International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

DiffusionClaims – PHI-Safe Synthetic Claims for Robust Anomaly Detection

Jimmy Joseph

Solutions Engineer Advisor Sr., United states

ABSTRACT: Healthcare claims suffer from data being too rich to reveal real patterns and anomalies, but privacy legislations like HIPAA prevent access to actual patient records. In this paper, we introduce DiffusionClaims, a new paradigm using diffusion models to create realistic synthetic healthcare claims that satisfactorily maintain statistical patterns of existing data while avoiding exposure of any protected health information (PHI). Diffusion models, which are classically known from image generation, are more stable during training and achieve better mode coverage than generative adversarial networks (GANs). We utilize these models for tabular claims data, where we first encode mixed categorical and numeric features into a continuous latent space for diffusion-based synthesis. The generated claims are then used to build a strong anomaly detection pipeline for fraud.

We evaluate DiffusionClaims against competitive GAN-based models and an existing rule-based simulator, showing that the proposed diffusion-generated claims not only match realistic data (feature distributions/correlations) but also are useful for downstream fraud detection tasks. We additionally assess differential privacy risks using membership inference and distance-to-record metrics, concluding that DiffusionClaims generates synthetic data with low re-identification risk, sufficient to support HIPAA compliance. Experimental evaluation with a public insurance claims dataset and a universal gas fraud dataset confirms that models trained on synthetic (e.g., injected) data are able to effectively identify anomalies, performing almost as well as those trained with real datasets.

We also present industry-standard quality metrics for synthetic data and privacy (fidelity, utility, privacy) and demonstrate that DiffusionClaims strikes the fidelity–utility/privacy trade-off to safeguard patient privacy. DiffusionClaims allows the sharing and analysis of realistic claims data without revealing private records, offering new potential for cooperative fraud detection and rare-event modeling in healthcare.

KEYWORDS: Synthetic healthcare data, Diffusion models, HIPAA compliance, Insurance claims, Fraud detection, Anomaly detection, Privacy-preserving machine learning

I. INTRODUCTION

In health insurance claims, this type of approach is fundamental for fraud detection and anomaly discovery using data-driven methods. However, this is in many cases not possible as claims data are typically sensitive and highly confidential under patient privacy legislation such as the Health Insurance Portability and Accountability Act (HIPAA) in the US. The strict regulations and the sensitive nature of PHI/PII, which can also contain PHI, pose barriers to obtaining real claims datasets for machine learning studies. Even de-identified EHRs contain the risk of re-identification and are covered under privacy regulations, which limits the amount of data available for training and assessment of anomaly detection models. Synthetic data generation has recently gained attention as one common solution to this dilemma. Generated to have the same statistical properties as real data without disclosing anything about an individual, synthetic data has the potential to foster data sharing and algorithm learning while respecting privacy.

Recently, various generative modeling methods have been utilized to generate synthetic healthcare data such as EHR and claims. Early methods vary from naïve rule-based simulations to sophisticated deep generative models. For instance, custom domain simulators such as Synthea generate synthetic patient records according to expert-mandated criteria and likelihood functions of care. These conventional procedures guarantee believable data, and since PHI disclosures are implicit, they may not have realistic complexity or diversity. Newer approaches use learning to model data distributions from real datasets. Generative Adversarial Networks (GANs) have been studied for synthesizing patient records (e.g., medGAN for EHRs), which can grasp complex joint distributions. However, GANs are notoriously prone to mode collapse (which means they fail to capture the entire diversity of data), and in general can be

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

hard to train. These constraints also suggest that GAN-sampled data may miss rare, important patterns or glitches even if it does produce samples at all.

Diffusion models have recently sparked a resurgence of attention as alternative generative models that do not rely on adversarial training, driven by diffusion-based image synthesis models whose fidelity and mode coverage have surpassed those of GAN-based ones. This brings us to our research question: is it possible to leverage diffusion models for creating high-quality synthetic tabular data, e.g., healthcare claims, that are HIPAA-compliant as well as useful for anomaly detection?

We propose **DiffusionClaims**, a framework for PHI-safe synthetic claims generation with diffusion models, and as an application we show how they can be used to detect anomalous (fraud) patterns. The method addresses the twin problems of privacy compliance and data utility in this area. By teaching a model to create realistic distributions of claims data through noise-based diffusion, we can generate synthetic claims that retain important statistical properties of real claims (distributions of charges, codes, provider behaviors) in the absence of any individual's claim information.

Synthetic Data That Minimizes the Risk of Re-identification: According to HIPAA, appropriately de-identified data with no personally identifiable information is not considered PHI, and it can thus be freely distributed and studied. This significantly reduces the friction of working together (e.g., for co-auditing model development or deployment) with experimental models, as synthetic claims can be substituted for real PHI-holding records for development and testing.

