

1. Чутливість до похибок вхідних даних. Невелика похибка в матриці вхідних даних може призвести до великої похибки в результаті.

2. Вимоги до пам'яті. Метод потребує зберігання оберненої матриці, що може стати проблемою для великих систем лінійних рівнянь.

3. Час виконання. Метод може бути повільним для великих систем лінійних рівнянь, особливо якщо потрібно виконувати обернення матриці.

Отже, у процесі розв'язання складних і великих систем лінійних рівнянь актуальним завжди залишається питання про ефективність та швидкість розв'язання. У цьому контексті використання різних методів розв'язання систем лінійних рівнянь є ключовим фактором.

Список літератури:

1. Метод Гауса. Вища математика в прикладах і задачах. Кленко В.Ю., Голець В.Л. К. : Центр учбової літератури, 2009. 594 с. URL: <https://cutt.ly/n6Ra2uz>

2. Метод Гауса-Зейделя: пояснення, додатки, приклади. URL: <https://cutt.ly/k6Rat83>
Обернена матриця. Системи лінійних алгебраїчних рівнянь. Основні означення. URL: <https://cutt.ly/y6RaSLr>

УДК 004.453

Лічереп А.О., студент¹

Мальований Д.В., студент²

Богач І.В., к.т.н., доцент^{1,2}

РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ДЛЯ ПРИЙНЯТТЯ ПОЗИТИВНИХ РІШЕНЬ У СИСТЕМІ БЕЗ СТАНУ

¹ *Донецький національний університет імені Василя Стуса, м. Вінниця*

² *Вінницький національний технічний університет, м.Вінниця*

Вступ

21 століття — це час різкого економічного та технологічного розвитку, де на фоні глобалізації зростаються потреби та можливості людства. Величезна кількість компанії потребують систему, яка може приймати рішення без урахування стану. Зазвичай так програмні продукти регулюються законодавством та для початку діяльності повинні пройти сертифікацію, що є однією з головних проблем. З точки зору математики потрібно забезпечити незалежність та унікальність кожного окремого рішення, також однією з задач є забезпечення збіжності прийнятих рішень до закону про великі числа.

Актуальність

Дана сфера діяльності є доволі актуальною і має великий попит, але через складність реалізацію, кількість компаній, які можуть надавати подібний функціонал є доволі невеликий. Зазвичай це великі компанії, які не надають систему прийняття рішень, а частіше інтегрують його у продукти. Такі механізми

обробки інформації є попередньо визначеними, і їх потрібно інтегрувати з наданим API, також при потребі впровадити зміни це стане майже неможливим або дорого коштовним.

Це створює передумови до створення системи прийняття рішень, із можливістю конфігурації, із генератором випадкових чисел, який потрібно сертифікувати, що дає можливість придбати чистий сервіс прийняття рішень та адаптувати його до своїх конкретних вимог без необхідності робити додаткові витрати та запити до системи підтримки прийняття рішень, для розширення бажаної функціональності.

Аналіз останніх досліджень

Під час дослідження було виявлено, що розробка повинна містити щонайменше два модулі: Генератор випадкових чисел і систему прийняття рішень. Генератор використовує криптографічно стійкий алгоритм, а систему прийняття рішень забезпечує збіжність математичного ряду з використанням сигмоїдальної функції, яка є показником випадковості і описує діапазон коливань часових рядів у фінансовій сфері.

Мета

Основною метою є проектування та розробка програмного модуля здатного приймати рішення без урахування попередніх рішень. Система повинна працювати під високим навантаженням та забезпечувати задовільний ступінь ентропії з дотриманням закону про великі числа.

Постановка задачі

Необхідно розробити сервіс, який буде забезпечувати випадкові числа та приймати рішення на основі згенерованих чисел. Для досягнення поставленої мети необхідно провести аналіз існуючих аналогів, переглянути необхідні інструменти та впровадити генератор випадкових чисел, що відповідає обмеженням. З математичної точки зору потрібно вирішити проблему створення системи рішення на основі ентропії, інтегрувати сервіс випадкових чисел та пройти сертифікацію. Остаточним результатом має стати опис розробленої системи.

Результат досліджень

У загальному випадку волатильність розподілу X можна виразити таким чином:

$$x_{mean} = \sum_1^n \frac{x_i}{n} [1]$$

$$\sigma = \sqrt{\left(\frac{1}{n-1} * \sum_1^n (x_i - X_{mean})^2\right)}$$

де σ – волатильність,
 n – кількість елементів у розподілі,
 x_i – i -й член розподілу,
 x_{mean} – середнє значення розподілу.

Для вирішення задачі вибірки учасників, які задовольняють певні умови, можна використовувати формули, які залежать від особливостей конкретної ситуації. Прийняття рішень на основі цих формул забезпечує збіжність рядів при збільшенні кількості рішень, що дозволяє отримувати чітку тенденцію. Цей підхід гарантує випадковість вибірки і забезпечує збіг з загальним розподілом згідно з правилом великих чисел у системах без стану.

Хоча існує подібний підхід для інвестиційної сфери, він має свої обмеження, оскільки працює лише на основі історичних даних і не може гарантувати відповідність умовам достатності кожного елемента для належності до загального розподілу.

Висновки

Проведено аналіз існуючих аналогів та встановлення, що на ринку відсутні прямі конкуренти, оскільки більшість сервісів пропонують системи прийняття рішень та генератори випадкових чисел, як частину більших послуг. Це свідчить про високий попит на чистий сервіс, який надає тільки випадкові числа та систему прийняття рішень.

Після дослідження інструментів для реалізації, була обрана мова програмування Java [2] та фреймворк Spring Boot [2] та отримані всіх обмежень та правила, встановлені законодавством та вимогами бізнесу. Як доказ того, що сервіс відповідає всім правилам і обмеженням, генератор випадкових був сертифікований британським акредитованим органом, що проводить аудит і сертифікацію.

Після інтеграції генератора випадкових та системи прийняття рішень завдання цієї роботи можна вважати виконаним.

Список літератури

1. Математичне очікування [Електронний ресурс] — Режим доступу: <https://studfile.net/preview/8831702/page:2/>.
2. Документація по мові програмування Java [Електронний ресурс] — Режим доступу: <https://docs.oracle.com/en/java/>.
3. Документація на фреймворк spring [Електронний ресурс] — Режим доступу: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>.

