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

№ 2 (4), 2022

https://doi.org/10.23939/ictee2022.02.012

ДОСЛІДЖЕННЯ МЕТОДУ АВТОМАТИЗОВАНОГО РОЗГОРТАННЯ МАШИННОГО НАВЧАННЯ НА ОСНОВІ ТЕХНОЛОГІЙ CI/CD

В. Федорченко, О. Красько, І. Демидов, Р. Колодій

Національний університет "Львівська політехніка" вул. С. Бандери, 12, Львів, 79013, Україна

Відповідальний за рукопис: В.Федорченко(e-mail: valfedorch@gmail.com).

(Подано30 жовтня 2022)

У цій статті запропоновано метод автоматизованого розгортання алгоритмів машинного навчання на основі програмного продукту Splunk Enterprise та додатку для нього Splunk Machine Learning Toolkit. Реалізація цього методу дасть можливість розгортати системи ML в найкоротші терміни, вносити зміни до її структурних одиниць з мінімальним впливом на інші складові та адаптувати моделі ML до змін у вхідних даних, переносити систему до іншого середовища чи постачальника хмарних послуг. Перевагою використання цього методу є можливість відслідковувати активність користувачів та, за необхідності виявляти аномалії у їх поведінці. Аномалії виявляються серед даних системних/аудит логів. Після встановлення потрібних джерел даних на серверах для моніторингу, дані будуть отримані на індексері та стануть доступними для подальшої обробки йнавчання моделі ML.

Ключові слова: *машинне навчання, модель, Machine Learning, Splunk.*
УДК 621.126

1. Вступ

Сьогодні" понад 59 % населення світу отримують ті чи інші послуги через мережу Інтернет. Хоча це і зробило наше життя швидшим та зручнішим, однак наражає на певні небезпеки, зокрема розголошення конфіденційної інформації, фінансове шахрайство [1]. Саме безпека даних зумовила стрімкий розвиток методів та алгоритмів машинного навчання. Зі збільшенням популярності використання різних типів машинного навчання зростає й потреба у постійній розробці, моніторингу, та впровадженні їх у існуючі системи, що не завжди здатні одразу надавати дані для обробки. Тому першочерговим завданням є отримання зрозумілого, легкого в освоєнні та повторенні алгоритму, що у результаті виконання зможе надавати послугу детектування аномалій, чи то вразливостей з мінімальним впливом на існуючу інфраструктуру, і в одночас, здатний обробляти дані з мінімальними затримками.

Машинне навчання використовується багатьма програмними засобами для оптимізації та автоматизації рутинних процесів. Для тренування та подальшої роботи моделей машинного навчання (ML) необхідно мати великий обсяг "якісних" даних. Дані, генеровані користувачами інформаційної системи (ІС), обов'язково проходять обробку та збір у програмних засобах SIEM, що залежно від використаного підходу обробки робить їх одною з найкращих баз для навчання моделей ML.

Однак для внесення змін у параметри, адаптації до нових даних чи банальної зміни версії моделі адміністратору потрібно особисто виконувати перераховані операції. У випадку переходу на нове програмне забезпечення чи інше середовище, перехідний період триватиме днями, а можливо

й тижнями, що ще більше загострює проблему автоматизації процесів, оскількиосновна увага інженерів буде прикута до відлагодження програмного продукту, що є витратним в умовах динамічного розгортання сервісів.

Саме тому потрібно запропонувати метод роботи з системою машинного навчання, що дасть змогу розгортати її в найкоротші терміни, вносити зміни до її структурних одиниць з мінімальним впливом на інші складові, дасть можливість адаптувати моделі ML до змін у вхідних даних та за потреби переносити систему до іншого середовища чи постачальника хмарних послуг. Таку можливість може надати методологія CI/CD. Ця методологія включає великий набір інструментів для вирішення різних завдань із серверними системами, таких як: автоматизоване розгортання серверів, їхня конфігурація та швидкий доступ до зміни їхніх налаштувань.

2. Аналіз існуючих досліджень на шляху до розв'язання проблеми

На ринку надання послуг з інтеграцією машинного навчання існує доволі велика кількість програмних продуктів, що надають послуги розгортання вже готових рішень: TensorFlow, Apache Spark MLlib чи Amazon Machine Learning. Більшою мірою організація управління збором, агрегацією, навчанням моделі MLта валідацією результатів виконується якщо не вручну, то використовуючи стандартні функції обраного середовища для роботи. Питання сумісності різних сервісів, постійна потреба бути на зв'язку виконавця та замовника та дуже велика інерційність процесу отримання послуг змушують компанії вдаватись до розробки нових методів та алгоритмів для машинного навчання.

Ситуація з виконанням гнучкої реконфігурації оброблюваних даних, чи адаптивного використання моделі до нових даних є досить складною, бо створюється потреба адаптувати такий алгоритм. Якщо виникає потреба у масштабуванні, чи тестуванні вихідних даних, використання різних фреймворків є не дуже гнучким, оскільки основна вимога від моделі ML–це постійно бути в актуальному стані та не залежати від середовища. Як наслідок, основна складність роботи полягає у необхідності відлагодити роботу усіх складових ІС між собою.

Через популярність, гнучкість та велику функціональність cloud-сервісів компанії переводять свої проєкти у хмару, або ж одразу будують проєкт на ньому. Для середовищ, яким не потрібне використання ML, це є одним із найкращих рішень. Але для тих, хто використовує аналіз даних за допомогою ML, чи для безпеки, чи то для аналізу інших метрик, буде великим випробуванням міграція з одного середовища у інше, як, наприклад, перехід з AWS до GCP або Azure. Основною проблемою міграції з одного Cloud до іншого – є сумісність та оптимізація.

Наприклад, для обробки сирих даних користувачів можна використати рішення Google: Dataflow – обробка даних та перетворення їх у таблицю CSV, AI Platform Training – для виконання навчання та створення моделі та AI Platform Predictions – для розгортання створеної моделі. За допомогою GCP весь процес перетворюється у використання попередньо сконфігурованих складових з мінімальним написанням коду та втручанням в нього. Цей спосіб розгортання є неймовірно зручним та на початках заворожує простотою, адже: натиснув кнопку – і все запрацювало! Але, якщо розглядати архітектурне рішення про перехід до іншого постачальника cloud-послуг та дослідивши можливість міграції з GCP до AWS, буде зрозуміло, що на AWS є відповідники використаних продуктів на GCP: Google Dataflow–Amazon Data Firehose, AI Platform Training–Amazon SageMaker та інші. Ці продукти є аналогами один одного, але конфігурування та відлагодження виконується через різні середовища, ледь не всі напрацювання зустрічають проблему сумісності. Попередньо представлене рішення є доволі спрощеним для розуміння самого процесу роботи ML у хмарному середовищі. Отримуємо дуже зручний засіб отримання бажаного результату, але прив'язуємо себе до одного конкретного постачальника хмарного рішення [24].

Представлений вище аналіз зводиться до необхідності у ефективному балансуванні між довгою розробкою продукту і використанням переваг віртуалізації в хмарному середовищі. Як наслідок, необхідно запропонувати новий метод автоматизованого розгортання алгоритмів машинного

навчання, що дасть змогу гнучко працювати і з хмарною інфраструктурою, і на фізичних серверах, та дозволятиме проводити моніторинг даних до та після їх агрегації, виконуватиме автоматичне навчання ІС та у випадку порушення безпеки даних дозволить запускати механізми їх захисту.

3. Метод автоматизованого розгортання машинного навчання на основі технологій CI/CD

Збір даних, тестування даних, управління ресурсами, з використанням великомасштабних обчислювальних ресурсів та використання результуючих даних для подальшої аналітики, візуалізації чи активації дій у розробленій системі, є частинами "конвеєрів" машинного навчання. Найзручнішим рішенням для прискорення розгортання та відлагодження середовища MLє створення чіткого алгоритму дій, в якому автоматизовано будуть виконуватись операції збору та обробки даних, навчання та тестування моделі машинного навчання, а також виконання необхідних дій на основі вихідних даних. Найкраще для організації такого алгоритму підходять практики CI/CD.

Такий підхід гарантує, що вихідні моделі можуть давати готові до експлуатації, надійні результати, які розвиватимуться з часом, використовуючи різні форми інфраструктури як в хмарі, так і в локальних середовищах. Моделі, з іншого боку, ніколи не є постійними структурами. Вони постійно розвиваються у відповідь на нові дані, оскільки розпад моделі вимагає перенавчання вже з новими даними[3]. Безперервну інтеграцію та розгортання можна використовувати для створення безперервного циклу зворотного зв'язку, що гарантує, що моделі оновлені та правильні, не вимагаючи постійного моніторингу чи втручання.

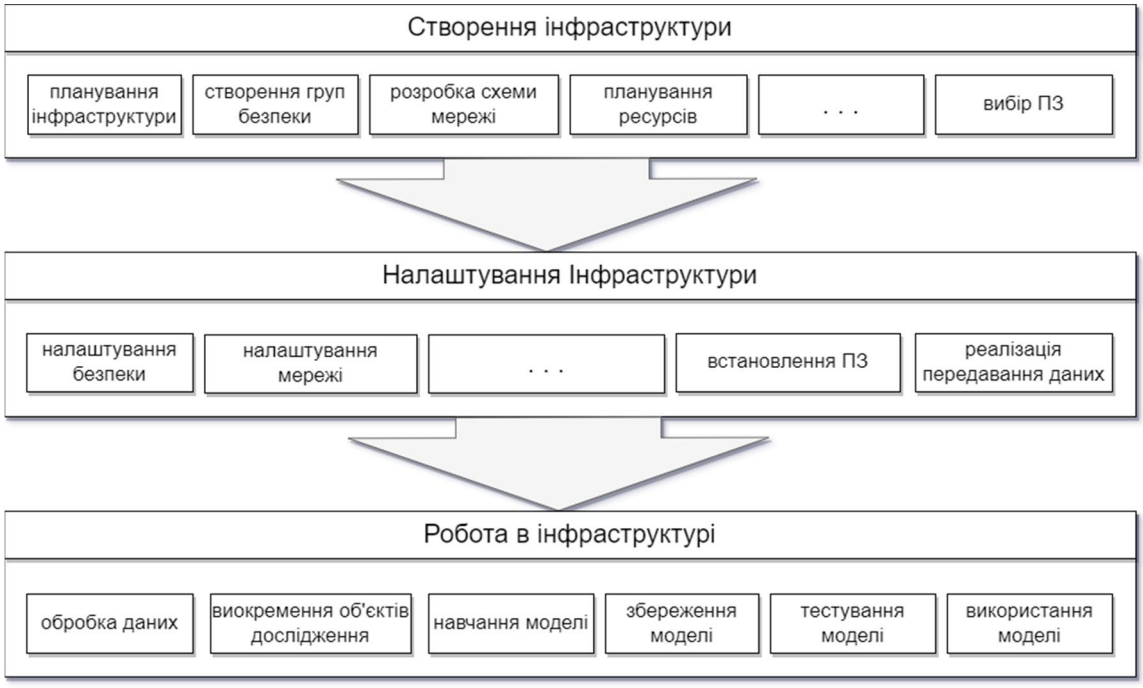


Рис. 1. Пропонована схема розгортання середовища та створення ML

За схемою, запропонованою на рис.1, можна виконати розгортання та налаштування середовища для експлуатації машинного навчання багатьма способами, використовуючи різні програмні засоби, та створити систему різної складності. Може бути представлена багатосерверна архітектура з великою кількістю точок входу, та навпаки, створена мінімізована структура, що буде обробляти дані лише на одному пристрої[4].

