Remerciements

Nous souhaitons exprimer notre profonde gratitude à toutes les personnes qui ont contribué à la réalisation de ce projet de recherche.
Tout d'abord, nous adressons nos sincères remerciements à
Monsieur Azim ROUSSANALY, notre tuteur, pour son encadrement, sa patience et ses précieux conseils tout au long de ce projet. Son expertise et son soutien nous ont été d'une aide précieuse pour mener à bien cette recherche.
Nous tenons également à remercier **Madame Marie-Laure ALVES** pour son aide précieuse dans la rédaction de ce rapport. Ses recommandations et ses retours constructifs nous ont permis d'améliorer la qualité de notre travail.
Enfin, nous adressons notre reconnaissance à toutes les personnes ayant contribué, de près ou de loin, à la réussite de ce projet.

Sommaire

Table des matières

Introduction **3**

1 État de l'art **5**
1.1 Données xAPI 5
1.2 Algorithmes de génération de séries temporelles 6
1.2.1 Description de l'algorithme CTGAN 7
1.3 Évaluation des données synthétiques 8
1.3.1 Critères d'évaluation des données générées 8
1.3.2 Méthodes d'analyse utilisées 10

2 Travail réalisé **12**
2.1 Conception et développement de l'application 12
2.1.1 Exigences et finalités de l'application 12
2.1.2 Conception de l'interface utilisateur 12
2.1.3 Architecture logicielle 17
2.2 Implémentation du modèle 18
2.2.1 Préparation des données 18
2.2.2 Modélisation de la génération 19
2.2.3 Modélisation de la sortie 20
2.2.4 Méthodes d'évaluation utilisées 21
2.2.5 Validation et contrôle qualité des données synthétiques générées 22

3 Résultats **23**
3.1 Analyse statistique des données 23
3.2 Analyse de la confidentialité des données 24
3.3 Analyse de la fidélité des données 24
3.4 Conclusion 25
3.5 Cas d'usage de l'application 26

4 Conclusion **28**

Introduction

La plateforme **LOLA** (Laboratoire Ouvert en Learning Analytics), développée au sein du **LORIA** et portée par le projet européen **EDGE Skills**, joue un rôle clé dans le développement d'une intelligence artificielle **fiable** et **éthique** en éducation. LOLA vise à soutenir l'évaluation des **technologies éducatives** dans un cadre sécurisé et conforme aux exigences européennes. Elle s'inscrit alors pleinement dans la mise en œuvre de l'**AI Act** qui impose à la fois une **analyse des risques** et des exigences de **transparence** pour les systèmes d'intelligence artificielle. À travers un **dataspace souverain**, LOLA structure ses travaux autour de trois piliers : des **datasets** fournis par les entreprises EdTech, des **algorithmes** mis à disposition par les chercheurs et développeurs, et des **scénarios d'évaluation** définis par des auditeurs sur la base de protocoles rigoureux.

Cependant, l'un des enjeux majeurs réside dans l'impossibilité d'extraire les **données réelles** de la plateforme, ce qui empêche les utilisateurs d'ajuster leurs algorithmes avant l'audit. Les **données synthétiques** constituent alors une solution efficace pour dépasser les contraintes liées à la collecte de **données personnelles**. Créées artificiellement pour simuler les propriétés de données réelles, elles peuvent être utilisées pour faire évoluer et tester des **modèles d'apprentissage** tout en satisfaisant aux contraintes éthiques et juridiques. Elles jouent ainsi un rôle central dans la phase de **tuning** des modèles d'apprentissage en permettant leur optimisation sans compromettre la **confidentialité**. Ces données sont particulièrement utiles pour créer ou reproduire des **séries temporelles**, très fréquentes dans le domaine des **Learning Analytics**. Il s'agit de séquences d'événements ordonnés chronologiquement (comme les interactions d'un élève avec une plateforme), qui sont indispensables pour le développement de **modèles prédictifs** ou d'analyse du **comportement d'apprentissage**.

Pour répondre à ces défis, notre projet de recherche consiste à concevoir et développer une **application graphique** simple d'utilisation, qui intègre un **algorithme de génération de séries temporelles**. Dans notre cas, ces séries sont construites à partir des **données du CNED** (Centre National d'Enseignement à Distance) et décrivent les interactions des utilisateurs avec la plateforme éducative. L'objectif est de produire des séries temporelles artificielles qui représentent fidèlement l'évolution des **parcours d'apprentissage**, tout en respectant les règles strictes de **confidentialité**.

Pour garantir la **qualité** et la **cohérence** des données synthétiques, nous nous appuyons sur des **études comparatives** réalisées lors de précédents projets de master. Ces travaux nous ont permis d'identifier deux approches

complémentaires : la première consiste à générer des séries qui correspondent à des **sessions d'utilisateur** précises et pour la seconde, à reconstruire en détail les **actions effectuées** pendant ces sessions. En combinant ces deux méthodes nous cherchons à mieux reproduire les **comportements d'apprentissage**, l'**organisation temporelle des interactions** et les **liens entre les différents événements**. Notre but est de créer des données synthétiques **proches de la réalité**, tout en assurant une **anonymisation complète**.

1 État de l'art

Dans le domaine de l'**éducation numérique**, les **technologies éducatives (EdTech)** désignent l'ensemble des outils, plateformes, méthodes et algorithmes visant à enrichir, personnaliser et rendre plus accessibles les expériences d'apprentissage. Ces technologies s'appuient fortement sur la collecte et l'analyse de données pour suivre les parcours des apprenants, adapter les contenus et améliorer l'efficacité des dispositifs pédagogiques. Dans ce cadre, la question de la qualité, de la traçabilité et de l'**anonymisation** des données devient cruciale, tant pour l'analyse que pour le respect des exigences légales (**RGPD**, **AI Act**).

Parmi les standards adoptés dans l'écosystème EdTech, **xAPI (Experience API)** s'impose comme un protocole flexible permettant de capturer et de structurer une grande variété d'interactions. Contrairement à d'anciens formats comme **SCORM**, xAPI permet de suivre plus précisément les activités d'apprentissage, même en dehors des plateformes classiques. C'est ce qui en fait un choix pertinent lorsqu'on veut créer des **données synthétiques** à partir de traces d'apprentissage, en particulier lorsqu'elles sont combinées avec les avancées récentes en **intelligence artificielle générative (GenAI)** [1].

De plus, le développement de l'**intelligence artificielle générative (GenAI)**, en particulier à travers des modèles comme les **GANs** et plus spécifiquement **CTGAN** dans notre projet, permet de créer automatiquement des données simulées. Cela permet de contourner les limites liées à l'accès aux données sensibles, tout en produisant des jeux de données d'entraînement utiles pour tester, valider et améliorer des modèles d'apprentissage.

En combinant les apports de xAPI, des technologies éducatives (EdTech) et de l'IA générative (GenAI), on ouvre la voie à des formes d'apprentissage plus intelligentes, personnalisées et respectueuses de la vie privée. La suite de cette section présente d'abord le fonctionnement de xAPI, avant d'aborder les méthodes de génération de données temporelles synthétiques.

