

Владислав Сергійович Зетченко¹, Артем Олександрович Задорожній²

¹аспірант кафедри інформаційних технологій та програмної інженерії

Національний університет «Чернігівська політехніка» (Чернігів, Україна)

E-mail: zetchenkovlads@gmail.com. **ORCID:** <https://orcid.org/0009-0002-9925-8628>. **ResearcherID:** ITW-2651-2023

²кандидат технічних наук, доцент кафедри інформаційних технологій та програмної інженерії

Національний університет «Чернігівська політехніка» (Чернігів, Україна)

E-mail: zaotroy@gmail.com. **ORCID:** <https://orcid.org/0000-0002-3424-7293>. **ResearcherID:** F-6358-2016

МОДЕЛЮВАННЯ ТА АНАЛІЗ НЕСПРАВНОСТЕЙ ТА ЗАТРИМОК ГРОМАДСЬКОГО ТРАНСПОРТУ З ВИКОРИСТАННЯМ БАЄСОВИХ МЕРЕЖ ДОВІРИ Й ГЕНЕРАТИВНОГО ШІ

Стаття присвячена актуальній проблемі управління перевантаженістю у системах громадського транспорту. Наразі фіксується зростання міського населення та збільшення навантаження на транспортну інфраструктуру, що підкреслює важливість забезпечення ефективного управління для підвищення комфорту пасажирів та зменшення операційних витрат. Стаття аналізує останні дослідження в цій області, які використовують різні підходи, такі як ймовірнісні моделі, генеративні змагальні мережі (GAN) та методи оптимізації. Представлений у статті підхід ґрунтується на інтеграції баєсових мереж довіри для моделювання складних залежностей між факторами перевантаження та генеративного ШІ для симуляції сценаріїв та оптимізації управління.

Наведені результати дослідження, які включають аналіз реальних даних за період 2015–2018 рр., демонструють ефективність запропонованого методу в прогнозуванні та запобіганні перевантаженості. Стаття має науково-практичний характер та пропонує новий інструмент для планувальників міського транспорту.

Ключові слова: баєсові мережі довіри, генеративний штучний інтелект, громадський транспорт, прогнозування завантаженості.

Рис.: 1. Табл.: 2. Бібл.: 10.

Актуальність теми дослідження. У сучасних умовах стрімкої урбанізації та зростання населення міст, системи громадського транспорту стикаються зі значною перевантаженістю, що призводить до низки негативних наслідків: зниження комфорту пасажирів, збільшення часу очікування, підвищення операційних витрат та погіршення екологічної ситуації через надмірні викиди шкідливих речовин. Традиційні підходи до управління транспортом часто виявляються неефективними через складність та невизначеність реальних умов експлуатації. Використання сучасних методів штучного інтелекту, таких як баєсові мережі довіри (БМД) та генеративний ШІ, дозволяє враховувати ці фактори та пропонувати інноваційні рішення для оптимізації роботи транспортних систем [1]. Актуальність дослідження полягає в розробці нового підходу, який поєднує переваги ймовірнісного моделювання та генеративних технологій для ефективного управління перевантаженням.

Постановка проблеми. Проблема перевантаженості громадського транспорту є складною та багатоаспектною [2]. Вона включає такі фактори, як:

- нерівномірний розподіл пасажиропотоку впродовж доби;
- обмежена місткість транспортних засобів;
- вплив зовнішніх факторів (погода, спеціальні події, дорожні затори);
- недостатня ефективність традиційних методів планування маршрутів та розкладів.

Існуючі підходи до моделювання та управління перевантаженням часто ґрунтуються на детермінованих моделях, які не враховують невизначеність та динамічність реальних умов. Це обмежує їхню ефективність та призводить до субоптимальних рішень. Виникає потреба в нових методах, які здатні інтегрувати великі обсяги даних, враховувати складні залежності між факторами та генерувати оптимальні стратегії управління.

