МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра програмних систем і технологій

Дисципліна
«МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ»

Лабораторна робота № 2

«Імітаційні моделі»

Виконав:	Гоша Давід	**Перевірив**:	
Група	ІПЗ-33	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		

2022

Тема (завдання) для дослідження – Доказ парадоксу Пуассона для біткоїну.

Аналіз предметної області – Біткойн - це електронна валюта та платіжна система, яка дозволяє здійснювати грошові операції без центрального органу. Вперше біткойн було запропоновано в white paper у 2008 році під псевдонімом Сатоші Накамото, і представлена як функціонуюча система програмного забезпечення з відкритим кодом у січні 2009 р. З тих пір отримав популярність і тепер має ринкову капіталізацію понад 368 мільярдів доларів США.

Заявлені переваги біткойна порівняно з оплатою Інтернет з традиційними валютами включають незмінність трансакції, відносну анонімність, свободу від впливу центральним органом влади, швидші та дешевші міжнародні перекази, порівняно з іншими. Технологія, яка також використовується в Bitcoin має потенціал для застосування в більш широкому діапазоні застосувань включаючи керування ідентифікацією, розподілену сертифікацію та розумні контракти.

Біткойн використовує однорангову мережу вузлів, комп'ютерів які перевіряють, поширюють і зберігають інформацію про біткойн операції. Біткойн веде глобальну книгу всіх біткойнів транзакцій, які коли-небудь проводилися – блокчейн біткойн – розподілений через мережу вузлів біткойн. Проблеми ведення та оновлення блокчейну без центрального контрагенту або органу влади, але все ще дозволяє будь-якому власнику біткойнів витрачати їх, як вони хочуть, вирішується серією криптографічних методів. Протокол, що вирішує алгоритм вибору лідера або так званий консенсус – Proof of Work, за допомогою якого ті, хто бажає зробити внесок для досягнення консенсусу щодо того, які транзакції включатимуться у blockchain, має виконати певну обчислювальну «роботу». Ця робота називається майнінгом біткойнів.

Користувачі Bitcoin здійснюють транзакції за допомогою криптографічних підписів повідомлень, які визначають, хто має бути дебетований, хто підлягає зарахуванню, і куди внести решту (за наявності). Потім ці повідомлення поширюються через біткойн мережу, перед додаванням до блокчейну. Кожен вузол в мережа зберігає список дійсних не підтверджених транзакцій, викликаних пулом транзакцій і передає його сусідам у мережі. Перш ніж транзакцію можна буде включити в блокчейн, новий блок, що містить цю транзакцію, повинен бути видобутий. Блок це список транзакцій разом із метаданими, які включають поточний час і посилання на останній попередній блок у блокчейні (звідси назва блокчейн). Блок також містить поле, яке називається nonce. Одноразовий номер не містить жодної інформації, тому його можна вільно змінювати, намагаючись знайти дійсний блок, який задовольняє вимогам блокчейну. Знайти такий блок і опублікувати його в мережі біткоїну називається майнінгом блоку.

Майнінг — це змагання між усіма майнерами біткойнів у пошуку дійсного

блоку для додавання до блокчейну. Щоб мати хороші шанси виграти цю гонку, потрібні значні обчислювальні ресурси, і винагородою за це є емісія валюти (наразі 6.25)

Вимоги до того, щоб блок був дійсним, базуються на хеш-функції Bitcoin. Загалом, хеш-функція f відображає рядок s (або, еквівалентно, ціле число, якщо бінарне представлення рядка інтерпретується як ціле число за основою 2) в інший рядок (ціле число) f(s), хеш s. Ключовою властивістю хеш-функцій, які використовуються в біткойнах, є те, що для будь-якої області значень f обчислювально неможливо знайти будь-який елемент f^{-1} (y). Крім того, хеш-функції розроблені таким чином, що будь-яка зміна у вхідному рядку s призводить до зовсім іншого результату f(s). Тобто $f(s_1)$ не дає інформації про $f(s_2)$, якщо $s_1 \neq s_2$, тому обчислення хешів для великої кількості вхідних рядків потрібно обчислювати кожен хеш окремо. Ця властивість дозволяє хешам діяти як короткі, захищені від підробки підсумки великих файлів даних. Обчислення хешів великої кількості аргументів s називається Proof of Work

Мета практикуму – Проаналізувати процесс підтвердження блоків у блокчйейні біткоїну. Довести що розподіл знайдених хешів є пуассонівським.