1.1 Données xAPI

xAPI (Experience API) [2], souvent surnommée **Tin Can API** , est un ensemble de règles permettant de suivre, d'enregistrer et de partager toute expérience d'apprentissage. Cette expérience d'apprentissage peut se produire en ligne ou hors ligne, au sein ou en dehors d'un système de gestion de l'apprentissage. Concrètement, xAPI enregistre les expériences et les activités de l'apprenant sous forme de déclarations (*statements*). Cela permet

de suivre toutes sortes d'interactions (cours, jeux, expériences en situation, etc.).

Voici un exemple de donnée xAPI :

```
{
  "actor": {
    "name": "Jean Dupont",
    "mbox": "mailto:jean.dupont@example.com"
  },
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/completed",
    "display": { "en-US": "completed" }
  },
  "object": {
    "id": "http://example.com/cours/intro-xapi",
    "definition": {
      "name": { "en-US": "Introduction à xAPI" }
    },
    "objectType": "Activity"
  },
  "timestamp": "2025-04-20T14:00:00Z"
}
```

Chaque déclaration xAPI est structurée autour de trois éléments principaux :

— **L'acteur** (*actor*) : qui a réalisé l'action.
— **Le verbe** (*verb*) : qui décrit l'action effectuée.
— **L'objet** (*object*) : sur lequel porte l'action.

Le champ `timestamp` indique la date et l'heure où l'action a été réalisée.

1.2 Algorithmes de génération de séries temporelles

Les algorithmes de génération de **séries temporelles** ont pour but de créer artificiellement des données qui imitent le comportement de séries temporelles réelles. Les séries temporelles représentent des suites de données qu'on enregistre au fil du temps. Parfois à intervalles réguliers, parfois non, et chaque point représente une observation faite à un moment donné. Elles sont indispensables pour l'étude des tendances et des comportements sur une période spécifique. Dans ce projet, on enregistre des actions sous forme de séries temporelles, comme les connexions des utilisateurs, la consultation des cours et d'autres interactions avec la plateforme du CNED.

Générer des **séries temporelles synthétiques** permet d'avoir beaucoup de données sans avoir à aller les chercher dans la réalité. C'est souvent compliqué ou même risqué de collecter des données réelles, surtout quand elles sont sensibles. Avec des données artificielles, on peut travailler librement, sans exposer d'informations personnelles. C'est aussi une solution plus rapide et moins coûteuse que de passer par des campagnes de collecte longues et lourdes. De plus, l'utilisation de **données synthétiques** permet de préserver la **confidentialité** des utilisateurs et d'éviter les risques liés à l'exploitation d'informations personnelles.

Pour être efficaces, les algorithmes de génération doivent non seulement capturer les relations temporelles (par exemple, la façon dont un événement influence le suivant), mais aussi maintenir les distributions statistiques des différentes variables. Un bon algorithme doit produire des données suffisamment réalistes pour être utilisées à des fins d'entraînement de modèles, de test ou d'analyse, tout en assurant que les données ne peuvent pas être reliées à des individus réels.

Pour notre projet, nous nous sommes basés sur une **étude réalisée** [3] par d'autres étudiants dans le cadre d'un projet de master. Dans leur étude, plusieurs algorithmes de génération de données ont été testés, parmi lesquels **PAR**, **TimeGAN** et **CTGAN**. Après analyse, il a été constaté que PAR et TimeGAN n'étaient pas adaptés aux données de type **XAPI**, qui contiennent beaucoup de variables **catégorielles**.
Suite à ces conclusions, l'algorithme **CTGAN** a finalement été choisi car il est le plus adapté à la génération de données tabulaires contenant à la fois des variables **continues** et **catégorielles**.

1.2.1 Description de l'algorithme CTGAN

Les **Generative Adversarial Networks (GANs)** [4] sont une famille de modèles d'apprentissage automatique conçus pour générer des données synthétiques. Leur principe repose sur l'opposition de deux réseaux neuronaux : un **générateur**, qui tente de produire des échantillons réalistes, et un **discriminateur**, qui essaie de distinguer les échantillons réels des synthétiques. Au fil de l'entraînement, le générateur s'améliore et parvient à produire des données qui ressemblent de plus en plus aux vraies.

CTGAN (Conditional Tabular Generative Adversarial Network) est une adaptation de cette architecture, développée par Xu et al. (2019), conçu pour produire des ensembles de **données tabulaires** comprenant des variables **continues** ainsi que **catégorielles**. Contrairement aux GANs traditionnels, souvent utilisés pour des images ou des séries continues, CTGAN

est optimisé pour gérer la structure mixte des bases de données classiques.

CTGAN suit le principe général des GANs : le générateur et le discriminateur s'entraînent ensemble de manière compétitive. Ce qui distingue CTGAN, c'est sa capacité à traiter efficacement les variables **catégorielles** grâce à un **échantillonnage conditionnel**, ce qui permet de préserver les distributions et les relations entre catégories dans les données synthétiques. Pour les variables **continues**, une **transformation logarithmique** est appliquée afin de mieux modéliser les distributions asymétriques.

Cependant, CTGAN n'est pas initialement conçu pour modéliser directement l'aspect **temporel** des séries chronologiques. Afin de préserver une certaine organisation temporelle dans les **données XAPI**, les **timestamps** ont été convertis en secondes, puis les événements ont été triés en fonction de cette valeur. Cette adaptation permet de conserver partiellement la structure temporelle des interactions utilisateurs.

Timecode	**Acteur**	**Verbe**	**Objet**
512348	198452	completed	resource_987654
7241563	215963	viewed	course_2035
1548923	302458	attempted	quiz_12
8627412	178654	viewed	course_2035
4839201	143256	completed	resource_987654

TABLE 1 – Exemple de données généré par CTGAN

1.3 Évaluation des données synthétiques

1.3.1 Critères d'évaluation des données générées

La gestion de données est un domaine très sensible, notamment dans le cadre des données synthétiques générées pour divers secteurs, tels que la santé, l'éducation ou la finance, entre autres. Elle est fondée sur trois concepts de base pour évaluer la qualité des données générées : la confidentialité, l'anonymat et la fidélité des données. Bien qu'ils ne soient pas étroitement liés, ces trois concepts sont de facto interdépendants et jouent un rôle essentiel dans la finalité de l'accès et de l'utilisation des données synthétiques, particulièrement lorsqu'il s'agit de projets d'IA ou d'apprentissage automatique.

Confidentialité : La confidentialité fait partie des piliers fondamentaux de la sécurité de l'information en informatique. Elle consiste à s'assurer que seules les personnes autorisées peuvent lire des informations sensibles, et ce, sans que l'information ne soit lue ou utilisée par un tiers. Elle est réglemen-

tée par des normes très strictes, comme le RGPD en Europe, qui impose des normes de sécurité très strictes aux entreprises. Pour assurer la confidentialité dans les données synthétiques, de nombreux moyens techniques sont utilisés (chiffrement des données, authentification forte des usagers, etc.), mais également organisationnels (politique de sécurité, sensibilisation des utilisateurs, audit régulier, etc.).

Alors que les volumes de données ne cessent d'augmenter, notamment avec l'arrivée du Big Data, mais aussi de l'intelligence artificielle, les enjeux de confidentialité deviennent particulièrement critiques, non seulement pour la protection des libertés individuelles, mais aussi pour préserver la confiance dans le numérique et se conformer aux règles juridiques et morales. C'est donc un enjeu majeur dans le développement de données synthétiques destinées à alimenter des algorithmes de machine learning.

Anonymisation : L'anonymisation des données est un processus très important dans la lutte pour la vie privée à l'ère numérique, y compris pour les données synthétiques. Elle vise la transformation ou la suppression de toutes les informations d'identification personnelle d'un jeu de données de départ pour qu'il soit impossible de revenir et d'identifier une personne, que ce soit directement ou indirectement.

Contrairement à la pseudonymisation, qui conserve des liens avec l'identité des personnes via un identifiant, l'anonymisation vise à exclure définitivement tout lien avec une personne physique. Dans de nombreux domaines tels que la santé, les transports ou l'éducation, elle est même souvent une nécessité pour prendre en charge les usagers d'une manière sécurisée.

Cependant, un des plus grands défis de l'anonymisation des données est d'équilibrer entre la confidentialité et la richesse analytique des données synthétiques. Plus un fichier est anonymisé, plus il est difficile d'extraire des analyses précises. De fait, l'anonymisation doit être faite d'une manière stratégique, en fonction du contexte d'utilisation et du niveau de risque acceptable de ré-identification.

Fidélité : La qualité des données synthétiques dépend de la capacité des informations qu'elles contiennent à être fiables, à ne pas contenir de valeurs manquantes ou erronées, et à être correctement formatées. Lorsqu'on génère des millions de données dans le cadre d'applications ou d'algorithmes d'intelligence artificielle, il est essentiel de disposer de données de haute qualité pour obtenir des résultats exploitables.

Un jeu de données synthétiques de qualité doit être exact, complet, ne pas contenir de données erronées, sans valeur manquante non justifiée (et non imputée), et cohérent. Les règlementations applicables aux fichiers de données personnelles imposent souvent de les transformer pour les protéger et rendre impossible l'identification, sans pour autant changer le contenu ou

réduire la précision de l'information.

À la frontière entre protection de la vie privée et information pertinente, les propositions de transformation des données varient d'un métier à l'autre (ou de projet à projet) et l'enjeu économique des informations non modifiées est grand. Maintenir un bon niveau de qualité des informations dans le respect de ces obligations de protection des données personnelles est un enjeu pour tout un chacun aujourd'hui, notamment dans certains domaines critiques où des politiques publiques intelligentes nécessitent des informations de haute qualité.

1.3.2 Méthodes d'analyse utilisées

Dans une **étude précédente** [3], plusieurs **méthodes d'analyse** ont été élaborées et appliquées pour évaluer la qualité des **données synthétiques générées**. Cette évaluation s'est structurée autour de trois grands axes fondamentaux : **la confidentialité**, **l'utilité** et **la fidélité** des données.

Concernant la confidentialité, des tests statistiques ont été utilisés pour vérifier si les données synthétiques différaient suffisamment des données originales afin de garantir l'anonymat :

- **Test du χ^2 (Chi-deux)** pour comparer les distributions des variables catégorielles entre jeux de données.
- **Test du χ^2 de contingence** pour analyser les dépendances entre différentes variables.
- **Coefficient de contingence** et **V de Cramer** pour mesurer la force d'association entre variables et leur préservation dans les données synthétiques.
- **Distance au plus proche enregistrement (DCR)** pour estimer la proximité individuelle entre données réelles et synthétiques et évaluer ainsi les risques de ré-identification.

Pour évaluer l'utilité des données synthétiques, plusieurs métriques classiques d'apprentissage automatique ont été utilisées :

- **F1-Score**, pour mesurer la performance de modèles de classification entraînés sur les données synthétiques.
- **MAPE (Mean Absolute Percentage Error)** et **pMSE (Propensity Mean Squared Error)**, pour quantifier la capacité à reproduire les comportements prédictifs observés sur les données originales.

La fidélité des données a été quantifiée par différentes mesures de similarité entre distributions :

- **TVC (Total Variation Correlation)** pour comparer directement les répartitions.
- **MSE (Mean Squared Error)** pour mesurer les écarts quadratiques moyens.

— **Divergence de Kullback-Leibler (KL Divergence)** et **Entropie croisée** pour estimer la distance et la similarité informationnelle entre jeux de données.
— **Analyse en Composantes Principales Temporelle (t-ACP)** pour visualiser la conservation de la structure temporelle et spatiale des données.

Ces méthodes d'analyse ont permis d'avoir une vision complète sur la qualité des données synthétiques à la fois d'un point de vue statistique et fonctionnel, et servent aujourd'hui de base méthodologique pour être réutilisées et adaptées dans notre nouvelle étude.

2 Travail réalisé

2.1 Conception et développement de l'application

2.1.1 Exigences et finalités de l'application

L'application développée a pour objectif de permettre la génération de séries temporelles synthétiques à partir de données réelles issues de **plateformes éducatives**. L'objectif principal est de produire des données artificielles fidèles aux caractéristiques des données originales, tout en garantissant un haut niveau de confidentialité afin d'éviter tout risque de **réidentification** des individus, notamment dans le cadre d'une utilisation interactive permettant l'exploration ou la visualisation des données générées.

Les finalités de l'application sont multiples : générer des séries temporelles réalistes respectant la structure et les comportements observés dans les données réelles ; assurer l'**anonymisation complète** des données conformément aux exigences légales et éthiques comme le **RGPD**, proposer une interface utilisateur simple et intuitive accessible même à des utilisateurs non spécialistes, permettre à l'utilisateur de configurer certains paramètres du processus de génération tels que le choix du modèle de génération, le niveau de fidélité attendu, le nombre de données synthétiques à produire ainsi que d'autres critères spécifiques au **jeu de données** et tout en mettant en place des **outils d'évaluation** afin de mesurer la qualité et la pertinence des données synthétiques créées.

Au-delà de ces fonctionnalités de base, l'application a été conçue pour être **évolutive**. Elle dispose d'une **architecture modulaire** conçue pour étendre à l'avenir les possibilités de **recherche** et d'**expérimentation**. Cette approche d'évolution prévoit l'ajout progressif de nouveaux algorithmes pour la génération de données synthétiques, à comparer différentes approches ou encore à explorer de nouvelles méthodes d'anonymisation adaptées à des jeux de données variés.

Par conséquent, l'application ne se limite pas à son usage immédiat mais constitue également un **outil de développement continu** pour la recherche en intelligence artificielle générative et en traitement de données éducatives.