Аналіз останніх досліджень і публікацій. Останні дослідження в галузі управління громадським транспортом зосереджуються на використанні різних методів штучного інтелекту та аналізу даних. Серед них можна виділити:

Байєсові мережі довіри: Використовуються для моделювання складних залежностей між факторами, що впливають на перевантаження [3]. Наприклад, дослідження Soukaina Bouhsissin (2022) демонструє ефективність БМД у прогнозуванні ризиків у транспортних системах [4].

Генеративний ШІ: Технології, такі як GAN (Generative Adversarial Networks) та VAE (Variational Autoencoders), застосовуються для симуляції сценаріїв та генерації синтетичних даних [5]. Існуючі дослідження показують потенціал GAN у моделюванні складних систем.

Оптимізаційні методи: Генетичні алгоритми, метаевристичні підходи та навчання з підкріпленням використовуються для оптимізації маршрутів та розкладів [6].

Проте, більшість існуючих досліджень зосереджуються на окремих аспектах проблеми, тоді як інтеграція ймовірнісного моделювання з генеративними методами залишається недостатньо дослідженою.

Виділення недосліджених частин загальної проблеми. Незважаючи на значний прогрес у галузі, залишаються недослідженими такі аспекти:

Інтеграція БМД та генеративного ШІ: Відсутність комплексних підходів, які поєднують переваги ймовірнісного моделювання з генеративними методами для симуляції та оптимізації.

Динамічне оновлення моделей: Більшість існуючих методів не враховують зміни в реальному часі, такі як раптові зміни пасажиропотоку або дорожніх умов.

Адаптація до мультимодальних систем: Існуючі рішення часто обмежуються однією транспортною системою, тоді як сучасні міста потребують інтегрованих підходів до управління різними видами транспорту.

Ці прогалини вказують на необхідність розробки нових методів, які здатні ефективно вирішувати складні задачі управління перевантаженням.

Мета статті. Метою статті є розробка нового підходу до моделювання та аналізу несправностей і затримок автобусів на основі БМД та генеративного штучного інтелекту.

Основні завдання дослідження включають такі етапи:

- Побудова БМД для моделювання факторів, що впливають на несправності та затримки автобусів.
- Використання генеративного ШІ для створення синтетичних даних, що імітують сценарії несправностей та затримок.
- Валідація запропонованого підходу на реальних даних.

Реалізація цієї мети дозволить запропонувати інноваційний інструмент для планувальників міського транспорту, який підвищить ефективність управління та покращить якість послуг для пасажирів.

Виклад основного матеріалу.

Інтеграція БМД. Ефективне передбачення та керування транспортними заторами вимагає комплексного аналізу багатьох змінних факторів, включаючи метеорологічні умови, якість дорожнього покриття та інші критичні показники. Для розв'язання таких складних завдань оптимальним рішенням є застосування байєсових мереж довіри, які здатні враховувати невизначеність та складні кореляції між різноманітними чинниками [7].

БМД можливість не тільки створювати ймовірнісні моделі на підґрунті існуючих даних, але й коригувати ці ймовірності з надходженням свіжої інформації. Така особливість дозволяє передбачати ризик утворення заторів та формувати висновки навіть за умов неповних або неточних вхідних даних, що має критичне значення в сучасних умовах. Застосування подібних моделей дає змогу визначити ймовірність появи транспортних заторів залежно від наявних обставин.

Для створення моделі прогнозування дорожніх заторів доцільно застосувати БМД, що складатиметься з декількох вузлів, кожен з яких відображає ключові чинники впливу на транспортний рух. Збільшення кількості таких чинників підвищує точність результатів, проте ускладнює архітектуру мережі.

Опис і структура набору даних. Для тренування і тестування моделі було використано датасет з відкритого джерела, що містить понад кілька десятків колонок і більше 400000 записів [10]. У статті наведено лише основні з них, щоб показати логіку обробки й застосування.