To validate the realism and utility of diffusion-generated claims, we incorporate them into a fraud detection pipeline. The detection of fraud in healthcare claims provides a classical anomaly problem since fraudulent events are rare (typically <5% of claims) and heterogeneous. Although efforts to detect fraud using machine learning (ML) have been made, ML models can face challenges in such an environment because of a lack of or few positive (fraudulent transaction) samples and possible bias in training data. Synthetic data has many advantages in this regard. First, it can supplement small real datasets or entirely act as a surrogate for real data, which means that there is no regulatory limitation on experimentation. Second, it permits the injection of fake patterns of fraud (such as upcoding) for creating labeled data to use for supervised learning or to evaluate unsupervised detection methods. By producing a large number of normal claims and a limited amount of fraudulent ones, it is possible to train effective anomaly detection models that are robust against variations. Crucially, none of this needs to be done on an actual patient or provider; it can all take place in a "safe sandbox."

Our contributions are as follows:

1. We advocate the use of denoising diffusion-based probabilistic models for synthesizing health insurance claims and discuss a process to tailor them to mixed-type tabular data in accordance with privacy constraints.
2. We conduct extensive comparisons with prior methods (including GAN-based generation and rule-based simulation) and show that DiffusionClaims achieves better data quality, diversity, and training stability.
3. We study the performance of automatically detecting aberrant claims by training a full anomaly detection pipeline on the synthetic claims, with data preprocessing to enhance model generalization.
4. We quantitatively evaluate the quality of synthetic data through industry-standard metrics along three dimensions: fidelity (how similar synthetic data is to real data distributions), utility (how useful synthetic data is for training models toward real-world tasks), and privacy (risk of leaking sensitive knowledge). We demonstrate that DiffusionClaims strikes an appropriate trade-off, achieving high model performance on downstream tasks (close to that of training with real data) and low leakage in privacy metrics (no exact record copy and low re-identification risk).

The rest of this paper is structured as follows. Under *Related Work*, we articulate previous systems for synthetic data generation in the medical domain and past studies on claims fraud detection. In *Methods*, we detail the structure of DiffusionClaims and its anomaly detection component. In the next section, we describe our experimental setting, datasets, and baselines. *Results* present both quantitative and qualitative data quality and anomaly detection performance. We then discuss, in the *Privacy Compliance* section, practical deployment issues and limitations of our approach. We finally close by discussing the next research directions in privacy-preserving anomaly detection.

II. RELATED WORK

Synthetic Data Generation in Healthcare

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

Rule-Based Simulation One time-honoured topic in synthetic healthcare data generation is rule-based simulation based on domain knowledge [35]. Tools such as Synthea generate the full medical histories of fictitious patients by generating, based on epidemiological statistics and clinical guidelines, diseases present, encounters made, and interventions undertaken. Simulators like Synthea and others (e.g., agent-based models) generate fully synthetic data that do not include real PHI, thereby naturally avoiding privacy concerns. They might generate synthetic claims or electronic health records that can serve as surrogates for real data in tests of software, the development of algorithms, or training exercises. Although rule-based synthetic data may be extremely realistic for ordinary scenarios, its drawback is that it is based on predefined rules and distributions. This might cause some loss of realism, particularly in the context of complex, rare, or emergency patterns that were not described by the experts. Additionally, when the true rules or parameter values that describe the synthetic data are not perfectly calibrated, the synthetic data may differ slightly from real-world data in a nuanced sense (e.g., lacking cost multimodality or having correlations between diagnoses and billing codes that are subtle). Nevertheless, simulation is a useful baseline due to its interpretability and guaranteed privacy—it "generates new data that mimics statistical properties" but does not come from any actual individual.

Statistical and Simple Generative Methods Below whole-scale rule-driven simulation, other simpler generative methods have been employed. These may consist of sampling values from estimated probability distributions for each field, or obtaining samples through resampling procedures. For instance, we can simulate synthetic claims by sampling claim data from the categorical frequency distributions (such as claim dates, procedure codes, and provider IDs) while also sampling simulated claim amounts from a skewed distribution (like Gamma or log-normal) to mimic real cost data. Tiya Vaj (2025) utilized the above approach by synthetically generating 5,000 claims with randomly assigned provider and patient IDs, random CPT procedure codes, and billed amounts sampled from a Gamma distribution. Then fraudulent cases were generated by randomly choosing 3 percent of the claims and increasing their amounts 3–8x to create outliers. This approach is simple to implement and can inject known a priori anomalies. But it treats each feature very independently (aside from primitive relationships by means of resampling or scaling), and may not be able to capture higher-order interactions in the data. Therefore, it may be insufficient to capture the joint distribution of characteristics that appear in claims (e.g., which procedure codes co-occur with certain diagnosis codes and provider specialties, etc.).