2.1.2 Conception de l'interface utilisateur

Pour la conception de notre application, nous avons établi un plan en deux phases essentielles. Nous avons commencé par une étude des besoins utilisateurs. Ensuite, nous avons conçu et réalisé des maquettes afin de visualiser l'interface et valider notre choix.

L'interface graphique contient un menu structuré en trois sections principales :

- **Source** : Importer et explorer les données réelles.
- **Model** : Charger, configurer et entraîner un modèle génératif (type CTGAN).
- **Target** : Générer, analyser, sauvegarder et visualiser les données synthétiques.

* Source

Le menu **Source** est la première section de l'application. Cette dernière constitue l'entrée du processus et offre à l'utilisateur une expérience fluide, progressive et intuitive. Elle permet l'importation et l'exploration des données réelles. Elle se compose de trois sous-parties : **Open file**, **Display** et **Inspect**.

Analyse des besoins

Nous avons mené une étude fonctionnelle pour définir les attentes des utilisateurs concernant cette section :

- Chargement du fichier de données (format JSON),
- Visualisation des données,
- Affichage des statistiques.

* Model

Le menu **Model** est la deuxième section de l'application. Cette section constitue le cœur du processus de génération des données synthétiques, car elle permet de charger ou de créer un modèle génératif, puis de l'entraîner ou de l'enregistrer. Elle se compose de trois sous-parties : *New*, *Build* et *Tools*.

Analyse des besoins

D'après une étude fonctionnelle, nous avons spécifié les besoins utilisateurs de cette section :

- Chargement ou création d'un nouveau modèle en choisissant le type souhaité (ex. CTGAN).
- Entraînement du modèle avec les données réelles.
- Réglage des paramètres du modèle (nombre d'époques, taille de lot, etc.).

* Target

Cette section intervient après la phase d'entraînement du modèle. Elle constitue la dernière étape de l'application, permettant à l'utilisateur de

générer, évaluer et sauvegarder les données synthétiques. Elle se compose de trois sous-parties : **Generate**, **Analysis** et **Save**.

Analyse des besoins

- Génération de données synthétiques à partir du modèle entraîné.
- Comparaison des statistiques des données réelles et synthétiques.
- Évaluation de la confidentialité et de la fidélité des données synthétiques.
- Sauvegarde et visualisation des résultats obtenus.

2.2.1 Prototypage de l'interface utilisateur

Avant de procéder au développement complet, nous avons conçu un prototype de l'interface graphique afin d'en valider la structure, l'ergonomie et la navigation. Ce prototypage s'est appuyé sur une analyse fonctionnelle des besoins utilisateurs, traduits ensuite en trois sections logiques correspondant au parcours d'utilisation de l'application : **Source**, **Model** et **Target**.

Source La première étape de l'interface permet à l'utilisateur d'importer et d'explorer un jeu de données réel, au format JSON. Trois sous-menus ont été définis :

- **Open file** : chargement du fichier source,
- **Display** : affichage des données sous forme de tableau, avec un filtre dynamique selon le verbe, l'acteur, l'objet, ou le volume de données affiché,
- **Inspect** : représentation graphique des principales statistiques de l'ensemble de données.

Cette section a été pensée pour offrir une première interaction fluide et intuitive avec les données sources.

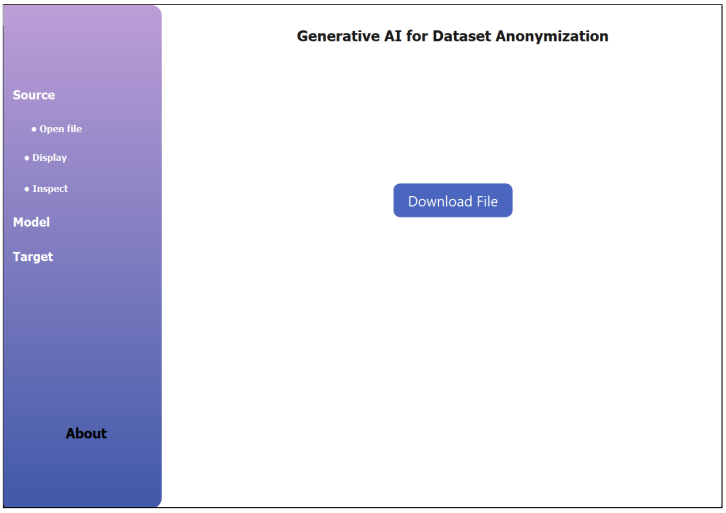


FIGURE 1 – Section *Source* : importation et exploration des données

Model Cette section représente le cœur fonctionnel de l'application. Elle permet de configurer et entraîner un modèle génératif, comme CTGAN. Elle est divisée en trois sous-parties :

— **New** : création ou chargement d'un modèle existant, avec sélection du type (par exemple Sessions ou Actions),

— **Build** : entraînement du modèle à partir des données importées, avec option de sauvegarde pour une réutilisation ultérieure,

— **Tools** : configuration des paramètres du modèle, tels que le nombre d'époques ou la taille du batch.

Ce module permet à l'utilisateur de gérer entièrement la phase de génération, tout en ajustant finement les réglages de l'entraînement.

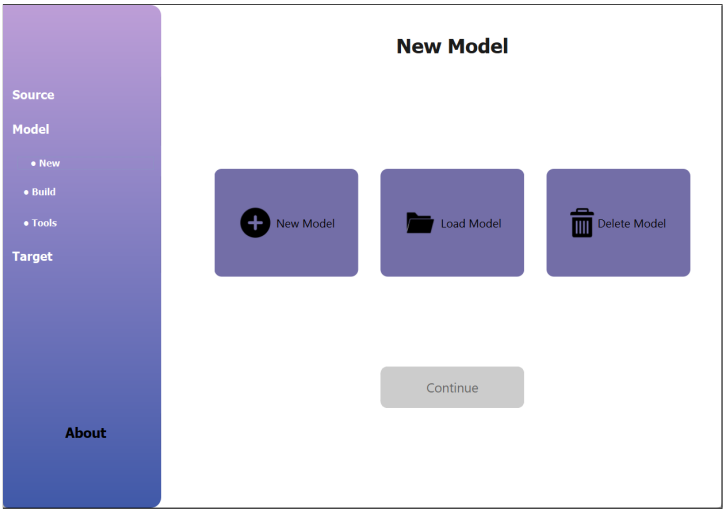


FIGURE 2 – Section *Model* : configuration et entraînement du modèle

Target Cette dernière étape permet de produire les données synthétiques, de les évaluer, et de sauvegarder les résultats. Elle comprend :

- **Generate** : génération des données à partir du modèle entraîné, avec choix du nombre d'actions et d'acteurs,
- **Analysis** : comparaison graphique des distributions entre données réelles et synthétiques. Deux sous-catégories sont prévues :
  - *Confidentiality* : évaluation de la confidentialité des données générées,
  - *Fidelity* : évaluation de leur fidélité statistique,
- **Save** : permet d'enregistrer les données synthétiques générées et d'afficher les résultats sauvegardés.