Основні колонки датасету:

- **School_Year** (varchar, шириною 9) – вказує навчальний рік, наприклад, 2022-2023. Це дозволяє аналізувати сезонні тренди та річні особливості роботи системи;

- **Route_Number** (varchar, шириною 5) – унікальний ідентифікатор маршруту, наприклад, A123 або B4567. Він є ключем для відстеження конкретних маршрутів та їх характеристик;

- **OPT_Code** (varchar, шириною 5) – унікальний код об'єкта або розташування, що отримує транспортну послугу (школа, офіс, медичний центр). Це дозволяє вносити інформацію про географічне розташування;

- **Site_Name** (varchar, шириною 30) – назва конкретної локації, наприклад, PS 123, NY High School. Це допомагає класифікувати точки обслуговування;

- **School_Age_or_PreK** (varchar) – категорія закладу за віковим рядком: PreK, Elementary, High School. Це важливий фактор, що впливає на кількість пасажирів і змінні поведінки;

- **Boro** (varchar) – район міста (Manhattan, Brooklyn, Queens), що дозволяє врахувати географічні особливості;

- **Reason** (varchar) – причина несправності або затримки, наприклад, Mechanical Issue, Traffic Jam, Accident;

- **How_Long_Delayed** (varchar) – рядок, що позначає тривалість затримки, наприклад, 15 min, 30 min. У процесі обробки ця колонка переводиться у числовий формат;

- **Occurred_On** та **Created_On** (datetime) – вказують дату і час події й внесення запису;

- Категорія – поломка або запізнення, що дає додатковий контекст;

- **Has_Contractor_Notified_Schools** і **Has_Contractor_Notified_Parents** (varchar) – ознаки повідомлення зацікавлених сторін; Ці показники формують фундамент для прогнозування дорожньої ситуації, що має вагоме значення для оптимізації маршрутів та оцінки ризиків на окремих дорожніх сегментах.

Для реалізації запропонованої моделі необхідно визначити умовні ймовірності для кожного вузла мережі.

Наприклад, таблиця умовних ймовірностей може мати наступний вигляд (табл. 1).

Таблиця 1 – Умовні ймовірності утворення заторів

School_Age_or_PreK	Reason	How_Long_Delayed
Elementary	-	0 min
Elementary	Traffic Jam	10 min
Elementary	Accident	40 min
High School	-	0 min
High School	Traffic Jam	10 min
High School	Accident	20 min

Джерело: розроблено авторами.

Застосування БМД. Залежності між цими вузлами моделюються, базуючись на розумінні транспортної системи та на основі аналізу даних. Наприклад, існує залежність між Reason та How_Long_Delayed, оскільки тип несправності безпосередньо впливає на тривалість затримки. Також, School_Age_or_PreK і Boro можуть впливати на Reason і How_Long_Delayed. Ці залежності визначають структуру БМД.

Для обчислення умовних ймовірностей (CPD) для кожного вузла використовується частотний підхід, базуючись на історичних даних. Наприклад, щоб розрахувати CPD для вузла `How_Long_Delayed`, треба аналізувати, як часто зустрічаються різні тривалості затримки для кожного значення вузла `Reason`.

Умовна ймовірність обчислюється як:

$$P(\text{HowLongDelayed} = t | \text{Reason} = r) = \frac{\text{count}(\text{HowLongDelayed}=t \text{ and } \text{Reason}=r)}{\text{count}(\text{Reason}=r)},$$

де `count()` – функція, що обчислює кількість випадків, коли виконується певна умова в наборі даних.

Інтеграція генеративного штучного інтелекту. Генеративні алгоритми являють собою клас методів машинного навчання, які здатні створювати нові дані, схожі на тренувальні зразки [8]. У контексті транспортних систем, ці алгоритми дозволяють генерувати синтетичні сценарії пасажиропотоків, що має критичне значення для тестування та оптимізації систем управління.

Основні типи генеративних моделей:

- Генеративні змагальні мережі (GAN) - базуються на концепції змагання між двома нейронними мережами;
- Варіаційні автоенкодера (VAE) - використовують ймовірнісний підхід до кодування даних;
- Дифузійні моделі - моделюють процес поступового додавання та видалення шуму.

У запропонованому підході використовуються GAN з наступних причин:

Недостатність реальних даних: рідкісні події (аварії, святкові дні, надзвичайні ситуації) трапляються нечасто. GAN дозволяють згенерувати синтетичні дані для цих сценаріїв.

Збереження складних залежностей: пасажиропотоки мають складні часові та просторові закономірності. GAN здатні відтворити ці нелінійні залежності краще за традиційні статистичні методи.

Адаптивність до реальних умов: дискримінатор забезпечує реалістичність згенерованих даних. Змагальний процес навчання створює високоякісні синтетичні сценарії.

Розширення тестового середовища: можливість тестувати транспортну систему в умовах, які ще не спостерігалися. Перевірка стійкості алгоритмів оптимізації до різних сценаріїв.

Інтеграція з БМД: Синтетичні дані покращують навчання БМД. Збільшується точність прогнозування через більший обсяг даних.

GAN вирішують проблему обмеженості реальних даних, створюючи реалістичні альтернативні сценарії для кращого навчання та тестування системи оптимізації [9].

GAN, які складаються з двох компонентів:

Генератор та дискримінатор: Генератор приймає випадковий шум та перетворює його в синтетичні дані, що імітують реальний розподіл пасажиропотоків.

Дискримінатор визначає ймовірність того, що вхідні дані x є реальними, а не згенерованими.

Процес навчання GAN базується на мінімакс-грі між генератором та дискримінатором: генератор намагається «обдурити» дискримінатор, створюючи дедалі більш реалістичні дані, а дискримінатор намагається відрізнити реальні дані від згенерованих. У результаті цього процесу генератор навчається створювати синтетичні дані, які важко відрізнити від реальних.

Лістинг програмного коду побудови генератора та дискримінатора:

```
def build_generator(latent_dim, output_dim):
    return models.Sequential([
        layers.Dense(16, activation='relu', input_dim=latent_dim),
        layers.Dense(32, activation='relu'),
```

```

        layers.Dense(output_dim, activation='linear')
    ])

def build_discriminator(input_dim):
    model = models.Sequential([
        layers.Dense(32, activation='relu', input_dim=input_dim),
        layers.Dense(16, activation='relu'),
        layers.Dense(1, activation='sigmoid')
    ])
    model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
    return model
    
```

Генерація синтетичних сценаріїв. Генеративні алгоритми створюють реалістичні сценарії пасажиропотоків для:

- Моделювання пікових навантажень;
- Симуляції надзвичайних ситуацій;
- Тестування нових маршрутів;
- Прогнозування впливу подій на транспортну систему.

Алгоритм генерації сценаріїв:

1. Збір історичних даних пасажиропотоків;
2. Препроцесинг та нормалізація даних;
3. Навчання GAN на історичних даних;
4. Генерація синтетичних сценаріїв;
5. Валідація якості згенерованих даних;
6. Інтеграція з байєсовою мережею для оптимізації.

Інтеграція байєсових мереж довіри з генеративними алгоритмами здійснюється через:

- Фаза навчання: GAN генерує додаткові тренувальні сценарії для БМД;
- Фаза інференсу: БМД використовує згенеровані сценарії для прогнозування.

На рис. 1 зображена архітектуру інтеграції.

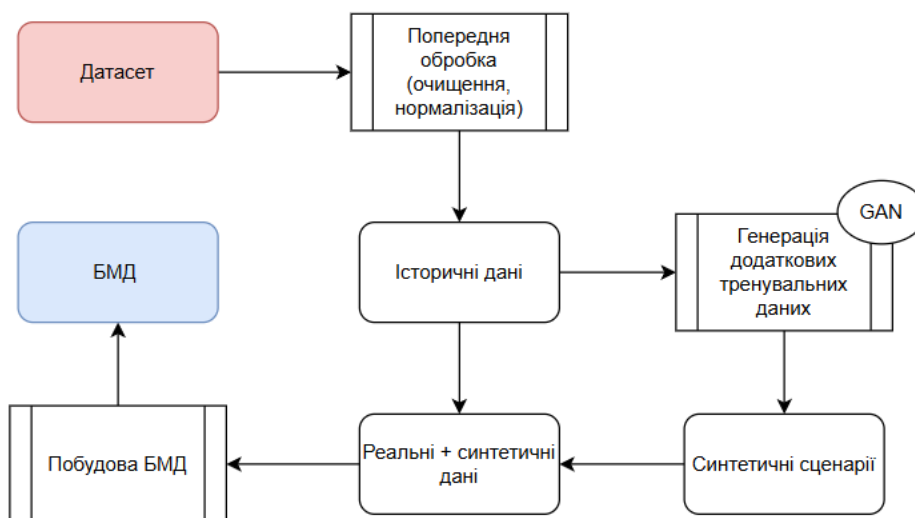


Рис. 1. Архітектура інтеграції

Джерело: розроблено авторами.