- Під час розробки системи вносяться чисельні корективи та редагування у план інфраструктури. Тому перш ніж приступати до реалізації інфраструктури, потрібно визначити її архітектуру та

складові частини, потрібно створити правила безпеки для кожного члена мережі, відповідно створити саму схему мережі між ресурсами, дослідити достатність чи надлишковість обраних потужностей в ресурсах, що зазвичай виконується залежно від обраного ПЗ.

- На другому етапі, коли створено детальний план інфраструктури та узгоджено усі нюанси її розгортання – усі схеми починають реалізовуватись у тому чи іншому вигляді. Виконується початкова конфігурація доступів до мережі від одних її членів до інших, створюються правила доступів та проходження даних, встановлюється ПЗ та отримуються перші дані про роботу середовища.
- На третьому етапі починається обробка даних, що генеруються за використання інфраструктури, дані зберігаються, агрегуються, з ними виконуються перетворення та з виокремленими об'єктами дослідження виконується навчання моделі машинного навчання з подальшим її збереженням, тестуванням та релізом.

Для вирішення задачі мультиплатформеності, гнучкості та простоти мігрування з одного середовища в інше використано Splunk Enterprise, що може використовуватись як збирач, агрегатор даних, для їх аналізу як SIEM. Разом зі Splunk використовується додаток для нього Splunk Machine Learning Toolkit, в сумі вони дають змогу збирати живі дані з різних джерел формувати їх у необхідний формат, робити підготовку даних та виконувати на попередньо оброблених даних навчання моделей машинного навчання.

Насамперед, розгортання серверів та всієї мережі, з якою буде виконуватись робота, можна реалізувати за допомогою різних інструментів, що дають можливість створювати IaaC. Для демонстрації використано Terraform, з його допомогою створено групу безпеки на AWS, і в ній розгорнуто п'ять віртуальних машин, чотири з яких умовно користувацькі та одна буде використана для обробки даних. За допомогою Ansible виконано конфігурацію створених інстансів: виконано налаштування мережі та фаєрволів, встановлено необхідне ПЗ та інші супутні налаштування серверів.

Для дослідження використовувались стандартні аудит-логи, які генеруються в ОС Linux, для їхньої агрегації використовувались Splunk Enterprise та Splunk UF. Splunk Enterprise використовувався у ролі декількох складових: Indexer та Search head з вебінтерфейсом, а Splunk UF –звичайний форвардер, що є агентом, встановленим на сервері, та виконує збір даних, які встановлені на моніторинг.

- Indexer: його основною задачею є збереження та агрегація даних, що далі будуть занесені в окремі індекси для покращення ефективності пошуку.
- Search head: виконує роль графічного інтерфейсу, уможливлюєосновнувзаємодію користувача з даними через search engine.
- Universal Forward: це легкий компонент, який пересилає дані логів, журналів або будь-яких файлів, які мають часову мітку, до Splunk Indexer. Він встановлюється на сервері додатків або на стороні клієнта.

Для виконання задач машинного навчання використано додаток до Splunk Enterprise–Machine Learning Toolkit. Набір інструментів машинного навчання (MLTK) дає змогу користувачам створювати, перевіряти, керувати моделями машинного навчання та впроваджувати їх у дію через зручний інтерфейс. Великим плюсом MLTK є те, що в ньому вже є понад 30 підготованих алгоритмів на мові Python, а також підтримуються сторонні алгоритми користувача [5–7].

Будь-який аналіз даних з використанням машинного навчання виконується в декілька етапів:

- Для збору даних з джерел для подальшої обробки можуть використовуватись різні відомості, наприклад, про активність користувача у мережі, або логи, згенеровані під час роботи програм. Тому й способи їх отримання та передачі можуть відрізнятись.
- Збереження даних може виконуватись у різний спосіб та в різних сховищах, наприклад, у базі даних чи в озері даних, вони відрізняються способом збереження інформації та їхнім поданням для подальшої обробки.