Гіпотеза – Більшість користувачів очікують підтвердження блоку більше ніж 10 хвилин, не дивлячись на те, що в середньому воно становить 10 хвилин.

Технічне завдання на розроблення моделі

лок, моделюється як процес Пуассона зі швидкістю λ. Тут середній час між блоками, $\frac{1}{\lambda}$, встановлюється на основі цільової складності в операції майнінгу PoW; для Bitcoin $\frac{1}{\lambda}$ = 10 хвилин. Пуассонівський процес — це процес, у якому нові події (або надходження) відбуваються через випадкові проміжки часу після експоненціального розподілу. Крім того, інтервали між будь-якими двома подіями не залежать один від одного та статистично ідентичні. Нагадаємо, що експоненціальна випадкова величина X з параметром λ має розподіл

$$\mathbb{P}\,(x \geq t) = exp(-\lambda t) \forall t \geq 0$$

Також нагадаємо, що пуассонівська випадкова величина Y з параметром λ має розподіл

$$\mathbb{P}\,(Y = k) = exp(-\lambda)\frac{\lambda^k}{k!} \forall k \geq 0$$

У пуассонівському процесі кількість подій на інтервалі довжиною T є пуассонівською випадковою величиною з параметром λT. Крім того, кількість подій у непересічних проміжках часу не залежить. Якщо ми розглядаємо малі інтервали ($\lambda T << 1$), то в інтервалі з ймовірністю λT є одна подія і жодної іншої. Таким чином, процес Пуассона можна імітувати шляхом «гри в орлянку» де послідовність (незалежних) підкидань монети має дуже малу вірогідність випаду «орла». Тепер ми розуміємо, чому процес майнингу має таку властивість.

Уявіть, що кількість майнерів у системі та загальна обчислювальна

потужність у їхньому розпорядженні є постійною протягом певного періоду часу. Припустімо, що кожен з комп'ютерів майнерів безперервно перебирає хеші, і це єдине обчислення, яке вони виконують. Тоді можна сказати, що загальна кількість хешів, які обчислюються (еквівалентно, загальна кількість оказій, які перебираються) за одиницю часу є приблизно постійною. Скажімо, щосекунди обчислюється мільярд хешів.

Крім того, припустимо, що рівень складності хеш-головоломки для кандидатного блоку дуже висока. Наприклад, скажимо, що перші тридцять п'ять бітів хешу мають бути нульовими, щоб блок був дійсним. Імовірність того, що певний nonce відповідатиме цьому критерію, становить $2^{-35} \approx 3 \times 10^{-11}$. Таким чином, ймовірність того, що хеш-головоломка буде розв'язана будь-яким майнером за певну секунду часу, становить 0,03 , невелике число (ми припустили, що кожну секунду обчислюється мільярд хешів). Те, чи знайдено хеш підтвердження роботи в певну секунду, не впливає на те, чи буде він знайдений у наступну секунду. Причина цього в тому, що хеш-функція, по суті, є випадковим оракулом; хеш-значення для різних вхідних даних не залежать одне від одного. Таким чином, процес видобутку добре моделюється як процес Пуассона. При моделюванні процесу видобутку як пуассонівського процесу ми зосереджуємося лише на моменті створення нових дійсних блоків.

Модель процесу Пуассона актуальна незалежно від кількості майнерів, їхньої індивідуальної обчислювальної потужності, того, чи працюють різні майнери над одним або різними блоками, і коли різні користувачі отримують щойно видобуті блоки. Параметр λ процесу майнінгу, який називається швидкістю майнінгу, дорівнює середній кількості блоків, видобутих за одиницю часу. У біткоїнах λ дорівнює 1/(600 с), тобто один блок кожні 600 секунд (десять хвилин).

З варіаціями загальної обчислювальної потужності припущення про фіксовану швидкість майнінгу спростовуються. Насправді загальна обчислювальна потужність не змінюється раптово. Таким чином, протягом невеликого періоду часу швидкість є приблизно постійною. Регулювання параметра складності через регулярні проміжки часу допомагає підтримувати швидкість видобутку на одному рівні.

Математичний опис моделі

Чи є надходження блоків Пуассонівським процесом. Спробуємо довести це. Відразу відмітимо що незалежно від того, чи моделюється глобальний хешрейт H(t). емпірично або параметрично, для кожного входження блоку моделі в цьому моделюванні хешрейт визначається до вибірка випадкового процесу $X_i(t)$.