Після збору, очищення та масштабування даних створюються структуровані таблиці даних, які використовуються для тренування, для реалізації даного підходу були використані інструменти Python, повний код та приклад вхідних даних було завантажено у публічний репозиторій: <https://github.com/DomCore/BMD>. В даній статті демонструється лише основна частина функціонального коду.

На етапі підготовки даних здійснюється очищення та нормалізація числових показників. Дані розбиваються на категорії або ж нормалізуються для забезпечення уніфікації шкал для подальшої генерації та моделювання.

Потім генеруються синтетичні дані. Модель дозволяє симулювати ситуації, що не були присутні у реальному наборі, зберігаючи структуру та статистичні властивості. Наприклад, можна моделювати сценарії перевантаження у неробочі дні або в умовах погіршення погоди.

Лістинг програмного коду генерування синтетичних даних:

```
def generate_synthetic_data(real_data, synthetic_frac=0.2, latent_dim=5,
epochs=10, batch_size=10000):
    features = ['How_Long_Delayed', 'Number_Of_Students_On_The_Bus']
    data = real_data[features].dropna().values
    data_dim = data.shape[1]

    generator = build_generator(latent_dim, data_dim)
    discriminator = build_discriminator(data_dim)
    # Build GAN model
    gan_input = layers.Input(shape=(latent_dim,))
    fake_data = generator(gan_input)
    gan_output = discriminator(fake_data)
    gan = models.Model(gan_input, gan_output)
    gan.compile(optimizer='adam', loss='binary_crossentropy')

    # Pre-create labels for efficiency
    real_labels = np.ones((batch_size, 1))
    fake_labels = np.zeros((batch_size, 1))

    # Training loop..
```

Наступним кроком є побудова БМД.

Лістинг програмного коду побудови мережі:

```
def build_bn(df, bn_nodes, potential_edges):
    existing_nodes = [node for node in bn_nodes if node in df.columns]
    G = nx.DiGraph()
    G.add_nodes_from(existing_nodes)
    for parent, child in potential_edges:
        if parent in existing_nodes and child in existing_nodes:
            G.add_edge(parent, child)
    return G

# Візуалізація
plot_bn_structure(G)
```

Далі продемонстровано структуру байєсової мережі у форматі JSON, сформовану в результаті роботи системи.

Результат роботи, вигляд JSON:

```
{
  "nodes": [
    "School_Age_or_PreK",
    "Reason",
    ...
  ],
  "edges": [
```

```
{
  "parent": "School_Age_or_PreK",
  "child": "Reason",
  "cpd": {
    "1": {
      "3": 0.5877862595419847,
      "6": 0.17938931297709923,
      "5": 0.11259541984732824,
      "9": 0.04389312977099236,
      "2": 0.04007633587786259,
      "4": 0.01717557251908397,
      "0": 0.007633587786259542,
      "7": 0.0057251908396946565,
      "1": 0.0057251908396946565
    },
    "0": {
      "3": 0.6901408450704225,
      "6": 0.18309859154929578,
      "5": 0.07042253521126761,
      "2": 0.028169014084507043,
      "9": 0.014084507042253521,
      "7": 0.014084507042253521
    },
    "2": {
      "8": 1.0
    },
    "-1": {
      "-1": 1.0
    }
  }
},
{
  "parent": "Reason",
  "child": "Breakdown_or_Running_Late",
  "cpd": {
    "3": {
      "1": 0.9971988795518207,
      "0": 0.0028011204481792717
    },
    "2": {
      "1": 0.5217391304347826,
      "0": 0.4782608695652174
    },
    "6": {
      "1": 0.897196261682243,
      "0": 0.102803738317757
    }
  },
}
```

Кожна колонка подана у вигляді ноди об'єкта із зазначенням її назви та умовних ймовірностей (CPD), які описані у відповідних гранях, що представлені у форматі JSON. Тут відображено причинно-наслідкові залежності, наприклад, як причина несправності (Reason) впливає на тривалість затримки (How_Long_Delayed), або як район (Boro) сприяє виникненню конкретних ситуацій. Значення "0", "1" – це закодовані категоріальні значення вузла Boro (наприклад, "0" – Manhattan, "1" – Brooklyn).

Таблиця `cpd` – словник, де ключі – значення `parent`, а значення – словники, що описують ймовірності кожного варіанту `child`.

Детальний опис елементів:

- Ключ "nodes" – це список усіх вузлів (нод) у вашій байєсівській мережі. Наприклад, вузол "School_Age_or_PreK" представляє одну з категорій, на основі яких побудовано мережу, а інші вузли – це окремі змінні, що використовуються як батьківські або дочірні;

- Ключ "edges" – це список ребер (зв'язків) між вузлами. Кожний елемент списку є об'єктом з наступними властивостями:

- "parent" – вказує на назву вузла, який є батьком, тобто значення якого впливають на розподіл дочірнього вузла;

- "child" – це дочірній вузол, для якого розраховано умовний розподіл залежно від значення ноди "parent";

- "cpd" – умовний розподіл для дочірнього вузла "child".

Наприклад, у першому елементі: "parent": "School_Age_or_PreK" означає, що для вузла "Reason" розраховано CPD залежно від значень вузла "School_Age_or_PreK".

Всередині "cpd" кожен ключ (наприклад, "1", "0", "2", "-1") позначають конкретне значення, яке може приймати батьківський.

Для кожного значення батька, вкладений словник показує, як розподіляються значення для вузла "Reason".

Наприклад, коли значення батька "School_Age_or_PreK" є рівним "1", то:

- "3": 0.5877862595419847 означає, що "Reason" має приблизно 58,78 % шанс бути рівним 3;

- "6": 0.17938931297709923 — близько 17.94% шанс бути 6, і так далі.

Таким чином, читаючи цей JSON можна отримати повну картину:

- Які вузли входять в мережу;

- Які є зв'язки між цими вузлами.

Такий JSON-документ є ключовим для подальшої автоматизації аналізу і інтеграції моделі в системи ухвалення рішень, дозволяючи прагматично використовувати модель для розрахунків і прогнозів у реальному часі. Для більшої зручності у моделюванні та прогнозуванні цю JSON-структуру можна використовувати у системах автоматичного аналізу та роботі з даними.

Вона дозволяє імпортувати всю ймовірнісно-залежну структуру та таблиці ймовірностей у інші програми або бібліотеки для подальших обчислень.

Експериментальні результати. Для оцінки впливу синтетичних даних, згенерованих за допомогою GAN, було проведено серію експериментів з двома класифікаційними моделями: Random Forest та Logistic Regression. Порівняння проводилось між моделями, навченими на оригінальному наборі даних, та моделями, що додатково використовували синтетичні дані.

Модель Random Forest, навчена на розширеному наборі даних, показала зростання загальної точності з 0.95 до 0.97. Особливо помітним є покращення точності по класу "Breakdown" – з 0.85 до 0.92, що свідчить про зменшення кількості хибнопозитивних прогнозів. При цьому чутливість залишилася на рівні 0.79, що вказує на збереження здатності моделі виявляти справжні випадки поломок. F1-метрика зросла з 0.81 до 0.85, демонструючи збалансоване покращення продуктивності.

Таблиця 2 – Результати експериментів

Модель	Точність	Точність Breakdown	Чутливість Breakdown	F1-score	Data Source
Random Forest	0,97	0,92	0,79	0,85	Розширені дані
Logistic Regression	0,80	0,36	0,93	0,52	Розширені дані
Random Forest	0,95	0,85	0,79	0,81	Оригінальні дані
Logistic Regression	0,68	0,29	0,86	0,43	Оригінальні дані

Джерело: розроблено авторами.

Для Logistic Regression вплив синтетичних даних виявився ще більш вираженим. Точність зросла з 0.68 до 0.80, а чутливість – з 0.86 до 0.93, що свідчить про суттєве зростання здатності моделі виявляти поломки. Незважаючи на те, що точність залишився відносно низьким (зріс з 0.29 до 0.36), загальне покращення F1-score з 0.43 до 0.52 демонструє ефективніше балансування між виявленням і точністю класифікації.

Аналіз результатів. Особливо варто відзначити, що:

- Random Forest отримала покращення точності класифікації завдяки зменшенню кількості хибнопозитивних прогнозів;
- Logistic Regression значно покращила чутливість, що дозволяє виявляти більшу кількість реальних поломок.

Ці результати підтверджують ефективність підходу використання GAN для покращення класифікаційної здатності моделей, особливо при роботі з менш представленими класами.

Висновки: У ході дослідження було розроблено комплексний підхід до моделювання та аналізу завантаженості міського громадського транспорту, який поєднує машинне навчання та ймовірнісний аналіз. Основою є БМД, яка відображає складні зв'язки між ключовими факторами: причиною несправності, районом розташування, рівнем затримки, кількістю пасажирів тощо.

Для розширення навчальних даних використовувалися генеративні нейронні мережі, що дозволило створити більш різноманітний набір даних для тренування системи. Результати були зведені у JSON-структуру, що відображає ієрархію змінних, їх залежності та умовні ймовірності. Такий підхід спрощує інтеграцію моделі в сучасні системи автоматичного аналізу та управління, забезпечує прозорість та гнучкість прийняття рішень.

Отримані результати показали, що БМД дозволяють точно і швидко аналізувати вплив різних факторів у реальному часі. Створена модель здатна ідентифікувати критичні залежності та надавати рекомендації щодо оптимізації маршрутів, розкладів і плану обслуговування.

Загалом, застосування сучасних ітеративних моделей та ймовірнісних підходів є перспективним напрямком для розробки високотехнологічних систем управління міським транспортом, здатних адаптуватися до умов, що швидко змінюються, та забезпечити сталий розвиток транспортної інфраструктури.

Список використаних джерел

1. Singh, A., & Singh, K. K. (2025). *Multimodal generative AI*. Springer Nature Singapore.
2. Macioszek, E., Sierpiński, G., & Masoumi, H. (2023). *Challenges and solutions for present transport systems*. Springer Nature Singapore.
3. Vegni, A. M., Vasilakos, A. V., & Loscrì, V. (2017). *Vehicular social networks*. CRC Press.
4. Bouhsissin, S., Sael, N., & Benabbou, F. (2022). Prediction of risks in intelligent transport systems. In M. Lazaar, C. Duvallet, A. Touhafi, & M. Al Achhab (Eds.), *Proceedings of the 5th International Conference on Big Data and Internet of Things (BDIoT 2021)* (Vol. 489, pp. 303–316). *Lecture Notes in Networks and Systems*. Springer, Cham. https://doi.org/10.1007/978-3-031-07969-6_23.
5. Martinez-Ramon, M., Ajith, M., & Kurup, A. R. (2024). *Deep learning*. Wiley.
6. Benaini, A., & Boukachour, J. (2023). *Transport and logistics planning and optimization*. IGI Global.
7. Taroni, F., Biedermann, A., Bozza, S., Garbolino, P., & Aitken, C. (2014). *Bayesian networks for probabilistic inference and decision analysis in forensic science*. Wiley.

8. Kumar, R., Sahu, S., & Bhattacharya, S. (2024). *The pioneering applications of generative AI*. IGI Global.
9. Pathak, P. D., Raut, R., Sakhare, S. R., & Patil, S. (2023). *Generative adversarial networks and deep learning*. CRC Press.
10. Kaggle. (2025, February 26). *NY bus breakdown and delays*. <https://www.kaggle.com/datasets/new-york-city/ny-bus-breakdown-and-delays>.

