobile network operators, it is proposed to use artificial intelligence (AI) technology to analyze and predict the load on the network. AI is great for wireless environments, as it has a lot of data available for analysis and obtaining certain patterns. The article proposes a model of machine learning and neural network architecture for forecasting the load on 5G cells.

Key words: *artificial intelligence; machine learning; neural networks; 5G.*

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua