Технічні науки

УДК 004.852

**Ловчинський Сергій Броніславович**
студент
Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»
**Ловчинский Сергей Брониславович**
студент
Национального технического университета Украины
«Киевский политехнический институт имени Игоря Сикорского»
**Lovchinsky S.**
student
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"

# АНАЛІЗ ПОВІДОМЛЕНЬ СОЦІАЛЬНОЇ МЕРЕЖІ ДЛЯ ВИЯВЛЕННЯ ПОДІЙ ЗА ДОПОМОГОЮ APACHE SPARK
# АНАЛИЗ СООБЩЕНИЙ СОЦИАЛЬНОЙ СЕТИ ДЛЯ ОБНАРУЖЕНИЯ СОБЫТИЙ С ПОМОЩЬЮ APACHE SPARK
# SOCIAL NETWORK ANALYSIS TO DETECT EVENTS USING APACHE SPARK

**Анотація**: Стаття присвячена дослідженню задачі аналізу потоку повідомлень соціальної мережі з метою виявлення подій в реальному часі використовуючи систему обробки даних Apache Spark.

Показано, інтеграція системи Apache Spark із соціальною мережею Twitter, яка виступає в ролі джерела даних. Визначено особливості потокової обробки даних за допомогою Spark Streaming. Продемонстровано аналіз повідомлень з використанням Spark MLlib.

**Ключові слова**: Потокова обробка, аналіз даних, великі дані, Apache Spark, Spark Streaming, MLlib.

**Аннотация**: Статья посвящена исследованию задачи анализа потока сообщений социальной сети с целью выявления событий в реальном времени используя систему обработки данных Apache Spark.

Показано, интеграция системы Apache Spark с социальной сетью Twitter, которая выступает в качестве источника данных. Определены особенности поточной обработки данных с помощью Spark Streaming. Продемонстрировано анализ сообщений с использованием Spark MLlib.

**Ключевые слова:** Потоковая обработка, анализ данных, большие данные, Apache Spark, Spark Streaming, MLlib.

**Abstract**: The article is devoted to the investigation of the task of analyzing the flow of social network messages in order to identify events in real time using the Apache Spark data processing system.

It shows the integration of the Apache Spark system with the social network Twitter, which acts as a data source. Defined specific features of stream processing with Spark Streaming. Demonstrated messages analysis using Spark MLlib.

**Keywords**: Stream processing, data analysis, large data, Apache Spark, Spark Streaming.

**Постановка проблеми.** Важливою характеристикою соціальних мереж являється їх реальний характер. Наприклад, коли стається природній катаклізм, люди створюють велику кількість повідомлень, які пов'язані з ним. Аналіз повідомлень із соціальних мереж, дозволить швидко реагувати на небезпечні події, а також запобігати негативним наслідкам. Тому обробка інформації має виконуватись у режимі реального часу, для того щоб надати важливу інформацію з короткою затримкою.

**Метою статті** є дослідження побудови системи аналізу повідомлень соціальної мережі для виявлення цільових подій за допомогою Apache Spark. Для реалізації цього завдання поставлені наступні задачі:

1. Дослідити принципи та вимоги до побудови систем потокового оброблення даних в реальному часі.
2. Проаналізувати можливі застосування Spark Streaming.
3. Розглянути підхід Apache Spark до вирішення задач потокового оброблення даних.
4. Розробити програмний додаток для кластеризації вхідних повідомлень за вказаними темами.

**Виклад основного матеріалу дослідження.** Система реального часу – це система, яка повинна реагувати на події у зовнішньому відношенні до середовища або впливу на середовище в рамках необхідних тимчасових обмежень. [1] В контексті дослідження, подія – це довільна класифікація простору або часової області, яка може бути визначена шляхом аналізу повідомлень соціальної мережі. Ці події мають кілька властивостей:

- Мають великий масштаб (багато користувачів відчувають цю подію);
- Впливають на повсякденне життя людей (з цієї причини спонукають цитувати їх);
- Мають як просторові, так і тимчасові регіони (таким чином можна буде оцінити місце розташування в реальному часі);

Щоб класифікувати отримані повідомлення, в яких можуть міститись згадки про цільову подію, у позитивний клас або негативний клас, потрібно використати метод опорних векторів (SVM), який є широко використовуваним алгоритмом машинного навчання. [2] В машинному навчанні метод опорних векторів – це метод аналізу даних для класифікації та регресійного аналізу за допомогою моделей керованого навчанням з пов'язаними алгоритмами навчання, які називаються опорно векторними машинами.

Підготувавши позитивні та негативні приклади як навчальний набір, можна створити модель класифікації повідомлень у позитивні та негативні

категорії. Для цього потрібно підготувати наступні групи особливостей для кожного повідомлення: [3]

1. Визначається кількість слів у повідомленні та позиція слова запиту
2. Визначається за вказаним ключовим словом
3. Визначається контекст слова в тексті

Для побудови системи аналізу повідомлень соціальної мережі з метою виявлення цільових подій буде використано Apache Spark та його компоненти Spark Streaming, MLlib.

Spark Streaming – компонент Apache Spark для обробки потокових даних. Прикладами джерел таких даних можуть служити файли журналів, які заповнюються діючими веб-серверами, або черги повідомлень, що посилаються користувачами веб-служб. Spark Streaming – це модуль в складі Spark, призначений для створення додатків потокової обробки даних з використанням API, який дуже схожий на той, що застосовується в пакетних завданнях (batch jobs), що полегшує реалізацію, оскільки потребує тих самих навичок програмування. [4] Spark Streaming має API для керування потоками даних, яке близько відповідає моделі незмінній розподіленій колекції елементів, яка використовується для представлення розподілених даних і результатів обчислень, що підтримується компонентом Spark Core, які зберігаються в пам'яті, на диску або надходять в режимі реального часу. Прикладний інтерфейс (API) компонента Spark Streaming розроблявся з ціллю забезпечити таку ж надійність, пропускну здатність і масштабованість, що і Spark Core. На рисунку 1 зображено схему процесу роботи Spark Streaming.

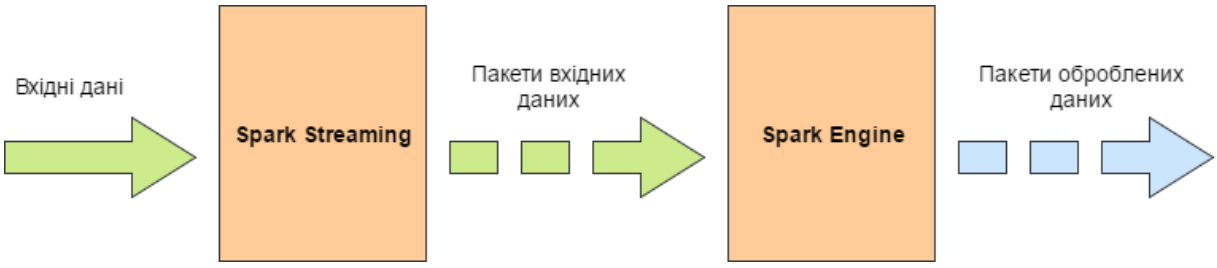


Рисунок 1 – Схема процесу роботи Spark Streaming

Подібно до того, як весь фреймворк Spark побудований на понятті наборів даних RDD, Spark Streaming надає власну абстракцію, яка називається DStreams, або Discretized Streams (дискретизовані потоки). DStream – це послідовність даних, яка надходить за деякий інтервал часу. Внутрішньо кожен потік DStream представлений послідовністю наборів RDD, які надійшли за інтервал часу. Потоки DStream можуть створюватися на основі будь-яких джерел даних, таких як Flume, Kafka або HDFS. [5] Після створення вони пропонують два типи операцій: перетворення, які породжують нові потоки DStream, і операції виведення, що записують дані в зовнішні системи. Потоки DStream підтримують більшість операцій з тих, що доступні для наборів RDD, а також нові операції, пов'язані з часом, такі як визначення ковзного вікна. На відміну від програм пакетної обробки, додатки на основі Spark Streaming потребують додаткового налаштування, щоб працювати постійно в безперебійному режимі.

MLlib – це бібліотека функцій машинного навчання (machine learning), що входить до складу Spark. Призначена для використання в кластерах, бібліотека MLib містить реалізації різних алгоритмів машинного навчання і може використовуватися у всіх мовах програмування, підтримуваних фреймворком Spark. [6] Бібліотека MLib має дуже просту архітектуру і філософію: вона дозволяє застосовувати різні алгоритми до розподілених масивів даних, представлених у вигляді наборів RDD. Схема на рисунку 2 демонструє алгоритм виконання типових задач з машинного навчання за допомогою MLlib.

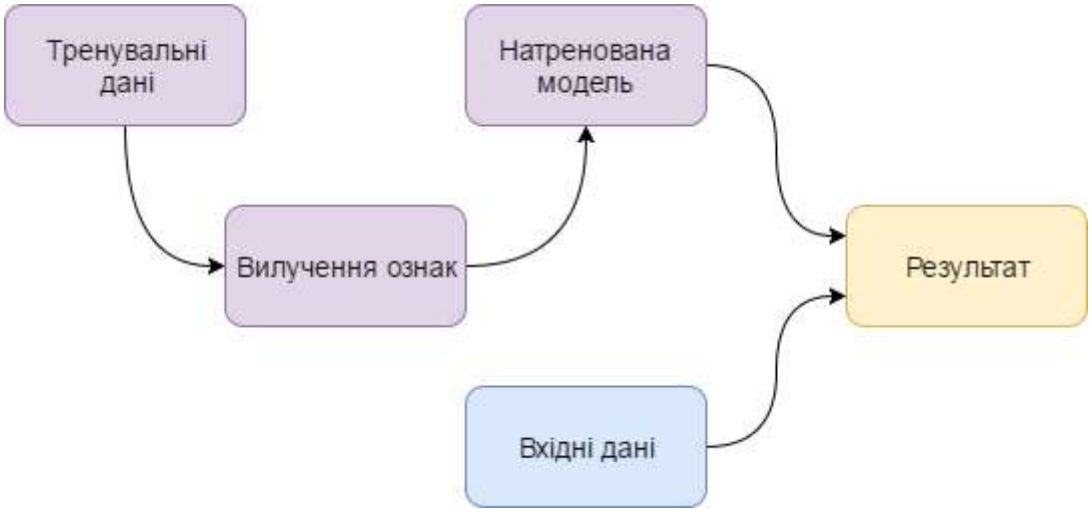


Рисунок 2 – Алгоритм роботи бібліотеки MLlib

В першу чергу програма має створити об'єкт StreamingContext, що є головною точкою входу в механізм потокової обробки. При цьому автоматично буде створено об'єкт SparkContext, який використовується для обробки даних. [7] Конструктор StreamingContext() приймає інтервал часу, який визначає, як часто повинні оброблятися нові дані, в даному випадку інтервал, рівний 1 секунді.

```
JavaStreamingContext jssc =
    new JavaStreamingContext(Durations.seconds(1));
```

Після цього потрібно відфільтрувати ті повідомлення, які здаються нам доречними – наприклад, зі згадуванням завчасно визначених слів. Це можна легко зробити за допомогою Spark Streaming.

```
TwitterUtils.createStream(ssc, auth)
    .filter( _.getText.contains("someword1") ||
             _.getText.contains("someword2") ||
             _.getText.contains("someword3"));
```

Потім потрібно буде провести певний семантичний аналіз повідомлень, щоб визначити, чи актуальна отримана інформація. Для цього можна використати метод опорних векторів (SVM). Отриманий в результаті зразок коду з MLlib має наступний вигляд.

```
val data = MLUtils
.loadLibSVMFile(sc, "sample_correct_messages.txt")
```

Далі потрібно розділити дані на тренувальні та тестові.

```
val splits = data.randomSplit(Array(0.6, 0.4), seed =
11L);

val training = splits(0).cache();

val test = splits(1);
```

Запустити тренувальний алгоритм, щоб побудувати модель.

```
val numIterations = 100;

val model = SVMWithSGD
     .train(training, numIterations);
```

Очистити порогове значення, задане за замовчуванням.

```
model.clearThreshold();
```

Обчислити показники по тестовій множині.

```
val scoreAndLabels = test.map {

     point =>
          val score = model.predict(point.features);

};
```

Отримати параметри обчислень.

```
val metrics =
     new BinaryClassificationMetrics(scoreAndLabels);
val auROC = metrics.areaUnderROC();
```

В результаті отриманих даних, якщо відсоток вірних прогнозів даної моделі задовольняє умови, після цього можна переходити до наступного етапу, а саме відповідно реагувати на цільову подію. Для цього потрібно скористатися SparkSQL і запросити наявну таблицю Hive, де зберігаються дані про користувачів, що бажають отримувати повідомлення про

визначену цільову подію, вилучити їх електронні адреси і розіслати їм персоналізовані сповіщення.

```
val sqlContext =
    new org.apache.spark.sql.hive.HiveContext(sc);
sqlContext
    .sql("FROM earthquake_warning_users SELECT
            firstName, lastName, city, email")
                .collect().foreach(sendEmail);
```

**Висновки та пропозиції.** В даній статті було досліджено особливості побудови системи аналізу повідомлень соціальної мережі в реальному часі з метою виявлення цільових подій за допомогою Apache Spark. Проаналізовано можливості застосування Spark Streaming для обробки потокових даних. Визначено засоби класифікації отриманих повідомлень за допомогою Spark MLlib.

В результаті дослідження було проаналізовано взаємодію в реальному часі подій, які цитуються в соціальній мережі. Запропоновано алгоритм для моніторингу повідомлень і виявлення цільової події. Для виявлення цільової події розроблено класифікатор повідомлень за допомогою метода опорних векторів на основі таких функцій, як ключові слова в повідомленні, кількість слів та їх контекст. Наступним етапом розробки буде створення ймовірнісної просторово-часової моделі для цільової події, яка може знайти центр і траєкторію розташування події.

**Література:**

1. Jean J. Labrosse. DSP in Embedded Systems / Jean J. Labrosse. – Newnes, 2007. – С. 792.
2. Cortes C. Support-vector networks. Machine Learning / Cortes C., Vapnik, V. – Kluwer Academic Publishers, 1995. – С. 297
3. G. Grosseck. Analysis indicators for communities on microblogging platforms / G. Grosseck, C. Holotescu– eLSE Conference, 2009. – С. 314.
4. James A. Scott. Getting Started with Apache Spark / James A. Scott. – USA: MapR technologies Inc, 2015. – С. 88.
5. H. Karau. Learning Spark: Lightning-Fast Big Data Analysis / H. Karau, A. Konwinski, P.Wendell, M.Zaharia. – USA: O'Reilly Media Inc, 2015. – С. 257.
6. Jacek Laskowski. Mastering Apache Spark. – Режим доступа: https://www.gitbook.com/book/jaceklaskowski/mastering-apache-spark/details. – Дата доступа: 25.05.2017
7. Офіційна документація Apache Spark. – Режим доступа: https://spark.apache.org/. – Дата доступа: 25.05.2017