

Кулаков Ю.О., д.т.н.,
orcid.org/0000-0002-8981-5649,
e-mail: ya.kulakov@gmail.com,

Корочкін О.В., к.т.н.,
orcid.org/0000-0002-6569-5849,
e-mail: avcora@gmail.com,

Омелянський А.С.,
e-mail: aomelyansky@gmail.com

ВИКОРИСТАННЯ КЛАСИФІКАТОРА РЕСУРСІВ В МОДЕЛЯХ ПРОГНОЗУВАННЯ ТРАФІКУ SDN МЕРЕЖ

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Вступ

В програмно-конфігурованих мережах (*Software-defined Networking, SDN*) площина керування відокремлюється від площини даних. Логіка пересилки переноситься на централізований *SDN* контролер. Це дозволяє йому збирати мережеві дані в режимі *online*, зберігати їх детальну історію та оперувати глобальним представленням мережі. Програмованість *SDN* мережі та гнучкість в її конфігуруванні дозволяють значно спростити процес керування мережевими ресурсами, прогнозування *QoS/QoE* та використовувати ефективні алгоритми маршрутизації. Однак, однією з ключових проблем керування залишається прогнозування на наступний інтервал часу потреби в пропускній можливості каналів між мережевими пристроями. Прогнозування трафіку полягає в можливості передбачити якомога точніше майбутні потоки мережевих даних на основі статистики таких потоків за попередні періоди.

Зазвичай для прогнозування трафіку використовуються методики аналізу часових рядів які розглядають *SDN* мережу як модель, параметри якої описують стохастичний процес [1]. Прикладом таких моделей є авторегресійні моделі *ARMA* і *ARIMA* [2] та їх багатовимірні аналоги, які працюють з векторами даних *VARMA, VARIMA*.

В останні часи в прогнозуванні трафіку почали широко використовувати методи штучного інтелекту та машинного навчання [3]. Глибинні нейронні мережі

(*Deep Neural Network, DNN*) за допомогою нелійних шарів виділяють проміжні ознаки, які використовуються для розрахунку прогнозу на фінальному шарі повноз'язної нейронної мережі. Основними архітектурами *DNN* для прогнозування трафіку є [4]:

- згорткові нейронні мережі (*CNN*) з шарами причинних згорток (*causal convolution*);
- рекурентні нейронні мережі (*RNN*) з прихованим внутрішнім станом (коміркою *RRN*) та механізмом вентилів (*input/output/forget gates*);
- мережі з шарами уваги (*attention layers*).

Для отримання прогнозу на кілька точок у майбутньому використовуються багатогоризонтні моделі прогнозування. До них відносяться ітераційні авторегресійні архітектури *DNN* (прогнози через рекурсивну подачу даних на майбутні кроки) та архітектури «*sequence-to-sequence*» (кодер для узагальнення минулої інформації та декодер для поєднання з відомими майбутніми даними).

Розрахунковий інтервал залежить від області в якій буде використовуватися прогноз. Короткострокове прогнозування використовується для динамічного розподілу каналів та керування перевантаженням вузлів мережі. Для маршрутизації та керування ресурсами необхідно розраховувати прогноз на більш довгі інтервали часу.

В розглянутих методах прогнозування трафіку не враховується його характер (які саме дані передаються). Однак, наприклад, запит до *FTP* сервера (*get file*) чи запит до *HTTP* серверу (отримання комплексної сторінки, яка будується за допомогою запитів до *backend* сервісів та СУБД) мають приблизно однакові розміри пакетів запитів однак в майбутньому викликають різну реакції мережі (майбутній трафік). В даній роботі пропонується виконувати класифікацію запитів клієнтів до фронтальних сервісів та прогнозувати реакцію системи (майбутній трафік) на них. Контролер *SDN*, отримавши запит клієнта та виконавши його класифікацію оперує не стохастичними, а детермінованими даними, що дозволяє виконувати якісніше прогнозування. Дана модель короткочасного прогнозування може бути використанна в сервісах *QoS/QoE* та виявленню аномального характеру трафіку в системах моніторингу та захисту мережі. Для використання в маршрутизації трафіку пропонується використовувати гібридну модель в яку додатково вводиться прогнозування майбутніх клієнтських запитів.