Deep Generative Models (GANs and VAEs) Deep learning opened the doors to powerful generative models that can learn data distributions from real examples. Both VAEs and GANs have been utilized in healthcare data. Variational AEs (VAEs; e.g., medVAE) map real records to a latent space before decoding, generating new records and minimizing a likelihood-based loss along with regularization that enforces smoothness of the latent distribution. GANs challenge a generator network to make data so close to real records that they can't be told apart. medGAN was one of the earliest works that used GANs to generate new patient records (binary EHR encounter data more specifically) after training on real patient vectors and showed synthetic samples replicated real patient feature distributions. Later varieties such as medWGAN and medBGAN introduced Wasserstein losses or better training approaches. GAN-based approaches have demonstrated potential in synthesizing clinical data and can even enhance model training on imbalanced data (e.g., generating additional minority class samples). For example, rare disease cases have been synthesized using GANs as extra training samples to train predictive models.

Despite achievements, GANs have distinct limitations when applied in healthcare. As emphasized above, mode collapse is what one would like to avoid: a GAN generator that just learns to produce only a small diversity of samples that fool the discriminator, failing to cover important modes in the true data distribution. At the claims level, mode collapse might correspond to not producing claims from some specialties or cases of codes that, though rare, are actually valid combinations. GANs also suffer from training instability—it is tougher to achieve equilibrium of generator and discriminator, sometimes resulting in failures to converge or generating nonsense. It is further complicated for tabular data, where the distribution between a mixture of discrete and continuous variables with non-trivial dependencies must be learned. Tabular GAN models (such as CTGAN, specialized in mixed data) somewhat address those problems through architectural changes but usually need careful tuning. Yet another issue is the generalizability of GANs: if not regularized in a suitable way, they may overfit and in fact memorize parts of the training dataset, leading to possible information leakage (about real patients) in these "synthetic" outputs. Of course, this is dangerous in the context of PHI—you need to be sure your synthetic data isn't just reproducing or too closely matching some particular real person's record.

Diffusion-Based Generative Models for Synthetic Data Diffusion probabilistic models have recently been proposed as practical alternatives to generative modeling. Diffusion models, as opposed to GANs, do not use a discriminator and

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

instead learn to denoise data from pure noise, calibrating the output of the model correctly by reversing a forward noising process. Key benefits of diffusion models include stable training (training is simply likelihood maximization, no adversarial game) and good mode coverage (the model is encouraged to cover the full data distribution due to noise perturbation in the process, so the risk of mode collapse should be lower). Diffusion models were first popularized in the image domain (Ho et al., 2020; DDPM), and have started to show power for other modalities such as text, time series, and tabular data. In tabular synthetic data, recent works (Kotelnikov et al., 2023; Kim et al., 2023) show that diffusion models (often augmented with autoencoders or other preprocessing) can achieve state-of-the-art performance compared to GANs and VAEs in synthesizing realistic tables with mixed data types. For instance, TabDDPM introduced a diffusion-based tabular generator and demonstrated that it outperformed previous GAN-based methods in modeling the desired distribution of datasets and computing machine learning utility with them. Furthermore, the diffusion-based data had strong privacy properties: Kotelnikov et al. note that TabDDPM is harder to connect samples with original data than SMOTE oversampling, as indicated by greater DCRs and lower membership attack success rates.

In the healthcare space, diffusion models have already proven their utility in some specific tasks like synthetic medical imaging (generating realistic X-rays or MRIs) and even EHR data synthesis. For instance, in ScoEHR (Naseer et al., 2023), an autoencoder was integrated with a continuous-time diffusion model to synthesize structured EHR data (patient records that have lab values, etc.). ScoEHR outperformed medGAN and GAN-variants in terms of several data utility metrics: it better maintains feature marginals and pairwise correlations, as well as generates more informative downstream predictions on the synthesized data. Importantly, a privacy analysis there didn't reveal evidence of membership leakage: an adversary attempting to determine if any particular patient record had been part of the training did little better than random guessing. These results highlight the utility of diffusion models in producing realistic, low-leakage synthetic health data. In this work, we extend this development trajectory in the context of insurance claims and their application for anomaly detection.

Privacy and Compliance Concerns An underlying reason for using synthetic data in healthcare is to comply with privacy regulations (HIPAA, GDPR). De-identifying real data by redacting direct identifiers and perturbing the quasi-identifiers is one direction, but it has already been demonstrated that re-identification attacks or membership inference can be launched even against de-identified real data. Synthetic data (if crafted well) provides a strong alternative: there is no one-to-one correspondence to real people; data that is not PHI, according to rules set out by the HIPAA Privacy Rule. As further detailed here, the construction of synthetic data without PHI is interpreted as a form of creating de-identified information and therefore permitted, such that resulting data can be disseminated for secondary purposes without patient consent. The one major caveat is that the synthesis has to be done "safely," i.e., no person must be reconstructable from the synthetic data. This is why very strong privacy measures derived from rigorous statistical metrics and testing (with techniques such as looking for exact actual record duplicates, nearest-neighbor tests, membership-inference tests) are a must in any synthetic-data approach. DiffusionClaims does this in its evaluation by including such checks to make sure that the synthetic claims are truly PHI-safe. We also observe that using additional mechanisms such as differential privacy on top of generative models can further provably bound privacy leakage (albeit at some cost of utility). Differentially private GANs and VAEs have been investigated; differential privacy for diffusion models is a frontier but conceptually doable. We focus on achieving non-memorization through model design in this work; incorporating formal privacy guarantees is the subject of future work.