- Перед початком навчання моделі ML виконується попередня обробка даних для того, щоб позбутись надлишковості у нихта, відповідно, зменшити вірогідність некоректного спрацювання моделі.
- Перш ніж використовувати модель, потрібно виконати її навчання та тестування. Процес може бути зацикленим, адже для подальшого використання моделі потрібно досягти найточнішого її спрацьовування.

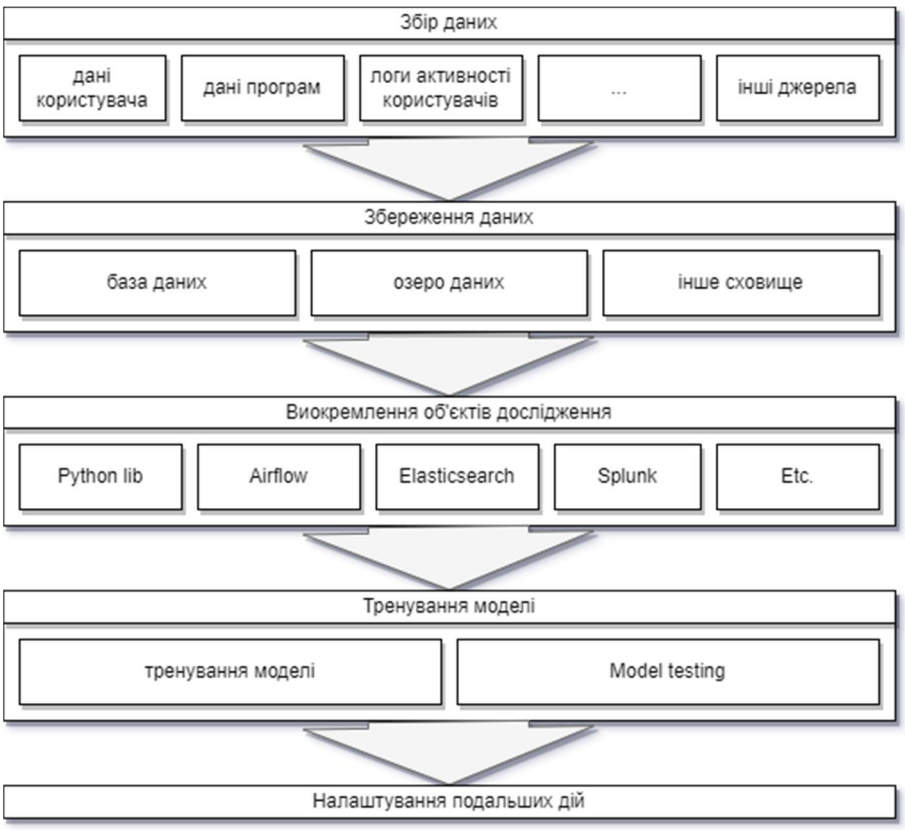


Рис. 2. Схема обробки зібраних даних

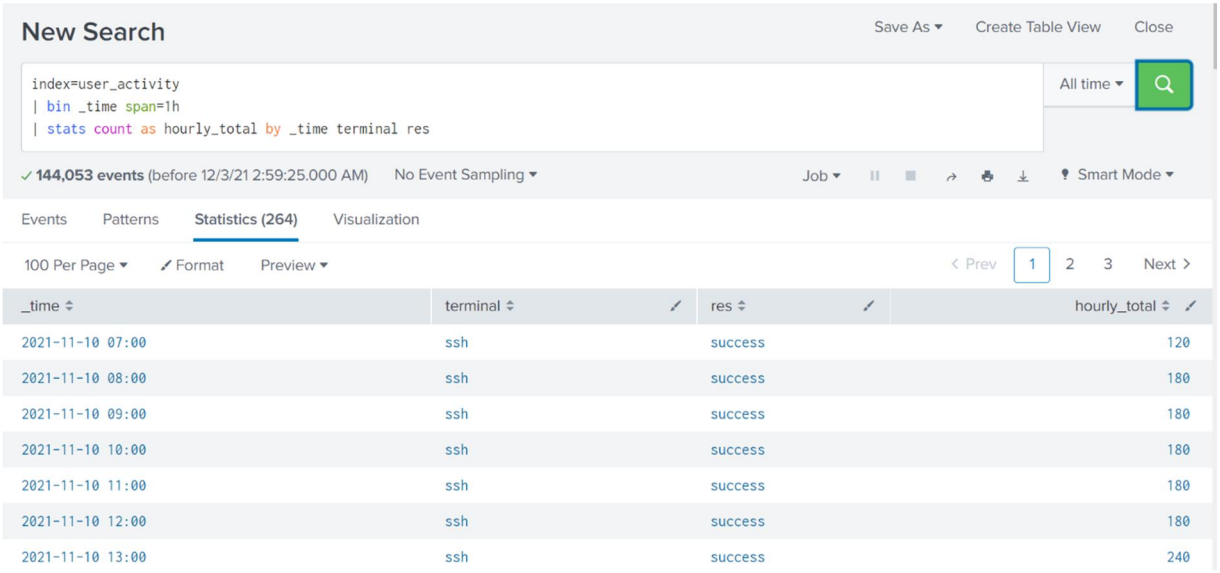
На рис. 2 наведено загальну схему обробки та використання даних до та після обробки моделлю. У традиційному виконанні для реалізації схеми необхідний цілий програмний комплекс з декількох різних пакетів ПЗ. Для вирішення цього завдання використаємо програмне середовище Splunk, що виконує усі функції зі збору, аналізу, навчання моделі ML (виконання стороннього коду, відправка сповіщення, формування іншої аналітики). Перевагою використання цієї системи є можливість відслідковувати активність користувачів та, за необхідності, виявляти аномалії у їх поведінці. Аномалії виявляються серед даних системних/аудит-логів. Після встановлення потрібних джерел даних на серверах для моніторингу дані будуть отримані на індексері та стануть доступними для подальшої обробки та навчання моделі ML.

4. Дослідження ефективності методу автоматизованого розгортання машинного навчання

Для розгортання розроблюваного середовища використано хмарні сервери AWS. За допомогою Terraformстворено п’ять інстансів типу t2.micro з використанням AMILinux. Один з хостів використано як Search head та Indexer, інші чотири використовувались для імітації роботи користувачів. У terraform-скриптітакож створено групу безпеки та налаштовано мережеві взаємозв’язки між користувацькими хостами та Search head. За допомогою Ansibleвиконані налаштування вже на са-