Об'єктом дослідження є створення моделі, короткострокове прогнозування трафіку в якій здійснюється з використанням препроцесінгу класифікації клієнтських запитів та нейронної мережі прогнозування реакції системи. Предметом дослідження є розраховані прогнози трафіку між вузлами *SDN* мережі та їх відхилення від реальних значень.

Метою даного дослідження є отримання практичних результатів тестування, які продемонструють переваги запропонованого рішення та доцільність його використання в задачах керування *SDN* мережею.

Основна частина

У більшості рішень трафік мережі розглядають як безперервний у часі стохастичний процес. Однак згідно теореми розкладання Лебега з даного процесу можна визначити детерміновані компоненти. Зазвичай стохастичним є представлення первинних клієнтських запитів до ресурсів

(наприклад, до вебсторінок). Реакція вузлів мережі на такий запит має детермінований характер, який визначається параметрами запиту. До того ж дана реакція на окремий запит є досить обмежена в часі (мілісекунди або кілька секунд). Тому після класифікації запиту ресурсу ми можемо, на основі попереднього досвіду виконання аналогічного запиту, спрогнозувати реакцію системи на нього. Агрегування значень реакцій на запити в певний період часу дає нам короткостроковий прогноз трафіку на наступний інтервал часового відрізка. Дана ідея складає основу пропонуємого в даній роботі методу прогнозування трафіку.

Розглянемо модель *SDN* мережі, яка представляється у вигляді графа $G(N, U)$, де:

- $N = \{n_1, n_2, \dots, n_n\}$ – множина усіх вузлів *SDN* мережі;
- $U = \{(n_i, n_j) \mid n_i \in N \cap n_j \in N \cap i \neq j\}$ – множина логічних з'єднань вузлів.

Оскільки ми розглядаємо процес прогнозування, а не конструювання трафіку, в нашому випадку, це логічні з'єднання між кінцевими вузлами (без врахування фізичних з'єднань та шляху передачі пакетів). У якості вузла у нашому випадку також виступає не фізичний вузол мережі, а процес (сервіс доступу до ресурсу, служба, клієнтський додаток, тощо). Приклад моделі приведено на рис. 1.

Розкладемо множину вузлів на клієнтські процеси $C = \{c_1, c_2, \dots, c_n\}$, фронтальні сервіси $S = \{s_1, s_2, \dots, s_n\}$ та бекенд сервіси (генерації та обробки вторинних запитів) $B = \{b_1, b_2, \dots, b_n\}$. При цьому:

$$N = C \cup S \cup B.$$

Аналогічним чином розкладається і множина з'єднань:

$$U = CS \cup UR = CS \cup SC \cup SB \cup BS \cup BB.$$

де: $CS = \{(c_i, s_j) \mid s_i \in C \cap s_j \in S\}$ – первинні клієнтські запити, UR – реакція системи, $SC = \{(s_i, c_j) \mid s_i \in S \cap c_j \in C\}$ – відповідь на первинний клієнтський запит, $SB = \{(s_i, b_j) \mid s_i \in S \cap b_j \in B\}$ – вторинний запит від

фронтальної системи до бекенду, $BS = \{(b_i, s_j) \mid b_i \in B \cap s_j \in S\}$ – відповідь бекенду на вторинний запит, $BB = \{(b_i, b_j) \mid b_i \in B \cap b_j \in B \cap i \neq j\}$ – зв'язки між бекенд вузлами.