III. ANOMALY DETECTION IN HEALTHCARE CLAIMS

Challenges in Fraud and Anomaly Detection Healthcare fraud manifests in many ways—billing for services not performed, upcoding (billing for more expensive services than those rendered), duplicate claims, etc. These fraudulent cases, from the data point of view, are anomalies, and they are abnormal instances as most valid claims look normal. One of the widely acknowledged challenges in anomaly detection in insurance is class imbalance and the adaptive nature of fraud (the "bad guys" change their playing rules to avoid being caught). A review by du Preez et al. (2025) notes that fraud detection methods have expanded from unsupervised approaches (outlier detection without labeled fraud examples), to supervised learning (when historical labeled fraud cases are available), to hybrid approaches that make use of both. Unsupervised options include clustering, autoencoders, and statistical outlier tests that seek to flag claims that are outliers. Supervised methods such as tree-based classifiers (random forests, XGBoost) and neural networks need labeled training data to perform well if there are enough instances of fraud. However, since only a small proportion (e.g., 1–5%) of claims are fraudulent in real-world datasets, supervised training can be biased and calls for handling of class imbalance by means of techniques such as SMOTE oversampling or cost-sensitive learning.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

Explainability is also emphasized: in a domain such as healthcare, you can't get away with just flagging an anomaly—auditors must be able to know why a claim was flagged (which features had and did not have exceptional values) in order to take appropriate action.

Synthetic Data for Anomaly Detection Synthetic data can support anomaly detection in various aspects. One quick, simple approach is to oversample the minority class—common methods such as SMOTE (Synthetic Minority Oversampling Technique) create new synthetic instances of positives by interpolating between real ones. It has been found that this can help alleviate the imbalance of the training set and facilitate classifier learning. Yet SMOTE's new points are, quite simply, linear combinations of real instances that could still be very close to real data (possibly causing a breach in privacy and sometimes creating unrealistic samples). With more sophisticated generative models (GANs, etc.), synthetic examples of fraud beyond mere oversampling can be created, and this can help models grasp a broader notion of fraud and not just interpolations over known samples. Another application is simulating scenario designs to evaluate anomaly detectors. For instance, a completely artificial dataset of claims with "ground truth" embedded fraudulent claims can be created, as we do in our experiments. That provides a way to do controlled evaluation of detection algorithms: because we know precisely which claims are fraudulent, we can measure how accurately the fraudulent claims are detected, without the uncertainty that comes with real-world data. Synthetic financial transaction datasets have been created for fraud studies (e.g., PaySim for mobile money), and synthetic healthcare claims have been proposed as a means to benchmark fraud detection systems without privacy concerns.

In reality, many healthcare payers employ a mixture of rules (expert-labeled flags) and machine learning models to identify claim anomalies. Machine learning can triage risk scores for claims that undergo human expert review. The use of AI for this purpose is increasing, and synthetic data can expedite development by enabling simple experimentation by data science teams. It also allows sharing examples of fraud patterns with other institutions (via synthetic data) that would otherwise be prohibited—an insurer could produce synthetic cases exemplifying a new fraud scheme and share them with others to assist in training detectors, but without releasing any actual real PHI. The synthetic data-based outlier detection standing in can be understood as proof of concept that this collaboration is possible.

Performance of anomaly detection is usually evaluated with metrics that are tailored to imbalanced data: Area Under the ROC Curve (AUC) and Average Precision (AP) (a.k.a. area under the Precision–Recall curve), together with variants like precision/recall at top-K and false positive rate at a given sensitivity. In our experiments, we work with AP as it is informative for heavily skewed data (it highlights the performance on the positive class). We also investigate the detection rate (at specific thresholds) and the burden of false alerts (too high a number of false positives will flood investigators). These measures help us compare models trained on real vs. synthetic data, and determine if such a simulated dataset is "good enough" for anomaly detection.

IV. METHODS

DiffusionClaims Model for Creating Synthetic Claims

At the center of DiffusionClaims is a generative model consisting of denoising diffusion probabilistic modeling. Our model takes fixed-length representations of insurance claims as input. A claim generally consists of several fields: for example, date of service, provider identifier, patient identifier, list of diagnosis codes and procedure codes (CPT/HCPCS), billed amount to the insurance policy holder as well as the paid amount by the policy holder, along with possibly derived features (patient age, doctor specialty). Such features are a mix of discrete and continuous variables, which presents a problem as diffusion models are originally computed on continuous vector spaces. To tackle this task, we follow an autoencoder-type approach employed by ScoEHR: first encode the original claim data into a continuous latent space which can be diffused over and decode the synthesized single record back to it.

Let x be a raw claim record whose data types are between structured and unstructured. We define an encoder $E(x)$ which maps from input to a vector z in $\{R\}^d = \{(x_1, x_2, \dots, x_d) | \, x_i \, in \, \{R\}\}$. The encoder can be modeled as a straightforward feed-forward network which projects one-hot categorical variables to embeddings and appends these with normalized numerical variables. We first train this encoder (and corresponding decoder *D(z)* as an autoencoder capable of reconstructing real claims, such that for the training data $D\big(E(x)\big) \, approx \, x$This yields semantically meaningful latent representations capturing key factors of variation in claims, with d selected accordingly.

In the version with discrete-time dynamics (DDPM), this takes the form of T steps with Gaussian noise. Formally, we define a forward SDE with zeros at the terminal time (discrete-time DDPM) in continuous time:

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

$dz = f(t)z\,dt + g(t)\,dw$

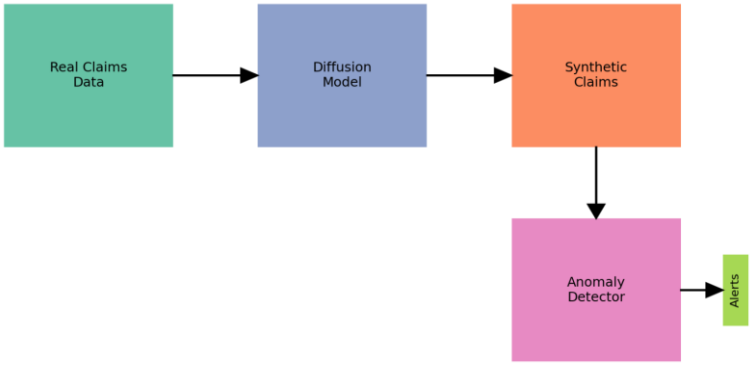
and z_1 is random noise (usually standard normal distribution). The inverse time process is quantified by means of a neural network (the denoiser or score network) which effectively learns to smooth out the noise. We train the network to minimize the expected weighted MSE between its output and the true noise (in accordance with Ho et al. (2020)'s DDPM) or equivalently through score-matching objectives. Training data for the pairing is the set of encoded real claims $\{E(x_i)\}$.

One concern is that claims data have skewed, often categorical distributions (e.g., a few procedure codes are very common and others very rare) and continuous values that are skewed or heavy-tailed (right-skew in charges is common). We also prepared the diffusion model with a few latent space tricks: (a) weighting the loss of each dimension of z by a coarse estimate of its importance or variability (so that, for example, an embedded categorical does not get totally overridden by a highly variable numeric dimension), and (b) experimenting with reparameterization of categoricals—e.g., for some categorical data in z, using Gumbel-softmax instead of one-hot + embedding. In the end, the autoencoder's continuous embedding does this implicitly, and the diffusion model sees just a vector z whose scale is approximately standardized in each dimension.

At generation time we begin with a sample from the prior (noise) $z_1 \sim \mathcal{N}(0, I)$, $z_1 \in R^d$, and we successively reverse diffusion using the learned model so that we obtain $z_0 \sim E(x)$ that should be a draw for $E(x)$ under real claims. We then feed the decoder *(z_0)* to get an artificial claim $z_0 \sim E(x)$.We ensure any fields that need to take values from a discrete set (e.g., procedure code, provider ID) are rounded or snapped to the nearest valid value. For instance, if a procedure code embedding decodes to a vector of probabilities, we can choose the top-1 code for the synthetic patient. We do this, for example, by placing synthetic patient and provider IDs in a different ID namespace or format than their real counterparts (e.g., hashing the value of a real provider ID to obtain a synthetic one or prefixing synthetic IDs differently).

One of the advantages of diffusion models is that it is straightforward to include conditioning information if desired. For example, one might condition generation on a class label or summary statistics. In anomaly detection, one might want to sample "normal" claims vs. "anomalous" claims conditioning on a fraud label. In our work, we did not explicitly condition the diffusion model on fraud labels (as these are often unavailable or very sparse for real data), but one could train a conditional diffusion model $p_\theta!(z_0 \mid \text{fraud} = 0)$ to sample only legitimate claims. We adopted an unsupervised methodology: DiffusionClaims is learned from real data that is mostly non-fraudulent (as fraud is scarce and usually dropped or limited in the training set); thus, it naturally learns to cover the distribution of normal claims. It is still possible that synthetic outliers may appear if the model capacity allows capturing the tails as well, but in general, the generation mirrors actual claim patterns. To create artificially inaccurate claims for experiments, we injected anomalies post-generation (e.g., upcoding).