мих хостах, такі як мережа, харденінг та створення користувачів. Ansibleplaybooktакож встановив агенти (SplunkForwarder) для передачі користувацьких логів до індексера. На хості Search head всновлено SplunkEnterpriseдля попередньої обробки даних та додаток до Splunk Enterprise–Machine Learning Toolkit для навчання моделей машинного навчання і подальшого їх використання.

Виконавши попереднє налаштування середовища, було відпрацьовано декілька імітацій bruteforce-атак на користувацькі сервери,що являло собою підбір паролів для входу черезSSHдо користувацьких хостів. Це дало можливість зібрати дані активності користувачів. Одразу після налаштування середовища почався збір користувацьких даних та збереження подій до індексів у Splunk. Дослід проводився вподовж 18 днів, отримані дані зводяться до табличного вигляду, та зі свіжих даних виконується виокремлення полів, що будуть використовуватись у подальшій аналітиці. Також потрібно виокремити дані для навчання моделі ML (рис. 3).



_time	terminal	res	hourly_total
2021-11-10 07:00	ssh	success	120
2021-11-10 08:00	ssh	success	180
2021-11-10 09:00	ssh	success	180
2021-11-10 10:00	ssh	success	180
2021-11-10 11:00	ssh	success	180
2021-11-10 12:00	ssh	success	180
2021-11-10 13:00	ssh	success	240

Рис. 3. Статистична таблиця з даними для навчання ML

На основі даних, сформованих в табличному вигляді у Splunk MLTK, потрібно розпочати створення моделі машинного навчання. Під час підготовки до навчання моделі ML SplunkMLTK виведе графік гістограмирозподілу відхилень кількості подій від медіани (рис. 4).