- Для моделей із детермінованим налаштуванням складності, складність коригується в детермінований час моменту y_n, які не відповідають випадковим моменти надходжень блоків та швидкість надходження блоків $\lambda(t)$ не залежить від надходжень блоку в попередньому сегменти. Якщо

немає затримки, то на кожному інтервалі модель є неоднорідним процесом Пуассона.

- Для моделей із випадковим налаштуванням складності, складність регулюється у випадкові моменти часу, після кожного 2016 сегменту блокчену використовуючи рівняння. Якщо немає затримки розповсюдження, тоді кожен сегмент процесу є неоднорідним пуассонівським зі швидкістю, заданою як $\lambda(t) = H(t)/D_i$. Оскільки швидкість надходження блоку $\lambda(t)$ залежить від першого та останнього надходження в попередній сегмент блокчейну, процес не Пуассонівський протягом послідовного періоду часу сегменту.
- Якщо присутня затримка поширення, то прибуття блоку процес навіть не є неоднорідним процесом Пуассона на одному сегменті. У наступному розділі ми порівняємо їх моделювання до даних про позначку часу з блокчейну біткоїна.

Тому ми знехтуємо динамічною складністю і моделюватиме тільки сегмент з 2016 блоків, приблизно 1 місяць реального часу, або 20160 хвилин. Також припустимо, що розповсюдження блоків в мережі є моментальним та не викликає жодних затримок.

Точковий процес N є процесом Пуассона на $\mathbb{R}$, якщо він має наступні дві властивості.

1) Випадкова кількість точок N([a, b)) точкового процесу N, розташованих в обмеженому інтервалі [a, b) ⊂ R, є пуассонівською випадковою величиною із середнім Λ([a, b)), де Λ є невід'ємною мірою Радона.
2) Кількість точок точкового процесу N, розташованих на k інтервалах [a1, b1), . . . , [ak, bk) утворюють k незалежних пуассонівських випадкових величин із середніми Λ([a1, b1)), . . . ,Λ([ak, bk)).

Відтепер будемо записувати N([a, b)) як N(a, b) і Λ([a, b)) = Λ [a, b) для зручності. Перша властивість передбачає що

$$\mathbb{P}(N(a,b) = n) = \frac{\Lambda(a,b)^n e^{-\Lambda(a,b)}}{n!}$$

і $E[N(qb)] = \Lambda(a,b)$, а друга властивість – це Основна причина придатності процесу точки Пуассона і зазвичай це основа статистичних тестів, які вимірюють адекватність моделей Пуассона. Розподіл Пуассона N(a, b) означає, що його дисперсія Var[N(a, b)] = Λ(a, b), факт який також використовується як статистичний тест. Міра Λ відома як міра інтенсивності або середнє значення міри процесу точки Пуассона. Припустимо, що а існує така функція λ(t), що

$$\Lambda(a,b) = \int_a^b \lambda(t)\, dt$$

Тоді λ(t) визначена як функція швидкості. Якщо λ(t) є сталою λ > 0, то процес називається однорідним точковим процесом Пуассона. Інакше процес називають

неоднорідним або **неоднорідним точковим процесом Пуассона**. Якщо обмежити нашу увагу інтервалом невід'ємних чисел [0, ∞), міра інтенсивності задається формулою

$$\Lambda(t) := \Lambda([0,t]) = \int_0^t \lambda(t)\, dt$$

Для пуассонівського процесу N з мірою інтенсивності Λ ймовірність існування n точок в інтервалі [a, b) дорівнює

$$\mathbb{P}(N(a,b) = n) = \frac{[\Lambda(b) - \Lambda(a)]^n e^{-[\Lambda(b)-\Lambda(a)]}}{n!}$$

Час надходження та час між надходженнями: розглянемо точковий процес $\{X_{(i)}\}_{i\geq 1}$, визначений на невід'ємних дійсних числах із майже напевно кінцевою кількістю точок у будь-якому обмеженому інтервалі. Тоді ми можемо інтерпретувати точки процесу як часи добування нових блоків та розмістити їх у порядку зростання, $X_1 \leq X_2 \leq \ldots$ Тоді відстані між сусідніми точками дорівнюють $T_i := X_i - X_{i-1}$ для i = 2, 3, . . . і $T_1 = X_1$. Випадкові величини T_i відомі як час очікування або час між надходженнями. Для однорідного процесу Пуассона зі швидкістю λ відповідні часи між надходженнями є незалежними та однаково розподіленими експоненціальними випадковими величинами із середнім значенням 1/λ

$$\mathbb{P}(T_k < t) = 1 - e^{-\lambda t}$$

Де властивість експоненціального розподілу без пам'яті було використано. Це не стосується неоднорідного точкового процесу Пуассона з інтенсивністю λ(t), де перший час між надходженнями $T_1 = X_1$ має розподіл

$$\mathbb{P}(T_1 \leq t_1) = 1 - e^{-\int_0^{t_1} \lambda(s)\, dS}$$

За першого часу очікування $T_1 = t_1$ умовний розподіл другого часу очікування T_2 є

$$\mathbb{P}(T_2 \leq t_2 | T_1 \leq t_1) = 1 - e = \int_{t_1}^t \lambda(s)\, ds$$

і так далі для k ≥ 2

$$\mathbb{P}(T_k \leq t_k | T_{k-1} \leq t_{k-1}) = 1 - e^{\int_{t_{k-1}}^{t_k} \lambda(s) ds}$$

Можна показати, що k-й час надходження X_k має розподіл

$$\mathbb{P}(x_k \leq t) = e^{-\Lambda(t)} \sum_{n=k}^{\infty} \frac{\Lambda(t)^k}{k!}$$

З щільністю

$$f x_k(t) = \frac{\lambda(x) \wedge (t)^{k-1}}{(k-1)!} e^{-\Lambda(t)}$$

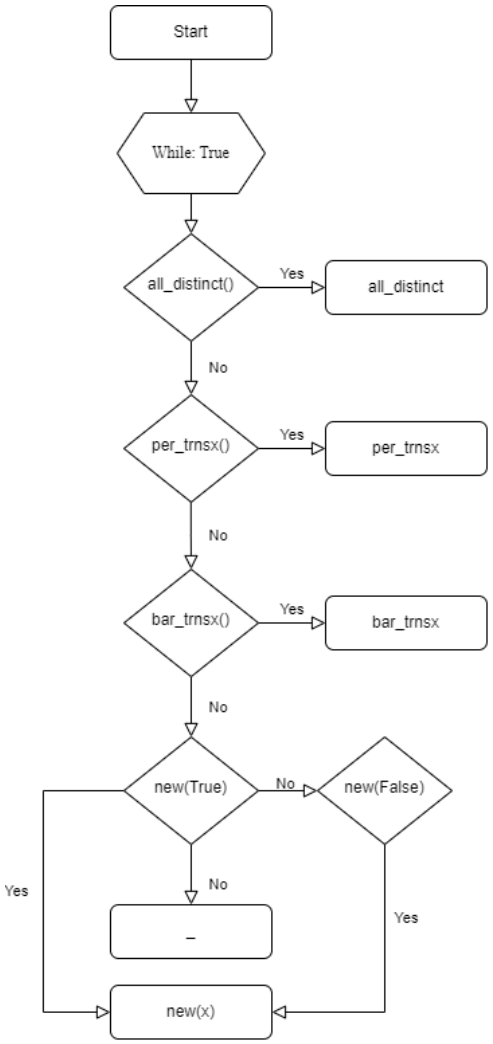
Умова на n точок $\{U_i\}_{i=0}^{n}$ пуассонівського процесу, що існує в деякому обмеженому інтервалі [0, t]. Ми називаємо ці точки умовним часом надходження блоку. Якщо процес Пуассона є однорідним, то умовні часи надходження рівномірно і незалежно розподілені, утворюючи n рівномірних випадкових величин на [0, t]. Ця різниця між часом очікування T_i та умовним часом надходження U_i відіграє роль у тесті Пуассона.

Для неоднорідного пуассонівського процесу кожна точка U_i незалежно розподілена на інтервалі [0, t] із розподілом

$$\mathbb{P}(U_i \leq u) = \frac{\Lambda(u)}{\Lambda(t)}, u \in [0, t]$$

Якщо розподіл кожного U_i відомий і оборотний, то кожен U_i може бути перетворений в рівномірну випадкову величину на [0, 1], що призводить до n незалежних рівномірних випадкових величин. Іншими словами, Λ(t) перетворює процес Пуассона на однорідний процес Пуассона з густиною один на відрізку дійсних чисел. Отже, статистичні методи для неоднорідних процесів Пуассона часто передбачають перетворення даних перед виконанням аналізу.