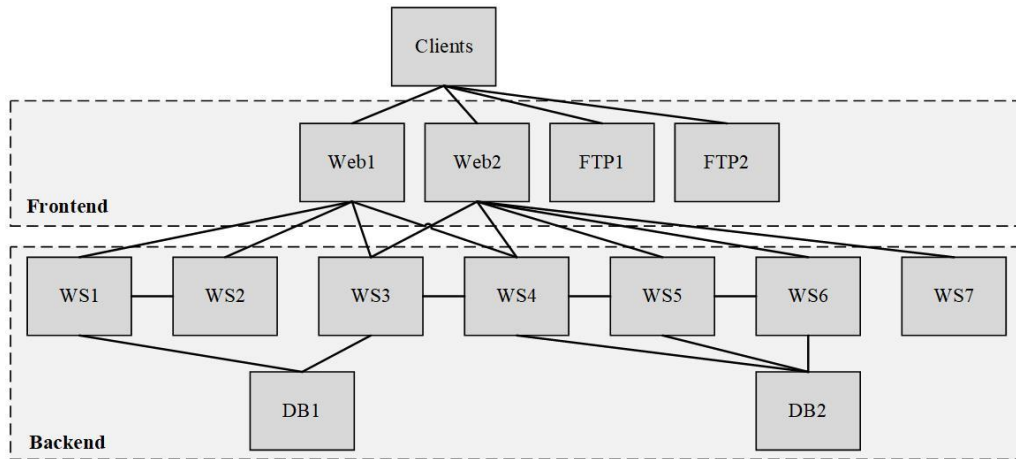


Рис. 1. Тестова модель мережі

Реакцію системи (прогнозований трафік) ми обчислюємо за допомогою *DNN*:

$$Y_{UR_T} = f_T(X_{CS}),$$

де: Y_{UR_T} – множина прогнозів трафіку для визначеної кількості точок прогнозування (горизонту прогнозування), f_T – функція *DNN*, X_{CS} – параметри запиту (вихідні дані класифікатора ресурсів).

Контролер *SDN* мережі отримавши клієнтський запит повинен виконати його класифікацію. В даній роботі для цього використовується препроцесор, який на основі аналізу пакету виконує класифікацію згідно профілю *http* (табл. 1) або профілю *ftp* (табл. 2). Дані профілі були використані в тестовій моделі для фронтальних *Web* та *FTP* серверів.

Таблиця 1. Профіль сервісу *http*

Назва	Тип	К-сть входів	Опис
<i>method</i>	бітова маска	8	<i>GET</i> (10000000), <i>POST</i> (01000000), <i>HEAD</i> (00100000), <i>PUT</i> (00010000), ...
<i>path1</i>	флаг+хеш	4	1 – присутній, 0 – відсутній хеш першого елемента в шляху запиту (3 входи)
<i>path2</i>	флаг+хеш	4	1 – присутній, 0 – відсутній хеш другого елемента в шляху запиту (3 входи)
<i>path3</i>	флаг+хеш	4	1 – присутній, 0 – відсутній хеш третього елемента в шляху запиту (3 входи)
<i>queryname1</i>	флаг+хеш	4	1 – присутній, 0 – відсутній хеш назви першого параметру запиту (3 входи)
<i>queryval1</i>	флаг+хеш	4	1 – присутній, 0 – відсутній хеш значення першого параметру запиту (3 входи)
<i>queryname2</i>	флаг+хеш	4	1 – присутній, 0 – відсутній хеш назви другого параметру запиту (3 входи)
<i>queryval2</i>	флаг+хеш	4	1 – присутній, 0 – відсутній хеш значення другого параметру запиту (3 входи)
<i>body</i>	флаг+хеш	4	1 – присутній, 0 – відсутній хеш перших 16 байт тіла запиту (3 входи)