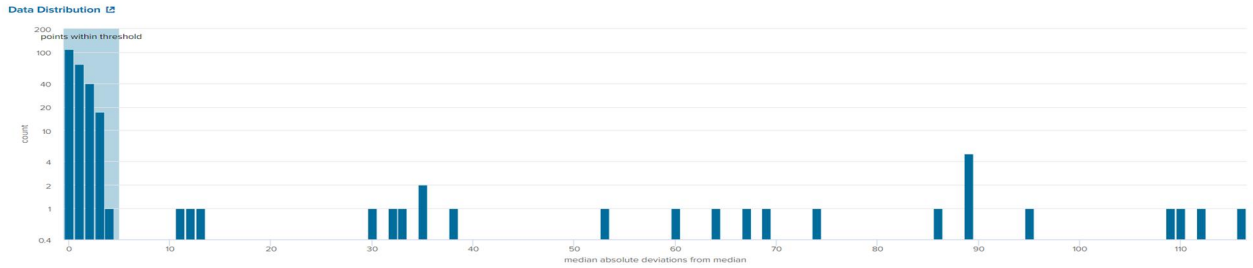


Рис. 4. Гістограма відхилень

Ознайомившись з гістограмою на рис. 4, потрібно звернути увагу на область, виділену синім – це частина графіка, де переважно спостерігається тип розподілу подій відносно медіани.Вибираємо експоненційний тип розподілу подій. Підбираючи значення порогу чутливості для моделі, переходимо до навчання моделі. Завершивши навчання, програма будує графік (рис. 5) з кількісною характеристикою активності користувачів та підсвічує кількісне відхилення від нормальної.

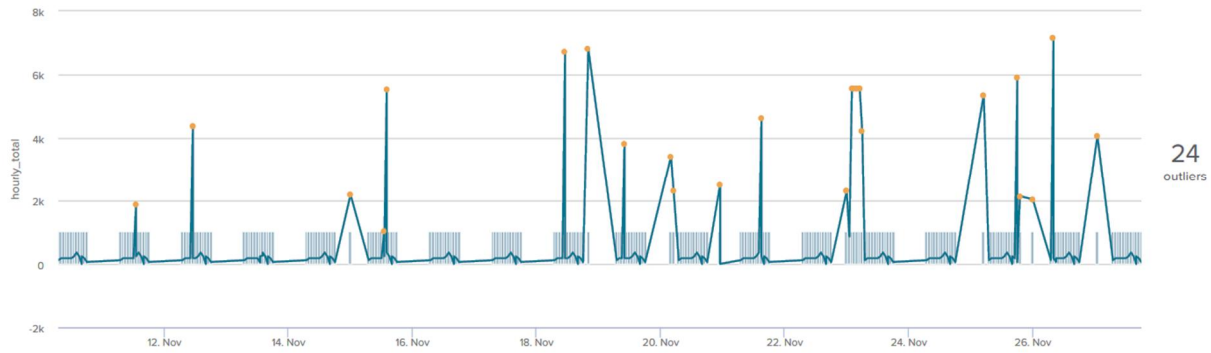


Рис. 5. Розпізнані аномалії серед активності користувачів

У фіналі створення системи розпізнавання аномалій потрібно зберегти модель ML та виконувати потрібні дії використавши команду "…| apply *назва моделі*". Splunk та Splunk MLTK.