Алгоритм роботи моделі та її ключових функцій



Комп’ютерна програма мовою Python.

```
def per_trnsx():
    # Draw Plot
    plt.figure(figsize=(13,10), dpi= 80)
    sns.distplot(df.loc[df['transaction_count'] < 500, "time_per_block"], color="dodgerblue", label="- 500",
hist_kws={'alpha':.7}, kde_kws={'linewidth':3})
    sns.distplot(df.loc[(df['transaction_count'] < 1000) & (df['transaction_count'] > 500), "time_per_block"],
color="orange", label="500 - 1000", hist_kws={'alpha':.7}, kde_kws={'linewidth':3})
    sns.distplot(df.loc[(df['transaction_count'] < 2000) & (df['transaction_count'] > 1000),
"time_per_block"], color="g", label="1000 - 2000", hist_kws={'alpha':.7}, kde_kws={'linewidth':3})
    sns.distplot(df.loc[df['transaction_count'] > 2000, "time_per_block"], color="deeppink", label="2000 +",
hist_kws={'alpha':.7}, kde_kws={'linewidth':3})
    plt.ylim(0, 0.2)

    # Decoration
    plt.title('Density Plot of City Transaction by Count', fontsize=22)
    plt.legend()
    plt.show()
```

Приклад функції , що малює діагращу щільності в залежності від кількості транзакцій.

```
def new(x):
    df_raw = pd.read_excel("C:/Users/admin/Desktop/Labs/3 course/ММП/ЛАБ2/12.xlsx")
    df = df_raw[['transaction_count', 'median_time','time_per_block']].groupby('median_time').apply(lambda x:
x.mean())
    if x:
        df.sort_values('transaction_count', inplace=True)
    df.reset_index(inplace=True)

    # Draw plot
    import matplotlib.patches as patches

    fig, ax = plt.subplots(figsize=(16,10), facecolor='white', dpi= 80)
    ax.vlines(x=df.index, ymin=0, ymax=df.transaction_count, color='firebrick', alpha=0.7, linewidth=20)

    # Annotate Text
    for i, trx in enumerate(df.transaction_count):
        ax.text(i, trx+0.5, round(trx, 1), horizontalalignment='center')


    # Title, Label, Ticks and Ylim
    ax.set_title('Bar Chart for Transaction Slow Blocks', fontdict={'size':22})
    ax.set(ylabel='Transaction per block', ylim=(0, 4000))
    plt.xticks(df.index, df.median_time, rotation=60, horizontalalignment='right', fontsize=12)

    # Add patches to color the X axis labels
    p1 = patches.Rectangle((.57, -0.005), width=.33, height=.13, alpha=.1, facecolor='green',
transform=fig.transFigure)
```

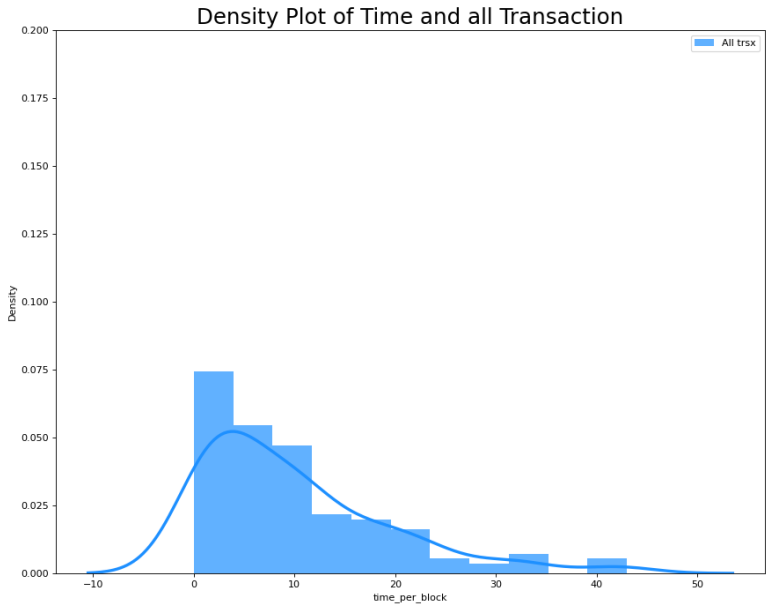
```
    p2 = patches.Rectangle((.124, -0.005), width=.446, height=.13, alpha=.1, facecolor='red',
transform=fig.transFigure)
    fig.add_artist(p1)
    fig.add_artist(p2)

    # y = df['transaction_count']
    if x:
        ax.plot(df.sort_values(by=['transaction_count'])['time_per_block'] * 50)
    else:
        ax.plot(df['time_per_block'] * 50)

    plt.show()
```