Cette section assure la dernière phase du processus, avec des indicateurs concrets sur la qualité des données synthétiques produites.

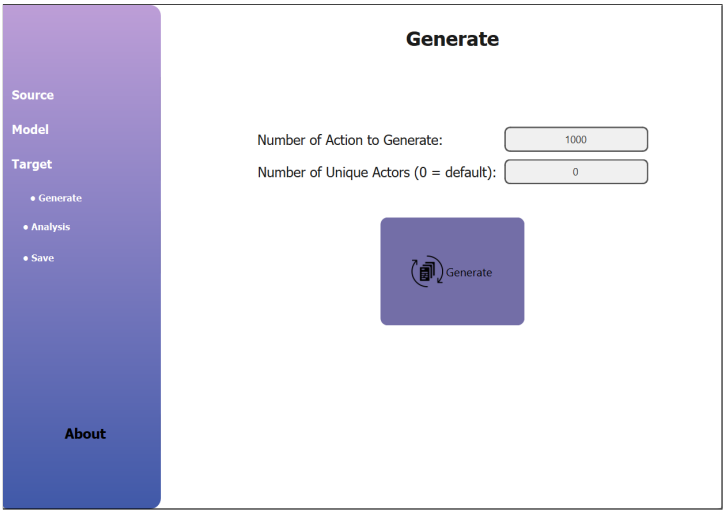


FIGURE 3 – Section *Target* : génération et évaluation des données synthétiques

2.1.3 Architecture logicielle

Pour ce projet, nous avons opté pour une architecture basée sur le modèle **MVC** (Modèle–Vue–Contrôleur). Ce **pattern** est couramment utilisé dans le développement d'applications afin de séparer les responsabilités entre la **gestion des données**, l'**interface utilisateur** et la **logique de contrôle**, facilitant ainsi la **maintenance** et l'**évolutivité** du projet.

L'architecture se décompose de la manière suivante :

Modèle (Model) : Le modèle est chargé de centraliser l'ensemble de la **logique métier** de l'application. Il est responsable du **traitement des données**, de la génération des séries temporelles synthétiques à partir des modèles d'intelligence artificielle **CTGAN**, ainsi que de l'**évaluation** de la qualité des jeux de données produits à travers différentes **métriques** (**TVC**, **MSE**, **KL Divergence**, etc).

Vue (View) : La vue représente l'interface graphique avec laquelle l'utilisateur interagit. Elle permet à l'utilisateur de configurer les paramètres de génération (modèle choisi, nombre de séries à produire, etc) et d'afficher les **résultats des évaluations** de manière claire et interactive.

Contrôleur (Controller) : Le contrôleur assure la **communication** entre l'utilisateur (via l'interface) et la **logique interne** de l'application. Il gère les **interactions** (par exemple, lorsqu'un utilisateur soumet une demande de génération), orchestre les appels aux modules du modèle, récupère

les **résultats** et actualise la vue en fonction des actions réalisées.

L'organisation en **MVC** apporte à l'application une structure **extensible**. Elle permet d'ajouter facilement de nouvelles **fonctionnalités** comme l'intégration future de nouveaux **modèles génératifs** ou de **méthodes d'évaluation**, sans remettre en cause l'ensemble de l'architecture existante.

2.2 Implémentation du modèle

2.2.1 Préparation des données

Le prétraitement des données est une opération essentielle qui garantit la qualité des données utilisées lors de l'entrainement d'un modèle d'apprentissage automatique. Nous avons utilisé une méthode qui prend deux paramètres : un DataFrame qui contient les données brutes et un paramètre qui précise le type de traitement 'Sessions' ou 'Actions'.

Elle permet de faire le nettoyage et retourne un DataFrame propre et structuré, prêt à être utilisé pour entrainer un modèle génératif (CTGAN).

a. Simplification des colonnes principales :
Cette étape est exécutée en premier qui sert à simplifier les données. Elle extrait les valeurs essentielles des colonnes contenant 'Actor', 'Verb' et 'Object'. Ainsi de normaliser les informations textuelles et de les rendre utilisables par d'autres traitements.

b. Validation et conversion de la colonne temporelle :
Cette étape consiste à vérifier si la colonne 'timestamp' est présente. Elle est obligatoire pour les analyses temporelles. Son contenu est converti au format datetime.

c. Tri des données :
Les données sont triées par 'Actor' ensuite par ordre chronologique, cela permet d'identifier les sessions ainsi que de calculer la durée des évènements.

d. Calcul de la durée des actions :
Cette étape permet d'ajouter une nouvelle colonne appelée 'Duration'. Cette dernière contient la durée de chaque action qui est calculé à partir de l'écart temporel entre deux événements consécutifs effectués par le même acteur. Si la durée calculée entre les deux actions est moins de 300 secondes (5minutes) alors cette durée est conservée. Sinon une durée fixe de 60 secondes est attribuée.

e. Gestion du mode Session :
Lorsque l'utilisateur choisit le mode 'Sessions', cela permet de regrouper les

actions d'un même utilisateur en sessions.

f. Sélection et formatage final :
Une fois que toute les étapes précédentes sont réalisées, une sélection et un formatage final est réalisé.
La sélection des colonnes est comme suit :
- En mode 'Actions' : 'timestamp', 'Duration', 'Actor', 'Verb', 'Object'.
- En mode 'Sessions' : 'timestamp', 'Duration', 'Actor', 'Verb', 'Object', 'session-id'.
Ainsi que la colonne 'timestamp' est convertie en chaîne de caractères.

2.2.2 Modélisation de la génération

La modélisation de la génération est une étape cruciale dans ce projet car c'est elle qui permet de créer et d'entrainer des modèles qui génèrent des données synthétiques.

A- Création d'un modèle génératif :
Le but de cette étape est de créer une instance du modèle génératif. Elle contient plusieurs phases.

Détection des Métadonnées :
Cette phase consiste à détecter les métadonnées à partir du DataFrame prétraité. Cela permet de configurer le modèle.

- Mise à jour des colonnes Catégorielles :
Les colonnes seront mises à jour pour qu'elles puissent être traitées comme des variables catégorielles dans un ordre spécifique, ce qui permet au modèle de comprendre les relations entre les différentes catégories.

- Initialisation du Modèle :
A partir des paramètres fournis, une instance CTGANSynthesizer est créée.

B- Entrainement du modèle :
Dans cette étape le modèle apprend à capturer les relations dans les données. Pendant l'entrainement, les données d'entrée sont utilisées pour ajuster les poids du réseau générateur ce qui permet au modèle d'apprendre à générer des données synthétiques plus fiables.

2.2.3 Modélisation de la sortie

Une fois le modèle **CTGAN** entraîné, les données tabulaires synthétiques doivent être converties en traces d'apprentissage normalisées (inspirées de xAPI) pour devenir réellement exploitables. Cette phase de *post-génération* applique plusieurs traitements successifs :

1. Structuration en actions pédagogiques Chaque ligne issue du générateur est transformée en une action complète :

- un identifiant d'action ;
- un horodatage normalisé (ISO8601) ;
- un verbe décrivant l'activité (*completed*, *viewed*, etc.) ;
- un acteur anonymisé ;
- un objet pédagogique ciblé ;
- la durée d'interaction (attribuée lors du post-traitement).