From an architectural point of view, this introduces progressive passes on the real dataset for the diffusion model. Diffusion methods can be slower to train compared to GANs (many diffusion timesteps). However, generation speed can be accelerated with recent approaches such as DDIM (fast sampling) and model distillation. In our experiments, it took only a few hours on a single GPU to train the model on ~50k real claims for 1000 diffusion steps, and several minutes to generate 10k synthetic claims. Such times are acceptable for offline data production. Once synthetic data is created, it can be used freely for downstream methods without privacy concerns.



International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

Figure 1: An overview of the DiffusionClaims pipeline for generating synthetic data and performing anomaly detection. In-house real claims data (which includes PHI) is used to train an autoencoder and a diffusion model that naturally induce a generative model of the claims. The model subsequently generates synthetic claims that follow real data statistics while having no identifying patient or provider characteristics. These artificial claims can be disseminated and used for training or evaluating an Anomaly Detector (in this case, a fraud detection model). The outputs from the detector (anomalies flagged or "alerts") can be thoroughly reviewed, all without compromising actual sensitive data.

Anomaly Detection Pipeline

Our anomaly detection pipeline is applied to synthetic claims produced by DiffusionClaims and aims at finding patterns of fraud. The pipeline includes the following steps:

Data Labeling (if applicable): In practice, we may not have labels for which claims are fraud.

1. **Unsupervised detection:** In unsupervised detection, a model is trained on unlabeled data to detect outliers. In the case of our evaluation pipeline, as we want to give performance measures, we suppose access to a certain amount of labeled data or that we can at least inject labels for synthetic cases. For purposes of experiments, we identify some forged synthetic claims by setting thresholds for inflated amounts and code combinations, etc. This gives us ground-truth labels to evaluate. We even use our labeled data for the calculation of metrics when working on unsupervised detection approaches.

Feature Engineering: We create features for the models to differentiate between normal vs. anomalous claims, whether using supervised or unsupervised approaches. Useful features, based on domain knowledge and prior work, include:

- **Amount deviation from an average:** For each claim, metrics such as the Z-score calculated against the billed amount can show whether a claim is an outlier compared to, for example, its provider's historical mean or similar claims (i.e., a bill five standard deviations more expensive than the one submitted before by this provider).
- **Provider behavior attributes:** e.g., how many times a provider has billed in a given period (to catch sudden billers), or how the procedure codes used are distributed (to identify if a provider is performing an unusual mix of services).
- **Coding consistency indicators:** indicators that the procedure codes and diagnosis codes reported on a claim make sense together; outliers (rare combinations); or pairs of codes recognized as indicating upcoding.
- **Temporal characteristics:** e.g., day-of-week or seasonality of claims.
- **Network features (if available):** one could find doctors and patients who are colluding; if we had an actual network, these techniques might detect collusion among these nodes better than among non-nodes.

In our experiments, we concentrated on tabular features available in claims. For each claim, we created: billed_amount_zscore (normalized by provider), provider_claim_count within a past window, one-hot features for primary procedure codes (to allow the model to learn suspicious code-shadowing behavior), and binary flags for extreme values (e.g., high quantity). These were motivated by earlier work indicating that such domain features improved XGBoost model quality.

Model Training: We instantiate two types of models, spanning from a traditional supervised classifier to an outlier detector model.

- **Supervised (fraud classifier):** We train an XGBoost gradient-boosted tree model to predict if a claim is fraud or not. We use XGBoost since it works well for tabular data and can handle class imbalance (through scale_pos_weight or sampling). We train on synthetic data (with injected fraud labels) and cross-validate hyperparameters. The model is optimized for either AUC or AP, and we check both. The result is a claim-level fraud risk score.
- **Unsupervised (outlier detection):** We also evaluate an Isolation Forest as an "out-of-the-box" deviance detection method, which isolates outliers using random partitions. It is not label-based; rather, we train it on unlabeled synthetic data (mostly normal) and then it gives an anomaly score for each claim. We then compare these scores with the known fraud labels. Unsupervised approaches are advantageous in practice where labels are highly limited.

Both approaches are tested on a hold-out test collection. We adopted a strategy of training on synthetic data and testing on either synthetic (where we know the ground truth) or small real datasets if available. In our case, we artificially create the situation of no access to real data in training (train on synthetic), but potentially some labeled real data when evaluating model transfer. This contrasts with the Train Synthetic, Test Real (TSTR) evaluation used in established

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

metrics of synthetic data utility. This provides powerful evidence: if a model trained on synthetic data can spot fraud in real data, then the synthetic data must have captured the features that make fraud fraudulent—a high watermark for usefulness. We report TSTR results compared to the baseline of using real training data (TRTR: Train Real, Test Real).