Наведені дві функції, що читають файл з вихідними даними взяті за 18.10.22 - 19.10.22 з офіційного блокчейну біткойну, всі блоки, знайдені за цей проміжок часу.

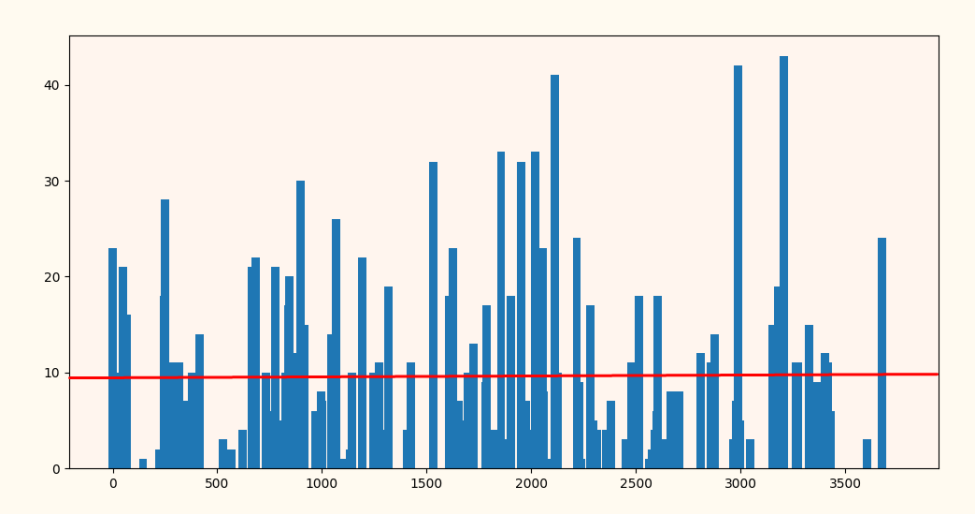
Аналіз результатів



Підсумовуючи, швидкість майнингу біткойна відповідає розподілу Пуассона, тобто більшість блоків знайдено протягом 10-хвилинного інтервалу. Але оскільки це випадковий процес, завжди є деякі повільні блоки, на пошук яких майнерам потрібно більше або менше часу. Тим не менш, середній час підтвердження транзакції має становити близько 10 хвилин, як показано вище.

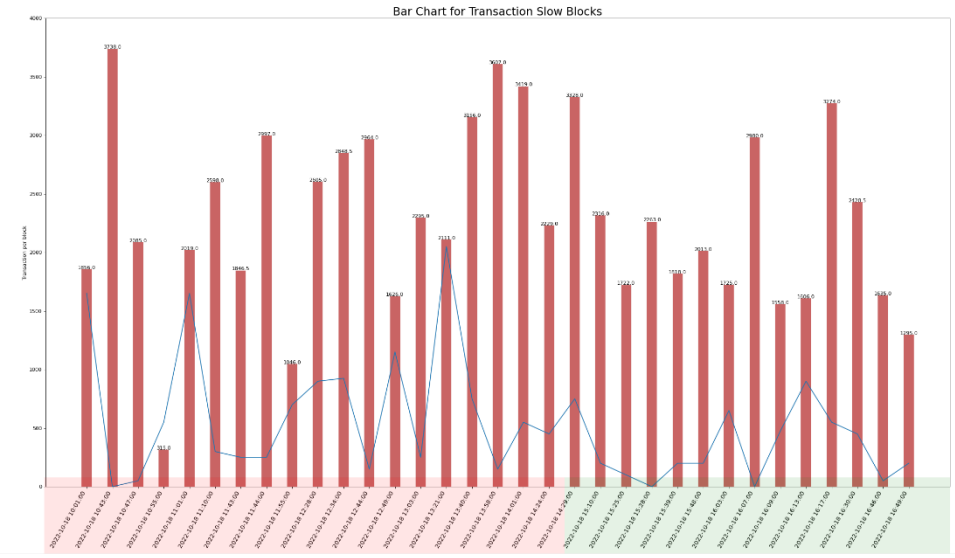
Парадокс — це на перший погляд абсурдне твердження, яке при дослідженні виявляється обґрунтованим, але нелогічним. Отже, наприклад, припустимо, що більшість людей очікують довший час підтвердження транзакції, незважаючи на те, що середній показник становить 10 хвилин.