Ainsi, le jeu de données synthétiques adopte la même granularité que des traces xAPI réelles.

2. Reconstitution des sessions Si la variable session_id est présente, les actions sont regroupées afin de reconstruire des séquences temporelles cohérentes : une même session conserve toujours le même acteur et un fil chronologique crédible.

> ***Dépendance au mode d'entraînement.***
> - Mode session *: le générateur produit déjà des sessions complètes ; le post-traitement se limite donc au formatage.*
> - Mode action *: seules des actions isolées sont produites ; le post-traitement recrée alors des sessions artificielles (regroupement temporel ou attribution aléatoire).*

3. Anonymisation contrôlée des acteurs

- **Cas par défaut**
  - *Niveau session* : chaque session reçoit un acteur unique, pseudonymisé et constant.
  - *Niveau action* : le générateur choisit librement le nombre d'acteurs et maintient la cohérence des identifiants d'une action à l'autre.
- **Cas contraint** : l'utilisateur peut imposer un nombre maximal d'acteurs uniques ; un ensemble fixe d'identifiants est alors tiré au hasard, puis ré-employé de façon aléatoire.
  - *Avantage* : aucune identité réelle n'est révélée et la cardinalité du champ `Actor` reste limitée.
  - *Limite* : la ré-attribution aléatoire aplatit la distribution des interactions par apprenant ; certaines corrélations acteur–chronologie–activité s'estompent, réduisant légèrement la fidélité statistique.

4. Sorties mises à disposition

- **Format session-centré** : liste de sessions contenant chacune la séquence ordonnée de leurs actions, idéal pour visualiser ou rejouer un parcours.
- **Format action-centré** : table plate de toutes les actions, prête pour des analyses statistiques rapides (comptages, histogrammes, agrégations).

2.2.4 Méthodes d'évaluation utilisées

L'évaluation des données synthétiques générées repose sur deux grandes dimensions complémentaires : la **confidentialité** (capacité à garantir la non-réidentification des individus) et la **fidélité** (ressemblance statistique et structurelle avec les données réelles).

1. Évaluation de la confidentialité

Trois métriques principales ont été mises en œuvre :

- **Cramer's V** : mesure l'association entre variables catégorielles dans les jeux de données réel et synthétique. Une valeur proche de 1 suggère une forte similarité, et donc un risque potentiel de divulgation.
- **DCR (Distortion of Categorical Representation)** : quantifie l'écart entre les distributions de variables catégorielles dans les données réelles et synthétiques. Elle est utilisée pour détecter un éventuel surapprentissage ou une reproduction excessive des profils réels.
- **pMSE (propensity Mean Squared Error)** : évalue la capacité d'un classifieur (Random Forest) à distinguer les données synthétiques des données réelles. Un pMSE proche de 0,25 indique que les deux jeux sont statistiquement indiscernables.

2. Évaluation de la fidélité

La fidélité des données est mesurée à travers plusieurs indicateurs, visant à évaluer la cohérence globale et locale des distributions :

- **KS Complement (Kolmogorov–Smirnov)** : évalue la distance maximale entre les fonctions de répartition cumulées (CDF) des colonnes réelles et synthétiques. Un score faible indique une bonne fidélité.
- **TV Complement (Total Variation)** : mesure la divergence totale entre les distributions. Cette métrique est complémentaire à KS pour les colonnes catégorielles.
- **Logique des séquences de verbes** : comparaison des séquences d'actions (paires de verbes consécutifs) entre acteurs dans les deux jeux. L'indice de Jaccard permet de quantifier la similarité entre les bigrammes les plus fréquents.

— **Matrices de transition de Markov** : permettent d'analyser les probabilités de transition entre types d'actions (verbes), dans une logique temporelle. La divergence L1 entre les matrices réelles et synthétiques indique la fidélité comportementale.
— **Analyse en composantes principales (PCA)** : utilisée pour projeter les données dans un espace réduit (2D) afin d'en évaluer visuellement la structure latente. L'analyse de la variance expliquée par les premières composantes permet de juger du pouvoir discriminant du modèle.

2.2.5 Validation et contrôle qualité des données synthétiques générées

Afin de vérifier que les données synthétiques générés sont utilisables, non sensibles et correspondent aux données réelles, nous avons utilisé une méthodologie de contrôle qui permet de réaliser une analyse exploratoire et tester la fidélité et la confidentialité des données.

Analyse exploratoire des données synthétiques

Dans cette étape on effectue une analyse statistique comparative entre les données réelles et synthétiques, dans le but de vérifier la cohérence des données générées en termes de distributions et de comportements, et cela avec :
- Les visualisations des distributions.
- L'analyse de l'activité utilisateur.
- L'analyse temporelle.

Evaluation de la confidentialité

La confidentialité est un critère très important dans la génération de données. Elle permet de vérifier à quel point les données sont anonymes, ce qui veut dire que les données synthétiques ne représentent pas les acteurs présents dans les données originales. Ce critère est vérifié avec :
- L'analyse de similarité entre utilisateurs.
- Analyse des cas rares ou uniques.
- Vérification de la distinction entre données réelles et synthétiques.

Fidélité aux Données originales

La fidélité des données est un critère crucial pour évaluer si les données générées gardent la structure statistique et les comportements présents dans les données originales.

3 Résultats

3.1 Analyse statistique des données

Dans cette phase d'analyse, plusieurs **statistiques descriptives** ont été calculées afin d'évaluer la **cohérence globale** entre les données synthétiques générées par l'application et les données réelles d'apprentissage. L'objectif principal est de s'assurer que les données générées conservent les **caractéristiques essentielles** du comportement des utilisateurs tout en garantissant un niveau suffisant de diversification et d'anonymisation.

Parmi les **indicateurs clés** examinés, on retrouve notamment la répartition du nombre d'événements par acteur, les durées moyennes associées à chaque verbe d'action, la fréquence d'utilisation des différents verbes ainsi que la distribution des objets pédagogiques. Ces éléments permettent d'observer les grandes tendances d'interaction des utilisateurs, leur **diversité comportementale** et l'étendue des ressources mobilisées au cours des parcours.

Le choix de ces indicateurs n'est pas arbitraire : chacun d'eux est représentatif d'un aspect fondamental des traces d'apprentissages. Par exemple, la distribution des verbes nous renseigne sur les types d'activités les plus fréquentes dans les deux jeux de données. L'analyse des durées permet quant à elle d'identifier des **comportements temporels typiques**, comme le fait de passer plus de temps sur certaines tâches que sur d'autres. Enfin, la distribution des événements par acteur met en évidence la **charge d'activité par utilisateur** ce qui est un bon indicateur du **réalisme de la simulation**.