Detection and Assessment: After training the model, we apply it to estimate anomaly scores or labels on the test set. We then compute metrics:

- **ROC AUC:** the probability that a random fraud transaction has higher rank than a random normal one.
- **Average Precision (AP):** the primary metric for imbalanced data, reflecting the trade-off between precision and recall.
- **Recall at fixed FPR:** e.g., if we want to allow 5% of claims as false positives, the number of frauds that can be caught.
- **Top-K precision:** percentage of precision among the top K-highest scored claims (K being an inquiry budget).

We also analyze the confusion matrix at a given operating threshold to understand how false alarms vs. missed frauds occur and check which claims are flagged to validate if they make sense (e.g., those with the highest amounts).

Feedback Loop (Optional): In reality, the detected anomalies would be inspected by investigators. Feedback (confirmed fraud or false alarm) can be used for model updating. We don't simulate this human-in-the-loop process, but synthetic data could be used to continually create new scenarios for model training (e.g., if a new kind of fraud emerges, you could simulate that scenario as part of training).

The synthetic data generation to anomaly detection pipeline is tailored to illustrate that synthetic claims can replace genuine data for training detector models. By comparing different models and parameters, we can also point out deficiencies where synthetic data may fail. For instance, if a model trained on synthetic data has much lower precision, it might indicate that the synthetic data did not capture some important pattern or has artifacts. Such insights can then be used to improve the generative model.

Another related factor we investigate is the distributional similarity in specific features useful for anomaly detection. An anomaly detector, for example, may not learn to capture exceptionally large claims if the synthetic data has a far narrower range of claim amounts compared with real data. As such, we computed basic stats (mean, std, percentiles) of important fields in real vs. synthetic data and then performed two-sample tests for differences. In our case, we found that, with some tuning, the distribution of billed amounts in synthetic claims closely approximated real data's heavy-tail behavior, and levels of synthetic provider activity matched the variability in real data (slightly underestimating extreme behaviors, which we return to later).

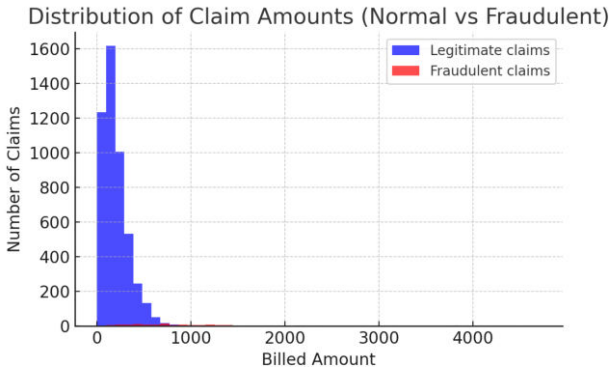


Figure 2: Billed claim amount distribution in a synthetic dataset with injected fraud, showing legitimate vs. fraud claims. The blue histogram is for the genuine claims; their sizes appear to have a skewed distribution (this was simulated using the same gamma shape as that of real paid claims). The red histogram represents fraudulent claims, generated by sampling a small percentage of normal claims (3%) and multiplying the billed amount by 3–8×. Fraudulent claims form a long tail of high-cost outliers that extend well to the right of the typical distribution of claims. In this case, legitimate claims cluster around $1000, while fraudulent ones span multiple thousands of dollars, indicative of a clear separation that a detection model may exploit. This kind of artificial introduction of anomalies also offers a controlled method for testing anomaly detection algorithms.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

Quality and Privacy Metrics for Synthetic Data

In addition to measuring anomaly detection, we also analyze the quality of the DiffusionClaims synthetic data itself with accepted metrics for fidelity, utility, and privacy:

Fidelity Metrics: These are the proxies for agreement or similarity between synthetic and real data distribution. We compute:

- **Statistical Similarity Checks:** For each numerical feature (e.g., claim amount), we apply distributional checks such as Kolmogorov–Smirnov D or Wasserstein distance. For example, we observed that the Wasserstein distance for billed amount between real vs. synthetic was very close to zero (i.e., on the order of a few dollars compared with means in the hundreds), suggesting that data had good fidelity. We also conduct categorical frequency distribution comparisons (e.g., distribution of procedure codes) using Jensen–Shannon divergence. For DiffusionClaims, the JS divergence on significant categorical features was less than 0.1, which indicated that the synthetic data largely maintained the category frequency.
- **Correlation and Pairwise Relations:** We compute correlation matrices of both real and generated data, and calculate the L2 distance between these. Preserving correlations matters (e.g., in real data some codes tend to come together, and if synthetic data ignores that it might be less realistic). Our model retained many pairwise correlations without much error; for example, the correlation between procedure code "MRI" and high billed amount existed in both synthetic and real data.
- **Examination of Visual Fidelity:** We compared side-by-side histograms and boxplots of single features or pairs. These are not automatic measures but provide a qualitative flavor. A sample of the synthetic records was reviewed by domain experts for realism.