Таблиця 2. Профіль сервісу *ftp*

Назва	Тип	К-сть входів	Опис
<i>cmd</i>	бітова маска	3	<i>GET</i> (10000000), <i>PUT</i> (01000000), <i>OTHER</i> (00100000)
<i>path1</i>	флаг+хеш	4	1 – присутній, 0 – відсутній хеш першого елемента в шляху запиту (3 входи)
<i>path2</i>	флаг+хеш	4	1 – присутній, 0 – відсутній хеш другого елемента в шляху запиту (3 входи)
<i>path3</i>	флаг+хеш	4	1 – присутній, 0 – відсутній хеш третього елемента в шляху запиту (3 входи)
<i>name</i>	флаг+хеш	4	1 – присутній, 0 – відсутній хеш назви файлу (3 входи)

Оскільки прогнози для окремих фронтальних систем є незалежними між собою, ми можемо їх розраховувати окремими *DNN*:

$$Y_{UR_T} = \{Y_{UR_{i_T}}\} \mid Y_{UR_{i_T}} = f_T(X_{S_{iC}}),$$

де, Y_{UR_T} – множина прогнозів трафіку для визначеної кількості точок прогнозування (горизонту прогнозування), $Y_{UR_{i_T}}$ – множина прогнозів для i -го фронтального сервісу, f_T – функція *DNN*, $X_{S_{iC}}$ – вихідні дані класифікатора ресурсів i -го фронтального сервісу.

Аналогічним чином можна виконати і декомпозицію по горизонту прогнозування (оскільки відсутній вплив прогнозу t_i на інші інтервали прогнозування):

$$Y_{UR_T} = \{Y_{UR_{T_j}}\} \mid Y_{UR_{T_j}} = f_T(X_{CS}),$$

де, Y_{UR_T} – множина прогнозів трафіку для визначеної кількості точок прогнозування (горизонту прогнозування), $Y_{UR_{T_j}}$ – прогноз для j -го відліку часу, f_T – функція *DNN*, X_{CS} – вихідні дані класифікатора ресурсів.

Оскільки в нашій моделі реакція системи є функцією тільки вихідних даних класифікатора ресурсів, ми можемо для розрахунку прогнозів окремих логічних з'єднань також використовувати незалежні *DNN*:

$$Y_{UR_T} = Y_{SC_T} \cup Y_{SB_T} \cup Y_{BS_T} \cup Y_{BB_T},$$

де Y_{SC_T} , Y_{SB_T} , Y_{BS_T} , Y_{BB_T} – прогнози для зв'язків відповідних підмножин *SC*, *SB*, *BS*, *BB*.

Архітектура пропонуємого рішення короткочасного прогнозування на базі реакції системи на клієнтські запити до фронтальних сервісів представлена на рис. 2.