Globalement, les comparaisons entre les statistiques issues des données réelles et celles des données synthétiques révèlent une bonne cohérence. On observe que les **formes des distributions** sont approximativement bien conservées. Les actions dominantes restent les mêmes, les répartitions des objets sont comparables. Les écarts mesurés entre les deux ensembles de données restent modérés, ce qui signifie que le modèle génératif parvient à restituer fidèlement les structures globales sans pour autant copier les données originales.

Cela montre que le générateur est capable de produire des jeux de données synthétiques à la fois crédibles et utiles notamment pour entraîner des modèles. En même temps, l'introduction de légères variations et l'absence de correspondance exacte avec les données d'origine contribuent à renforcer la **confidentialité des utilisateurs**. Ce bon équilibre entre réalisme et anonymat constitue un atout majeur pour l'usage de ces données dans un cadre sécurisé. Toutefois, il est important de souligner que la **qualité des**

données synthétiques dépend fortement des paramètres choisis lors de l'entraînement du **modèle génératif** [5]. Un mauvais réglage peut conduire à une perte de fidélité ou au contraire à un **surapprentissage** risquant de compromettre l'anonymat.

3.2 Analyse de la confidentialité des données

Dans cette phase nous allons analyser la confidentialité des données synthétiques par rapport aux données réelles et cela par l'utilisation de trois méthodes :
- **Cramer's V**
- **DCR**
- **pMSE**

Analyse de la dépendance statistique -Cramer's V

La moyenne des dépendances des acteurs, verbes et objets entre les données générées et réelles est généralement faible ou les distributions sont indépendantes. De plus que leur répartition est différente. Ce qui est essentiel pour garantir la confidentialité. Cela signifie que le modèle n'a pas reproduit directement les structures statistiques des données réelles.

Analyse DCR (Distortion of Categorical Representation)

Dans ce cas on a remarqué en général, les résultats sont moyens et varient d'un cas à l'autre. Par exemple, pour les acteurs les résultats sont moyens. Cela signifie qu'au moins un enregistrement des données synthétiques qui est presque identique à celui des données réelles. D'autre part, pour les verbes et les objets les résultats sont bons, ce qui signifie que les données générées ne sont pas proches des données réelles.

Analyse pMSE (propensity Mean Squared Error)

La moyenne des scores de pMSE montre que les données synthétiques sont trop différentes des données réelles. Cela indique un bon niveau en terme de confidentialité.

3.3 Analyse de la fidélité des données

Dans cette étape, une analyse a été appliquée sur les données générées pour vérifier si elles respectent le critère de fidélité par rapport aux données réelles. Nous avons utilisé plusieurs méthodes :
- **KS Complement (Kolmogorov–Smirnov)**
- **Logique des séquences de verbes**
- **Matrices de transition de Markov**

- Analyse en composantes principales (PCA)

Analyse KS Complement (Kolmogorov–Smirnov)

Les résultats de cette analyse indiquent qu'il y a une divergence entre les distributions générées et les distributions réelles, ce qui signifie qu'elles ne sont pas bien reproduites et ne suivent pas les mêmes modèles que les données réelles, ce qui reflète une faible fidélité.

analyse de la Logique des séquences de verbes

Dans cette analyse les résultats présents dans les données réelles sont aussi présents dans les données synthétiques et sont relativement proches. Cela se traduit par un score élevé de l'indice de Jaccard ce qui indique la bonne cohérence des séquences courantes des données.

analyse avec la matrices de transition de Markov

Cette méthode nous donne le score de divergence entre les matrices de transitions des données réelles et synthétiques. Dans notre cas ce score est généralement très bon, ce qui confirme une bonne fidélité du comportement des séquences dans les données.

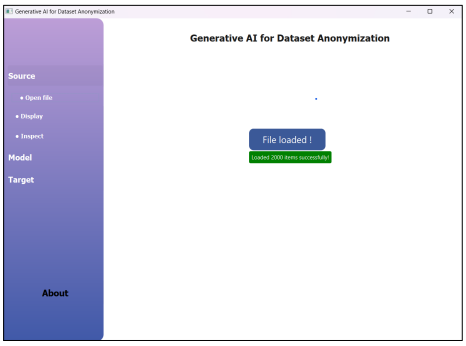
Analyse en composantes principales (PCA)

D'après les résultats des tests appliqués avec cette méthode. Il apparait que les données générées couvrent généralement les mêmes espaces que les données réelles. Ce critère confirme la bonne fidélité de ces données.

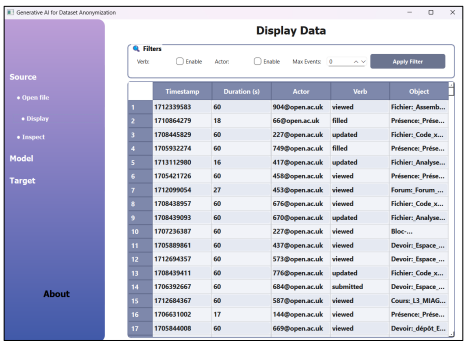
3.4 Conclusion

D'après les tests réalisés, il est difficile d'obtenir un résultat qui soit à la fois extrêmement réaliste tout en préservant la confidentialité et la fidélité des données. Cependant, il apparait que la génération des sessions donne de meilleurs résultats en termes de fidélité et de confidentialité par rapport à la génération des actions.

