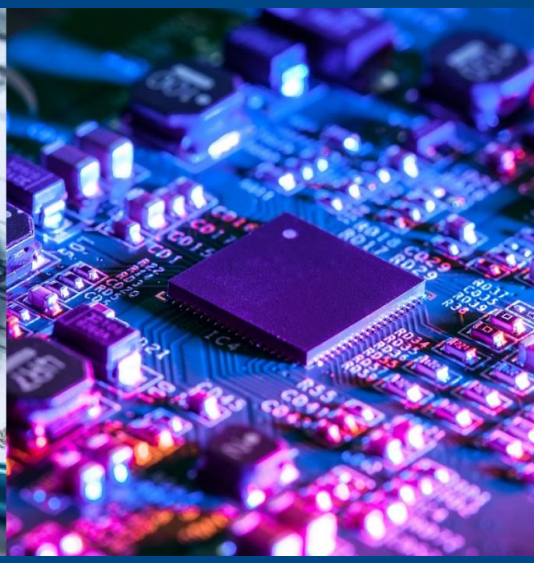
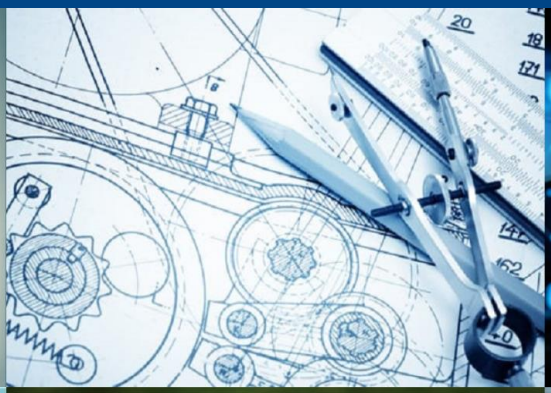
ISSN: 2582-7219

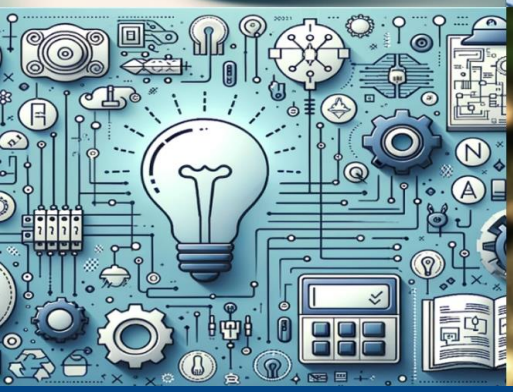




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206 **Volume 9, Issue 3, March 2026**

© 2026 IJMRSET | Volume 9, Issue 3, March 2026| DOI:10.15680/IJMRSET.2026.0903091

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

LLM-Based Synthetic Dataset Generation System

Dr. A. Jayalakshmi[1], Ms. M Iswarya[2]

Associate Professor, PG & Research Department of Information Technology, Sri Ramakrishna College of Arts & Science, Coimbatore, Tamil Nadu, India[1]

PG & Research Department of Information Technology, Sri Ramakrishna College of Arts & Science, Coimbatore, Tamil Nadu, India[2]

ABSTRACT: The rapid growth of artificial intelligence and machine learning has created a strong demand for large, high-quality, and diverse datasets for model training and evaluation. However, real-world data collection is often restricted by privacy regulations, limited accessibility, and high acquisition costs, especially in sensitive domains such as healthcare and finance. To address these challenges, this work presents a LLM-based synthetic dataset generation system that analyses a small seed dataset and produces large, logically consistent, and privacy-preserving synthetic data suitable for downstream machine learning tasks. The system integrates data preprocessing, feature extraction, and a locally deployed large language model to learn structural patterns, relationships, and constraints from the input data and generate noise-free synthetic records that follow the same schema. Experimental results demonstrate that the generated datasets preserve key statistical properties and logical dependencies of the original data while mitigating privacy risks, thereby improving data availability for research and development.

KEYWORDS: Large Language Models (LLMs), Synthetic dataset generation, Machine learning training data, Pattern analysis, feature extraction.