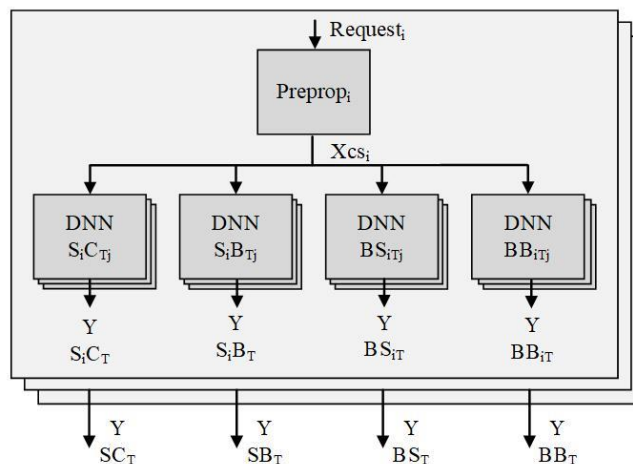


Рис. 2. Архітектура системи короткочасного прогнозування трафіку

Отримання тестових даних для навчання даної моделі в «ідеальних» лабораторних умовах є досить простою задачею. Необхідно організувати генерацію клієнтських запитів у вигляді черги з часовими проміжками між запитами не менше ніж максимальний час реакції системи на запит. У такому разі, отриманий SDN контролером стан мережі буде реакцією системи на одиночний запит. Однак цей алгоритм не спрацьовує в реальних умовах. Для кожного моменту часу t_k SDN контролер вимірює агрегований трафік від сукупності запитів, обробка яких ще не була завершена. Вичленити з такого трафіку складові окремих запитів неможливо. До того ж, оскільки запити почали виконуватися у різний час, кожен з них знаходиться на різному етапі виконання і виникає проблема синхронізації отриманих даних. Тому для навчання DNN в реальних системах (режимі *online*) пропонується використовувати модифіковану архітектуру рішення. У цьому випадку на кожній ітерації навчання DNN ми використовуємо множину запитів, які почалися не раніше $t_k - T$ та не пізніше t_k , де T – горизонт прогнозування. Крім того, в модуль препроцесінгу до параметрів кожного запиту додається також t_{start} – час старту. Препроцесор обчислює в які інтервали прогнозування T_j обробляється даний запит і передає для них вектори вихідних параметрів класифікатору

ресурсів X_{CS} , а для інших інтервалів передається вектор нульових значень:

$$j = \frac{t_k - t_{start}}{\Delta t},$$

$$X_{CS_{t_k}} = \begin{cases} X_{CS} | \forall SC_{T_j}, SB_{T_j}, BS_{T_j}, BB_{T_j} \\ 0 | \forall SC_{T_m}, SB_{T_m}, BS_{T_m}, BB_{T_m} | m \neq j \end{cases}$$

де, j – інтервалу уваги запиту, t_k – час тренувальної ітерації, t_{start} – старт запиту, Δt – інтервал між точками прогнозування, $X_{CS_{t_k}}$ – вихідні дані класифікатору запиту.

Прогнозовані агреговані значення трафіку обраховуються за формулою:

$$X_{a_{t_k}} = \{X_{CS_{t_k1}}, X_{CS_{t_k2}}, \dots, X_{CS_{t_kn}}\},$$

$$Y_{a_{UR_{t_k}}} = \sum_{i=1..n} Y_{UR_{t_k}i} = \sum_{i=1..n} f(X_{CS_{t_k}i}),$$

де, $X_{a_{t_k}}$ – множина активних запитів на момент t_k , $Y_{a_{UR_{t_k}}}$ – прогноз трафіку (агрегований), $Y_{UR_{t_k}}$ – прогноз для окремих запитів.

Для навчання DNN використовується зворотне поширення помилки, яка обчислюється за допомогою MSE (метод найменших квадратів):

$$MSE_{t_k} = \frac{1}{n} \sum_{i=1..n} (Y_{UR_{t_k}i} - \hat{Y}_{a_{UR_{t_k}i}})^2,$$

де, MSE_{t_k} – величина помилки, $Y_{UR_{t_k}}$ – дані з SDN контролера в момент часу t_k , $Y_{a_{UR_{t_k}}}$ – агрегований прогноз.

Модифікований варіант рішення наведено на рис. 3.

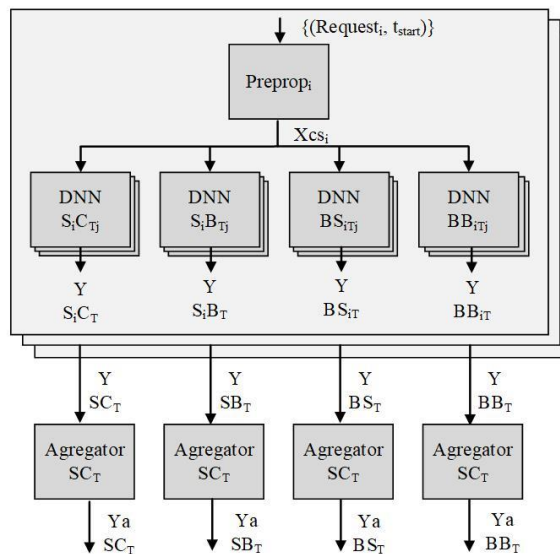


Рис. 3. Рішення з використанням агрегаторів

Для отримання тренувальних і тестових даних було розгорнута мережа зображена на рис. 1 у якій використано технологію контейнеризації *Docker*:

- *web1-web2, ws1-ws7*:
image: eclipse-temurin:17-jre-alpine
app: Apache Tomcat 10.1.18
- *ftp1-ftp2*:
image: garethflowers/ftp-server
app: vsftpd-3.0.4
- *db1-db2*:
image: mongo
app: MongoDB 7.0.8
- *client*:
image: eclipse-temurin:17-jre-alpine
app: OpenJDK 17.0.10+7

Для генерації клієнтських запитів з локальної мережі було знято статистику *http* запитів впродовж 3 діб (за допомогою *tcpdump*) та виконано їх конвертацію в запити для *web1-web2* за наступними типами: головна сторінка, статичні дані, сторінка ресурсу БД, сторінка результатів пошуку та модифікація ресурсу в БД. Аналогічним чином була зібрана статистика для відтворення запитів до *FTP* серверу (перегляд каталогу, завантаження, отримання файлу та його видалення).

Програми моделювання фронтальних серверів емулювали роботу наступним чином:

- статичні дані – завжди однаковим *URL* видавали однакові дані;
- запити до БД включали в себе дані *URL* та тіла запиту (при його наявності);
- запити до *REST* сервісів включали в себе дані клієнтських запитів з домішаною випадковою частиною.

Програми моделювання вебсервісів виконували запити до інших вебсервісів або СУБД (згідно рис.1). У якості БД було завантажено базу фільмів (1200 записів) та базу новин (800 записів).

Трафік між вузлами агрегувався в 100 мс інтервали часу.

Для розробки *DNN* було використано пакет *TensorFlow 2.16*. Препроцесор виконував класифікацію запитів ресурсів

згідно профілів сервісів, які приведені у таблицях 1 і 2. У якості блоків *DNN* були розроблені *CNN* з наступними характеристиками:

- *Conv1D (filters=32, kernel=3, 'relu')*;
- *Conv1D (filters=32, kernel=3, 'relu')*;
- *MaxPooling1D(pool=2)*;
- *Conv1D (filters=64, kernel=3, 'relu')*;
- *Conv1D (filters=64, kernel=3, 'relu')*;
- *MaxPooling1D(pool=2)*;
- *Conv1D (filters=128, kernel=3, 'relu')*;
- *Flatten*;
- *Dropout(0.2)*;
- *Dense (512, 'relu')*;
- *Dense(1, 'relu')*.

Модифікована версія рішення додатково включає блок батчу запитів ітерації навчання та вихідні блоки агрегації даних.

Додаткові характеристики *DNN*:

- *losses ('MeanSquaredError')*;
- *optimizer ('RMSProp')*;
- *metrics ('MAE', 'R2Score')*;
- *epochs = 20*.

Для класичного рішення видавався прогноз реакції системи на кожний окремий запит для наступних 8 інтервалів часу (800 мс). Для модифікованого рішення видавався прогноз наступних 8 інтервалів для вхідного вектору запитів з стартовим часом запуску кожного з них,

Для порівняння було виконано прогнозування на моделі нейронної мережі описаної в [5].

Навчання моделей проводилось на тренувальному наборі (248 754 інтервали), а тестування на тестовому наборі (73 019 інтервалів). Результати прогнозування ми оцінювали за допомогою коефіцієнту детермінації *R2* та середньої абсолютної помилки (*Mean Absolute Error*). Усереднені значення для різних типів зв'язків наведено у табл. 3, а приклад прогнозу для 20 інтервалів на рис. 4.