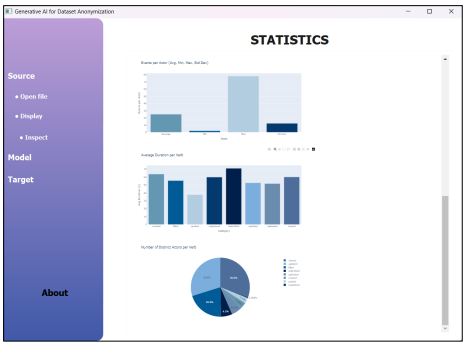
3.5 Cas d'usage de l'application



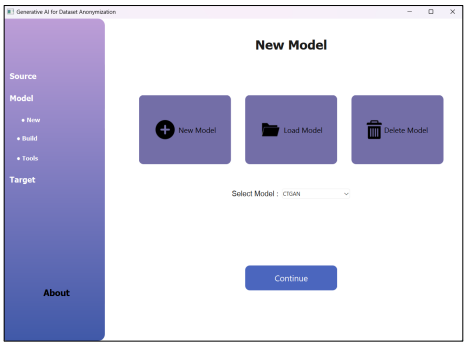
(a) Chargement du dataset



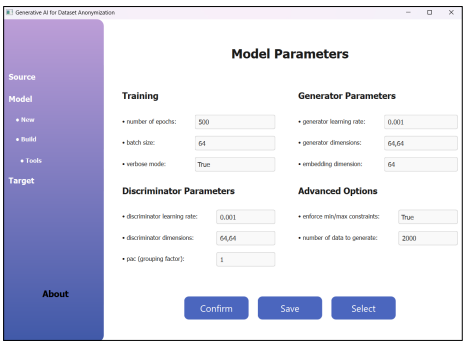
(b) Affichage du dataset



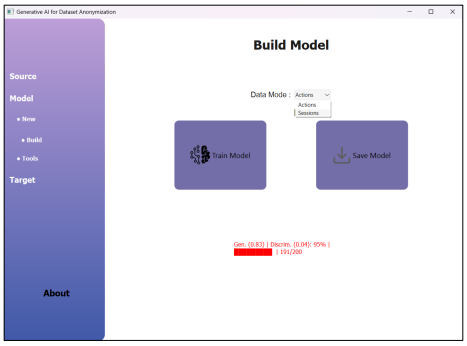
(c) Statistiques globales



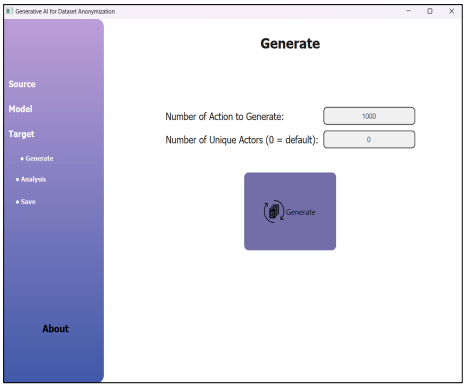
(d) Création d'un nouveau modèle



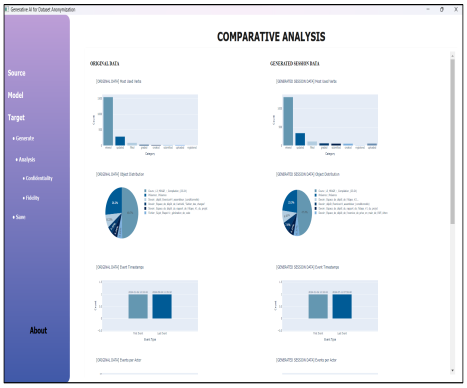
(e) Paramétrage du modèle génératif



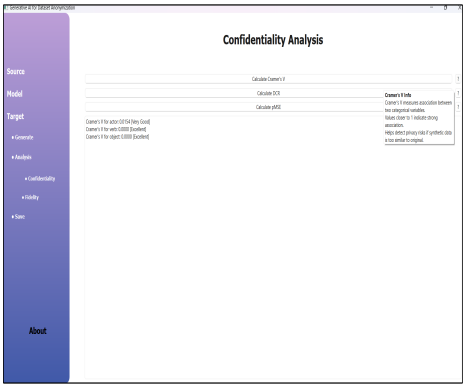
(f) Entraînement du modèle



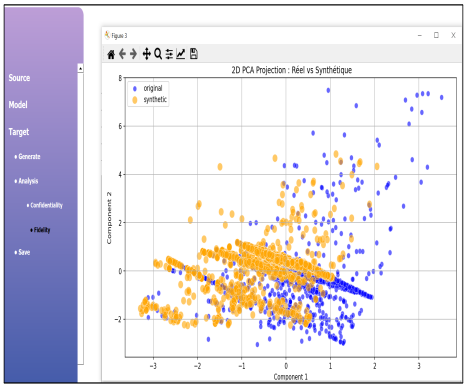
(a) Configuration des données à générer



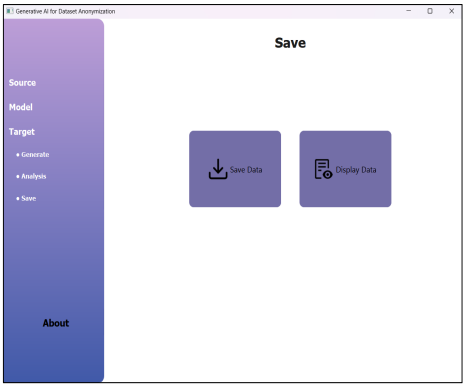
(b) Comparaison généré vs réel



(c) Évaluation de la confidentialité



(d) Évaluation de la fidélité **PCA**



(e) Sauvegarde et affichage des données synthétiques

FIGURE 5 – Cas d'usage de l'application (2/2)

4 Conclusion

Ce travail s'inscrit dans les objectifs portés par la plateforme LOLA, qui vise à mettre à disposition des données synthétiques afin de permettre aux concepteurs d'algorithmes d'ajuster leurs modèles avant leur évaluation finale sur des données réelles qui ne peuvent pas être directement partagées pour des raisons de confidentialité. L'application développée dans ce cadre permet de générer automatiquement des traces d'apprentissage synthétiques à partir d'un modèle CTGAN entraîné au sein d'une interface ergonomique et accessible. Elle répond ainsi à des besoins concrets que ce soit pour l'expérimentation, l'analyse ou la mise au point de modèles sans jamais exposer de données personnelles.

En plus de la génération de données, l'outil intègre des modules d'analyse permettant d'évaluer la qualité des données produites. Ces évaluations couvrent à la fois des critères statistiques (distribution des actions, diversité des objets, cohérence des séquences) et des critères de confidentialité. Les résultats obtenus montrent une cohérence globale acceptable entre les données réelles et synthétiques avec des risques de ré-identification plutôt bien maîtrisés.

Cependant, il convient de souligner que l'entraînement du modèle efficace reste une étape délicate notamment en raison de la nécessité d'ajuster finement les hyperparamètres. Cette tâche peut s'avérer particulièrement complexe et coûteuse en temps lorsqu'on travaille avec de grands volumes de données. Il serait donc pertinent dans la suite du projet d'automatiser davantage le réglage des paramètres et d'optimiser les performances d'apprentissage. Il serait également intéressant d'explorer d'autres modèles génératifs, d'autant plus que l'architecture MVC de l'application facilite leur intégration et permet une comparaison aisée des résultats.

Références

[1] UNESCO.
« Guidance for generative AI in education and research », 2023.
Disponible en ligne : unesco.org

[2] Advanced Distributed Learning Initiative (ADL).
xAPI (Experience API), 2017.
Disponible en ligne : adlnet.gov

[3] Bidou, Olivier, Mignot-Charvillat, Alice, Namoun, Yacine, Verbanaz, Emmanuel, Ait Chabane, Rabah, Maanni, Abdelhalim, Taoussi, Omar.
Étude de la qualité des données synthétiques pour les séries temporelles.
Projet tutoré, Université de Lorraine, LORIA / LOLA, 2024.

[4] Xu, L., Skoularidou, M., Cuesta-Infante, A., et Veeramachaneni, K.
« Modeling Tabular Data using Conditional GAN », *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
Disponible en ligne : arxiv.org

[5] Sustainability Methods.
How to Create Synthetic Data with CTGAN.
Disponible en ligne : sustainability.org

Amy Steier, Lipika Ramaswamy, Andre Manoel, Alexa Haushalter.
Synthetic Data Privacy Metrics.
arXiv preprint, 2025.
Disponible en ligne : https://arxiv.org/abs/2501.03941