На рис. 5 видно "сплески" кількості спроб входу на користувацькі хости. Верхні точки підсвічені жовтим –аномалії, розпізнанні моделлю ML. Тобто, з упевненістю можна стверджувати, що розгорнута система довела працездатність. Використаний програмний стекAWS, Terraform, Ansible, Splunk дав підтвердження, що автоматизація розгортання машинного навчання створює економію в часових та людських ресурсах, дає можливість впроваджувати зміни у системну складову та в складову обробки даних, адже функціонал використаних програм передбачає такий функціонал.

Висновки

Розроблено метод автоматизованого розгортання машинного навчання на основі програмного продукту Splunk Enterprise та додатку для нього Splunk Machine Learning Toolkit, що дало змогу зручно організувати збір та обробку даних, виконати дослідження оброблених даних та створити модель машинного навчання, яку можна інтегрувати до реальних процесів у виробництві чи в інших корпоративних рішеннях. Перевагою використання цієї системи є можливість відслідковувати активність користувачів та за необхідності виявляти аномалії у їхній поведінці. Аномалії виявляються серед даних системних/аудит-логів. Після встановлення потрібних джерел даних на серверах для моніторингу дані будуть отримані на індексері та стануть доступними для подальшої обробки й навчання моделі ML. Спосіб розгортання середовища та налаштування машинного навчання є найзручнішою на сьогодні методикою, адже підхід до розгортання інфраструктур серверів та вебдодатків вже більше десятка років використовується інженерами напрямку DevOps. Розгортання системи машинного навчання, використовуючи програмні продукти, представлені у роботі, значно спрощує процес навчання моделі машинного навчання та подальше її використання.

Список використаних джерел

[1] *ViolinoB. (2018). 6 ways to make machine learning fail. [Electronic resource]. URL: InfoWorld. https://www.infoworld.com/article/3310076/6-ways-to-make-machine-learning-fail.html.*

[2] *6 Steps to Migrating Your Machine Learning Project to the Cloud. [Electronic resource]. URL: https://towardsdatascience.com/6-steps-to-migrating-your-machine-learning-project-to-the-cloud-6d9b6e4f18e0.*

[3] *Aggarwal A. and JaloteP. (2006) "Integrating static and dynamic analysis for detecting vulnerabilities", in 30th Annual International Computer Software and Applications Conference (COMPSAC'06). Pp. 343–350.*

[4] *GriecoG., GrinblatG. L., UzalL. C., RawatS., FeistJ., and MounierL. (2016) "Toward large-scale vulnerability discovery using machine learning", in Proc. CODASPY, New Orleans, LA, USA. Pp. 85–96.*

[5] *Copyright (c) Splunk Inc. (2021) Welcome to the Machine Learning Toolkit [Electronic resource]. URL: https://docs.splunk.com/Documentation/MLApp /5.3.0/User/WelcometoMLTK.*

[6] Copyright (c) Splunk Inc. (2021) About the Machine Learning Toolkit [Electronic resource]. URL: https://docs.splunk.com/Documentation/MLApp/5.3.0/User /AboutMLTK.

[7] Karthik S., Tyler W. (2019). Detecting and Mitigating Insider Threats Using MLTK and Enterprise Security. [Electronic resource]. URL: https://conf.splunk.com/ files/2019/slides/SEC1305.pdf.

RESEARCH OF THE CI/CD APPROACH ADAPTATION POSSIBILITIES TO THE DEVELOPMENT OF MACHINE LEARNING MODELS

V. Fedorchenko, O. Krasko, I. Demydov, R. Kolodiy

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

In this paper we proposed a method for automated deployment of machine learning algorithms based on the Splunk Enterprise software product and the Splunk Machine Learning Toolkit application for IT. The implementation of this method will make it possible to deploy ML systems in the shortest possible time, make changes to its structural units with minimal impact on other components and adapt ML models to changes in input data, transfer the system to another environment or cloud service provider. The advantage of using this method is the ability to monitor user activity and, if necessary, detect anomalies in their behavior. Anomalies are detected among system/audit log data. After installing the required data sources on the servers for monitoring, the data will be received on the indexer and will be available for further processing and training of the ML model.

Keywords: *Machine Learning, model, Splunk.*