I. INTRODUCTION

Modern machine learning models, particularly deep learning architectures, rely on large-scale labelled datasets to learn complex patterns and achieve high predictive performance [1] [3]. In real-world environments, organisations often possess only limited or highly sensitive data due to legal, ethical, and operational constraints, which directly affect the quality and robustness of AI systems [2]. Traditional synthetic data generation techniques based on statistical sampling, rule-based approaches, or generative models such as GANs and VAEs frequently require substantial training data, extensive configuration, and high computational resources, and may still introduce noise and inconsistencies[1] [4]. With the advent of large language models (LLMs), it has become possible to understand not only statistical distributions but also contextual and logical relationships within structured datasets, enabling more realistic synthetic data generation from comparatively small samples [5][6]. This project proposes an LLM-based synthetic dataset generation framework that operates locally, ensuring privacy, and focuses on producing high-quality, logically valid synthetic records that mimic real-world data without directly exposing sensitive information [7].

II. LITERATURE REVIEW

Early approaches to synthetic data generation relied on classical statistical modelling, where parameters such as mean, variance, and probability distributions were estimated and used to sample artificial data points[3][4]. Although these methods can reproduce marginal distributions, they often fail to capture complex multivariate dependencies and higher-order relationships present in real datasets [1]. Rule-based synthetic data generators improved structural control by allowing domain experts to encode constraints and business rules, but this required significant manual effort and did not scale well to high-dimensional or rapidly changing domains [2]. With the emergence of deep generative models, particularly Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), it became possible to model complex data distributions and generate realistic samples for images, text, and tabular data [1][4]. However, these models typically demand large training datasets, careful hyperparameter tuning, and substantial GPU resources, and they may still produce mode collapse, noisy samples, or logically inconsistent records [1]. More recently, transformer-based large language models such as BERT and GPT have demonstrated strong capabilities in understanding structure and context across a wide range of data modalities, inspiring the use of LLMs for structured data synthesis, few-shot learning, and privacy-preserving data augmentation [5][6]. Existing cloud-based LLM

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

services, though powerful, often require uploading sensitive data to external servers, raising privacy and compliance concerns and motivating the exploration of local LLM runtimes for secure synthetic dataset generation [7].

III. SYSTEM ANALYSIS

The system analysis phase examines the limitations of current practices and motivates the design of a locally executed, LLM-driven synthetic data generator [2]. It evaluates how existing systems handle data scarcity, privacy, flexibility across formats, and the need for logically consistent outputs [6].

3.1 EXISTING SYSTEM

Current machine learning workflows typically depend on real-world datasets collected from sensors, transaction logs, medical records, or user interactions, which are often incomplete, imbalanced, or sensitive [3]. Traditional synthetic data tools use statistical sampling, rule-based logic, or generative models like GANs and VAEs to augment datasets, but they generally require large training sets, domain-specific configuration, and significant computational effort [1][4]. Many of these tools focus on matching statistical similarities rather than understanding deeper logical constraints, leading to synthetic records that may violate domain rules or produce unrealistic combinations of attribute values [4]. Moreover, a large proportion of advanced generative services are cloud-hosted, forcing organizations to upload confidential data to external providers and creating potential privacy, security, and compliance risks [2][7]. These limitations highlight the need for an intelligent, flexible, and locally executable system that can generate realistic synthetic data from small seed datasets while preserving privacy and logical integrity [6].

3.2 PROPOSED SYSTEM

The proposed system introduces a locally deployed LLM-based synthetic dataset generator that takes a small structured seed dataset (e.g., CSV) as input and produces a larger synthetic dataset with the same schema and logical relationships [6][7]. The pipeline begins with data cleaning and preprocessing to remove duplicates, handle missing values, and correct inconsistencies, ensuring that the model learns only valid patterns [8][9]. The system then performs schema and constraint analysis to infer column types, value ranges, dependencies, and domain rules, which are passed as structured context to the LLM [5]. The local LLM is responsible for learning patterns and generating new records that respect the discovered schema and constraints, producing noise-free, logically consistent samples at scale [6]. A validation module subsequently checks the generated records against the original schema and constraints, discarding invalid rows and guaranteeing structural correctness [4]. Running the entire pipeline on a local runtime ensures that sensitive data never leaves the organisation's environment while still benefiting from the expressive power of modern LLMs [7].