Таблиця 3. Результати тестування моделей прогнозування

Модель	SC		SB		BS		BB		TOTAL	
	R2	MAE	R2	MAE	R2	MAE	R2	MAE	R2	MAE
Classif	0.93	63	0.91	74	0.90	81	0.87	96	0.89	87
ClassifAgr	0.79	147	0.76	198	0.71	244	0.69	312	0.72	234
RRN	0.58	1187	0.57	1249	0.56	1338	0.54	1514	0.56	1356



Рис. 4. Приклад прогнозування трафіку

Кращі результати показані «Classif» та «ClassifAgr» для зв'язків SC (фронтальна система – клієнт) пояснюються статичними даними, які набагато легше спрогнозувати. Гірші результати «ClassifAgr» в порівнянні з «Classif» пояснюються похибками при виявленні в агрегованих даних компонент реакції на окремі запити.

Висновки

В роботі запропоновано рішення отримання короткострокового прогнозу трафіку SDN мережі методом прогнозування реакції системи на клієнтські запити ресурсів. Результати моделювання довели більш високу точність прогнозування даним методом на відміну від аналізу часових рядів. Запропонована архітектура має високий ступінь паралелізму та дозволяє динамічно переконфігурувати окремі вузли мережі або уточнювати їх DNN без необхідності перенавчати систему прогнозування в цілому. Дане рішення може включатися в комбіновану модель отримання короткострокового та довгострокового прогнозів.

Література

1. Joshi M., Hadi T. H. A review of network traffic analysis and prediction techniques. arXiv:1507.05722. 2015. 22 p.

2. Nyaramneni S., Saifulla M. A., Shareef S. M. ARIMA for traffic load prediction in software defined networks. *Lecture Notes on Data Engineering and Communications Technologies. Vol. 53. Evolutionary Computing and Mobile Sustainable Networks. Proceedings of ICECMSN 2020* / ed. by V. Suma, N. Bouhmala, H. Wang. Singapore, 2021. P. 815–824.

3. Yu B., Yang G., Yoo C. Comprehensive prediction models of control traffic for SDN controllers. *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)* : proceedings, Montreal, QC, Canada, 25–29 June 2018 / IEEE. 2018. P. 262–266.

4. Lim B., Zohren S. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*. 2021. Vol. 379, no. 2194. 20200209.

5. Azzouni A., Pujolle G. NeuTM: A neural network-based framework for traffic matrix prediction in SDN. *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium* : proceedings, Taipei, Taiwan, 23–27 April 2018 / IEEE. 2018. P. 1–5.

Кулаков Ю.О., Корочкін О.В., Омелянський А.С.

ВИКОРИСТАННЯ КЛАСИФІКАТОРА РЕСУРСІВ В МОДЕЛЯХ ПРОГНОЗУВАННЯ ТРАФІКУ SDN МЕРЕЖ

У даній роботі здійснено огляд існуючих рішень прогнозування трафіку SDN мереж. Запропоновано рішення використовує класифікатор клієнтських запитів до ресурсів для прогнозування реакції мережі на дані запити. Дане короткострокове прогнозування виконується за допомогою згорткової нейронної мережі. Розглянуто аспекти паралельних обчислень та прогнозування в динамічно реконфігурованій системі. Результати прогнозування можуть використовуватися в сервісах QoS/QoE, SDN контролером для маршрутизації трафіку та виявленню аномального характеру трафіку системами захисту мережі.

Ключові слова: прогнозування трафіку; класифікатор мережевих ресурсів; програмно-конфігуровані мережі; нейронна мережа.

Kulakov Y.O., Korochkin O.V., Omelyanskiy A.S.

USING A RESOURCE CLASSIFIER IN TRAFFIC FORECASTING MODELS FOR SDN NETWORKS

This publication reviews existing solutions for predicting SDN traffic. The proposed solution uses a resource classifier of client requests for predicting the network response to these requests. This short-term forecasting is performed using a convolutional neural network. The article considers aspects of parallel computing and forecasting in a dynamically reconfigurable system. Forecasting results can be used in QoS/QoE services, SDN controller for traffic routing and detection of anomalous traffic by network protection systems.

Keywords: traffic forecasting; network resource classifier; SDN; neural network.