Було використано невелику вибірку з 140 блоків BTC від 759149 до 759289, щоб дослідити цю гіпотезу. Спочатку ми запускаємо перевірку наших даних.



Гістограма нижче показує, що час підтвердження транзакції насправді нагадує розподіл Пуассона. Більшість транзакцій, 78%, займає від 5 до 20 хвилин. А середній час знаходження блоку становить 9,9 хвилин.

Нижче наведено графік хвилин між блоками, зображений помаранчевою лінією. Сума транзакцій на блок показана синіми стовпцями. Ми виявили, що середня кількість транзакцій на блок становить 1805.

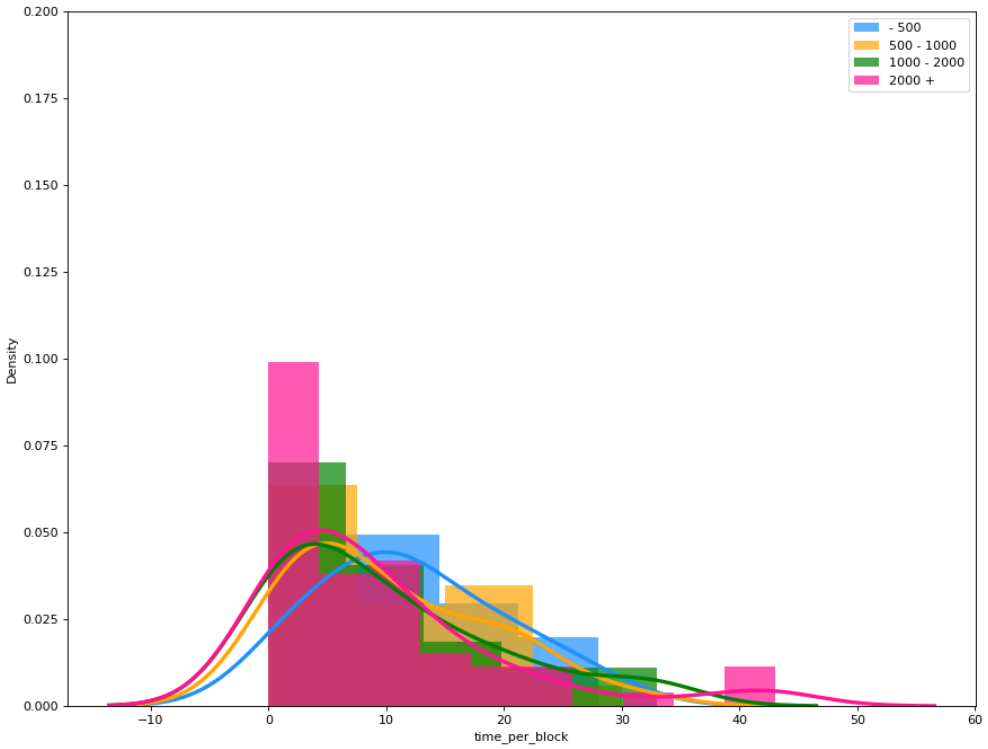


Синя лінія позначує час очікування підтвердження блоку. Отже можна побачити кореляцію, між очікуванням блоку та великою кількістю транзакцій у наступному блоці. Тобто, якщо підтвердження прийшло швидко, як о 10.47, то наступній блок буде порожнім і навіть якщо 2000 транзакцій підтвердились упродовж 5ти хвилин, блоки з ними були на-пів порожніми, що свідчить про те, що менша кількість користувачів отримала швидке підтвердження, ніж користувачі о 13.21 де 40 хвилин очікувало майже 11 тис транзакцій.