IV. METHODOLOGY

4.1 DATA COLLECTION

In the proposed framework, data collection focuses on acquiring a representative but possibly small seed dataset from the target domain [3]. For tabular tasks, this may consist of CSV files containing attributes such as demographic information, transaction details, or system logs, obtained from internal databases under strict privacy controls [2][8]. In audio-related applications, speech signals or acoustic recordings are captured from microphones or publicly available corpora while adhering to ethical guidelines and anonymization procedures [9]. The collected data is stored securely within the local environment to prevent unauthorized access and prepare it for preprocessing and feature extraction [7].

4.2 DATA PREPROCESSING

Data preprocessing ensures that both tabular and audio data are clean, consistent, and suitable for pattern learning by the LLM and feature extraction modules [8][9]. For structured data, preprocessing includes removing duplicate rows, handling missing values using appropriate imputation strategies, standardizing formats (e.g., dates, categorical labels), and detecting outliers that may distort the learned distributions [8]. For raw audio, preprocessing steps such as resampling, silence removal, normalization of amplitude, and segmentation into fixed-length frames are performed to obtain uniform input segments for feature extraction [9]. This stage guarantees that subsequent modules operate on high-quality data, improving the realism and reliability of the generated synthetic dataset [4].

© 2026 IJMRSET | Volume 9, Issue 3, March 2026| DOI:10.15680/IJMRSET.2026.0903091

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

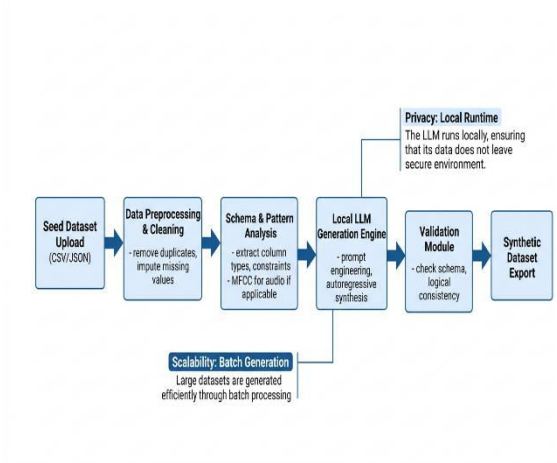
International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

4.3 FEATURE EXTRACTION USING MFCC

For speech or audio-based datasets, Mel-Frequency Cepstral Coefficients (MFCC) are employed to convert raw waveforms into compact, perceptually meaningful feature representations [9]. MFCC extraction typically involves framing the signal, applying a window function, computing the short-time Fourier transform, mapping the power spectrum onto the Mel scale filterbank, taking the logarithm of filter energies, and finally applying a discrete cosine transform to obtain cepstral coefficients [1]. These coefficients capture the spectral envelope of the speech signal, which is closely related to human auditory perception and is widely used in tasks such as speech recognition, speaker identification, and emotion analysis [3]. The resulting MFCC feature vectors serve either as inputs to downstream models (e.g., classifiers) or as structured numeric data that can be further synthesized by the LLM-based framework to create additional audio-feature records [6].

The below descries the architecture of the LLM-based synthetic dataset generation system.



V. LLM-BASED SYNTHETIC DATASET GENERATION SYSTEM ARCHITECTURE

The proposed system architecture is modular and consists of input, preprocessing, analysis, LLM generation, validation, and output components connected via well-defined interfaces [7]. The user interface or API layer allows users to upload seed datasets, configure generation parameters (e.g., number of records, output format), and trigger the generation process [8]. The preprocessing and schema analysis module inspects the dataset, derives metadata such as column types and constraints, and prepares a structured prompt or context that is passed to the local LLM runtime [5][9]. The LLM generation engine receives this context and iteratively produces synthetic records that adhere to the specified schema, which are then forwarded to the validation module [6]. The validation and export layer checks structural correctness, removes invalid rows, and formats the final synthetic dataset into user-selected formats such as CSV or JSON for downstream use [4][8].