References

1. Bouhsissin, S., Sael, N., & Benabbou, F. (2022). Prediction of risks in intelligent transport systems. In M. Lazaar, C. Duvallet, A. Touhafi, & M. Al Achhab (Eds.), *Proceedings of the 5th International Conference on Big Data and Internet of Things (BDIoT 2021)* (Vol. 489, pp. 303–316). *Lecture Notes in Networks and Systems*. Springer, Cham. https://doi.org/10.1007/978-3-031-07969-6_23.
2. Martinez-Ramon, M., Ajith, M., & Kurup, A. R. (2024). *Deep learning*. Wiley.
3. Benaini, A., & Boukachour, J. (2023). *Transport and logistics planning and optimization*. IGI Global.
4. Taroni, F., Biedermann, A., Bozza, S., Garbolino, P., & Aitken, C. (2014). *Bayesian networks for probabilistic inference and decision analysis in forensic science*. Wiley.
5. Kumar, R., Sahu, S., & Bhattacharya, S. (2024). *The pioneering applications of generative AI*. IGI Global.
6. Pathak, P. D., Raut, R., Sakhare, S. R., & Patil, S. (2023). *Generative adversarial networks and deep learning*. CRC Press.
7. Kaggle. (2025, February 26). *NY bus breakdown and delays*. <https://www.kaggle.com/datasets/new-york-city/ny-bus-breakdown-and-delays>.

Отримано 02.06.2025

UDC 004.62

Vladyslav Zetchenko¹, Artem Zadorozhnyi²

¹PhD student of the Department of Information Technologies and Software Engineering
Chernihiv Polytechnic National University (Chernihiv, Ukraine)

E-mail: zetchenkovlads@gmail.com. **ORCID:** <https://orcid.org/0009-0002-9925-8628>. **ResearcherID:** [ITW-2651-2023](https://orcid.org/0009-0002-9925-8628)

²PhD in Technical Sciences, associate professor of the Department of Information Technologies and
Software Engineering

Chernihiv Polytechnic National University (Chernihiv, Ukraine)

E-mail: zaotroy@gmail.com. **ORCID:** <https://orcid.org/0000-0002-3424-7293>. **ResearcherID:** [F-6358-2016](https://orcid.org/0000-0002-3424-7293)

CONSTRUCTION OF BAYESIAN TRUST NETWORKS FOR FORECASTING PASSENGER FLOWS OF PUBLIC TRANSPORT USING GENERATIVE ARTIFICIAL INTELLIGENCE

The article is devoted to the topical issue of managing congestion in public transport systems. Currently, urban population growth and increased stress on transport infrastructure are being observed, emphasizing urgency of ensuring effective management to enhance passenger comfort and reduce operating costs. In the article, recent research in this field is studied, which employs various approaches, including probabilistic models, generative adversarial networks, and optimization methods.

The article proposes the approach to forecasting and modeling passenger flows in public transportation, based on the integration of Bayesian belief networks—as a tool for probabilistic modeling of dependencies between various factors—and generative artificial intelligence, in particular representatives, including Generative Adversarial Networks and Variational Autoencoders, which allow modeling and simulating potential scenarios in transport systems. This synergy of approaches enables accounting for uncertainty and dynamic changes in the transport network, as well as generating synthetic data in cases where there is a shortage of real measurements—a typical problem in transport modeling.

The study is based on the analysis of empirical data from the period 2015–2018, demonstrating the practical effectiveness of the proposed model. The results confirm the approach's ability to provide more precise load forecasting, respond more promptly to changing conditions, and form adaptive transport management strategies. The proposed tool can be useful for urban planning authorities, transport operators, and developers of intelligent transportation systems. Thus, the article has both scientific and applied value and opens prospects for further research in the field of integrating generative AI with probabilistic analysis in transport modeling. Traffic statistics for the period from 2015 to 2018 are provided.

Keywords: Bayesian trust networks, generative artificial intelligence, public transport, congestion forecasting.

Figures: 2. **Tables:** 2. **References:** 10.