Таким чином, не обов'язково, що всі швидкі блоки малі за розміром. Наприклад, об 10:55 один блок займає 5 хвилин майнінгу, але підтверджує лише 200 транзакцій. У той час як об 15:28 пошук блоку займає 0 хвилин і містить 2982 транзакції. Таким чином, розмір блоку залежить від кількості транзакцій, які очікують підтвердження в mempool (перевантаження), а не від швидкості надходження блоку.

З рештою, це дослідження даних не доводить існування парадоксу Пуассона біткойна. Однак доказ насправді прихований у початковій гістограмі, тобто в довгому хвості.

Дійсно, більшість людей у нашій вибірці чекають на підтвердження довше 10 хвилин, хоча середнє значення становить 9,9 хвилин. Це тому, що розподіл Пуассона має довгий хвіст вправо. Тобто, існує більше можливостей для виявлення блоку між 10-40 хвилинами, оскільки часовий інтервал у чотири рази більший, ніж інтервал між 0-10 хвилинами. Дивіться нижче.



Час підтвердження блоку	Відсоток вибірки
0 – 10 хвилин	40 %
10 – 40 хвилин	60 %

Підсумовуючи, у 2/5 нашої вибірки трансакція підтверджується менш ніж за 10 хвилин, тоді як у 3/5 трансакція підтверджується більше ніж за 10 хвилин. Це парадокс Пуассона біткойна.

Є ще кілька причин, чому транзакції BTC можуть проходити повільніше або швидше, ніж 10 хвилин. Хоча це не стосується парадоксу Пуассона біткойна.

По-перше, майнери збирають і хешують транзакції з мемпулу в заголовок блоку, перш ніж шукати наступне значення nonce. Це може створити відставання. Наприклад, нові транзакції, не підібрані майнерами, залишаються бездіяльними в мемпулі, доки не буде видобуто попередній блок, а також час, необхідний для виявлення нового блоку(ів). Транзакції також можуть застрягти,

якщо комісія занадто низька.

Однак, якщо транзакція зависає, її можна збільшити, змінивши комісію за допомогою Child-Pays-for-Parent (CPFP) або Replace-by-Fee (RBF). Зміна вартості комісії за транзакцію підвищує її пріоритет, тому майнери з більшою ймовірністю включать її у свій наступний блок. Крім того, деякі майнінгові пули можуть додавати транзакції до своїх власних блоків, що прискорює час транзакцій.

Висновок

У лабораторній роботі ми представили набір моделей для точкового процесу епох майнінгу блоків у блокчейні біткойн і протестували його за допомогою моделювання та даних, доступних із самого блокчейну. Основними труднощами в цьому є:

1) Невідомий глобальний хеш-рейт, який керує швидкістю виявлення блоків;
2) Історично відоме, але випадкове значення складності майнінгу.

Крім того, дані про час надходження блоків є вичерпними, але не повністю надійними. Ми постулюємо модель точкового процесу, в якій процес **надходження блоку поводиться як неоднорідний процес Пуассона** в періоди між змінами складності зі швидкістю, пропорційною відношенню швидкості хешування до складності, але яка залежить від себе, коли враховуються зміни складності. Для заданої глобальної швидкості хешування час, коли змінюється складність (і, отже, змінюється швидкість надходження блоку), визначається шляхом вибірки процесу.

Тим не менш, глобальна швидкість хешування стабільно швидко зростає, а механізм зворотного зв’язку щодо труднощів має достатню затримку, тому швидкість надходження блоків була приблизно на 11,5% більшою за базову швидкість 6 блоків на годину. Це означає, що в цілому пропускна здатність транзакцій і загальний дохід майнера від бонусів вищі, ніж у базовому варіанті.

Крім того, час, коли винагорода за майнінг блоків зменшується вдвічі, і час, коли всі біткойни будуть створені, настане раніше, ніж це було б, якби блоки майнилися зі швидкістю шість на годину. Окрім надання моделі для процесу надходження блоків, ми вивели зв’язок між частотою надходження блоків і швидкістю експоненціального збільшення швидкості хешування. Ми також надали практичне наближення, яке демонструє граничну поведінку незалежно від початкових умов і збурень процесу надходження блоку, і перевірили це наближення за допомогою моделювання та вимірювань з блокчейну.

Також **було підтверджено гіпотезу стосовно парадоксу Пуассона** в очікуванні підтвердження транзакції. Через те що розподіл має довгий хвіст вправо більшість користувачів очікують підтвердження блоку довше за середній час, у той час коли менша кількість користувачів має пріоритет у швидкості.