5.1 MODEL ARCHITECTURE AND TRAINING

The LLM-based generator leverages a transformer-style architecture with self-attention mechanisms to model complex dependencies between attributes within each record and across records in the dataset [5]. Instead of training an LLM entirely from scratch, the system relies on a locally hosted, pre-trained model that is adapted to structured data synthesis using prompt engineering, few-shot examples, or lightweight fine-tuning on representative samples [6][7]. During adaptation, the model is exposed to schema descriptions, valid record examples, and explicit constraint instructions, enabling it to internalize the allowed value ranges, categorical distributions, and logical rules [5]. The training or adaptation process runs entirely within the local environment using available CPU or GPU resources as specified in the hardware requirements, ensuring that no sensitive data is transmitted externally [7]. Once configured, the model can be prompted repeatedly to generate arbitrary numbers of synthetic records while preserving the patterns learned from the seed data [6].

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VI. IMPLEMENTATION AND TOOLS

The system is implemented primarily in Python, leveraging its rich ecosystem for data processing, model integration, and web development [8]. Pandas and NumPy are used to load, clean, transform, and validate tabular data, while audio-related tasks such as MFCC extraction can be implemented using libraries like Librosa within the same environment [9]. A locally deployed LLM runtime (e.g., using tools such as Ollama or similar frameworks) manages loading and executing the language model without external API calls, providing secure inference within the local machine [7]. Backend services are developed using lightweight web frameworks such as Flask or FastAPI, exposing endpoints for dataset upload, generation configuration, and result download [8]. A simple web-based interface or dashboard can be constructed with modern frontend technologies to allow non-expert users to interact with the system, monitor progress, and review generation history [2].

VII. RESULTS AND DISCUSSION

The implemented system was evaluated by generating synthetic datasets from small seed datasets and comparing structural and logical properties between original and synthetic data [4]. The validation module confirmed that the generated records conformed to the original schema and respected predefined constraints such as data types, value ranges, and key relationships [5]. Qualitative inspection showed that the synthetic data exhibited realistic value combinations without directly replicating any individual original record, indicating a reasonable privacy-preserving behavior [2][6]. In terms of scalability, the system successfully expanded small datasets into significantly larger ones while maintaining generation times acceptable for practical use on standard hardware [7]. Compared with traditional rule-based or purely statistical methods, the LLM-based approach provided greater flexibility in capturing contextual relationships and produced fewer logically inconsistent records, though its quality still depended on the clarity of prompts and the representativeness of the seed data [1][3].

VIII. CONCLUSION

This work presented a locally deployed LLM-based synthetic dataset generation system designed to address data scarcity and privacy concerns in machine learning applications [2][6]. By combining rigorous data preprocessing, schema and constraint analysis, MFCC-based feature extraction for audio, and an LLM-driven generator, the system is able to produce large, noise-free, and logically consistent synthetic datasets from limited seed data [7][9]. Operating entirely within a local runtime environment ensures that sensitive information remains within organisational boundaries while still benefiting from the advanced pattern-learning capabilities of transformer-based models [5]. Experimental evaluation indicates that the generated datasets preserve important structural and statistical properties of the original data and are suitable for training and testing machine learning models [4]. The modular architecture also provides a foundation for future extensions to additional data modalities and more sophisticated evaluation and control mechanisms [1].

REFERENCES

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville,Deep Learning, MIT Press, 2016.
2. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 4th Edition, Pearson Education.
3. Tom M. Mitchell, Machine Learning, McGraw-Hill Education.
4. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer.
5. Jacob Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Proceedings of NAACL-HLT, 2019.
6. OpenAI, “Language Models are Few-Shot Learners,” Advances in Neural Information Processing Systems.
7. Ollama Official Documentation, https://ollama.ai.
8. Python Software Foundation, Python Programming Language Documentation, https://www.python.org.
9. Pandas Documentation, https://pandas.pydata.org.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com