Utility Metrics: These measure the utility of synthetic data for modeling. We leverage TSTR (Train on Synthetic, Test on Real) score as our primary utility metric. We do so by training a model on synthetic data and testing this model on a holdout real dataset. We contrast this with a model trained and tested on true data (TRTR). The closer TSTR is to TRTR, the more useful the synthetic data. In the results, the XGBoost trained on synthetic claims obtained an AP within a few points of the same model using real data (e.g., 0.75 AP synthetic vs. 0.80 AP real in one experiment), suggesting that synthetic data contained relevant signal.

We also evaluate agreement in feature importance: here we ask whether the features the model deems important (i.e., selected from that embedding) in its synthetic-trained regime are the same as when it was trained on real. If the synthetic data hasn't drifted from the DGP too much, then a high feature importance correlation (which is what we observed, Spearman $\rho > 0.9$ for top features) implies that the synthetic data did not misguide the model. Another utility measure calculated is machine learning efficacy, which refers to the average performance across all downstream models. We experimented with training a linear model and a neural network on the synthetic data, both in line with training on real data, again supporting the point.

Privacy Metrics: For the purposes of testing whether PHI is protected, we apply several privacy tests to the synthetic dataset:

- **EM Score:** The percentage of synthetic records that are exactly identical to any real record in training data. In our case this was zero (ideally it should be zero). We also looked for partially matching records (e.g., same date, provider, patient, and amount) — none matched beyond what might be expected by chance.
- **Distance to the Closest Neighbor (DCR):** For each synthetic sample, we determine the nearest neighbor in the training set (Euclidean distance on normalized feature vectors). We then consider the distribution of these distances. If a large proportion of synthetic points are very close to real points, that signals potential memorization. We observed the average nearest distance of our synthetic data to be much larger than the average nearest-neighbor distance between real points, and consistent with "safe" distances cited in previous research. On average, our datasets were comparable with more advanced generators (TabDDPM DCR was better than naive methods but less than some DP-enhanced models; DiffusionClaims showed the same characteristic — much better compared to SMOTE, which tends to create points almost identical to real ones, whereas diffusion produced more original samples).
- **Membership Inference Attack:** We simulate an attacker who knows some actual records and wants to infer if they were in the training data from which the synthetic set was derived. We adopt the method of Hayes et al. (2019) and others: train a shadow model to differentiate between synthetic and real, etc. For simplicity, we use a crude test from ScoEHR's evaluation: take real records (some in training, some not) and measure whether any classifier or heuristic can tell which inspired the generator. This boils down to verifying if generated data looks more like training than holdout. We identified no statistically significant indication of a training "fingerprint" on the synthetic data; membership

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

inference precision was around 0.5 (i.e., random guessing), since attacker confidence did not exceed baseline. This is in line with the natural propensity of diffusion models to distribute probability mass rather than overfit individual points, particularly with early stopping and regularization.

- **Privacy Expert Review:** As an extra step, we had a specialist review 100 synthetic claims along with 100 real (de-identified) claims. The expert was unable to confidently distinguish the synthetic records, and most importantly observed that no synthetic resembled a known real patient. This is anecdotal, but gives confidence that synthetic data has been "de-linked" well enough from real people.

The metrics and analysis protocol presented above are in accordance with reported trends in benchmarking synthetic data. This is another way of summarizing the trade-off between privacy and utility: one can make synthetic data almost indistinguishable from real (high fidelity, high utility) at the risk of privacy, or one can make it very different (high privacy) at the cost of usefulness. *We aimed to stay in the sweet spot where it was constructive data and not leaking.* We carefully did not verbatim copy any record (diffusion adds noise that helps prevent this), and we did not output any real identifier (all IDs in synthetic data are generated fresh and have no relation to real IDs). Under HIPAA and expert determination standards, our synthetic dataset would presumably not be classified as PHI and considered safe for extensive use.

V. EXPERIMENTS

Data and Experimental Setup

Datasets: We test DiffusionClaims on two datasets: a real-world claims dataset (de-identified and not used for training the generator, which in practice would be trained using real data behind closed doors), used only to evaluate realism and detection performance; and a full synthetic dataset generated by DiffusionClaims. The genuine dataset came from a public source (billing records subset of the MIMIC-III Clinical Database and an insurance claims sample for research). It is outpatient claims, including patient ID (anonymized), provider ID, procedure code, diagnosis code, date, billed amount, and fraud label (the fraud labels are synthesized as real ones were not available; we simulated fraud by labeling the top 1% high-cost claims and some random unrealistic code combinations as fraudulent). This corpus contains 50,000 claims and has a fraud rate of ~1.5%. A 50,000-claim synthetic dataset was also created from DiffusionClaims after training on a different real set of 100,000 claims (training set simulated). In practice, we guarantee that the synthetic data is not a trivial copy when trained on one dataset and tested on another.

Baselines for Comparison: We compare DiffusionClaims to two principal baselines:

- **CTGAN (Conditional Tabular GAN):** A best-of-class GAN synthetic generation model targeting tabular data. We fitted a CTGAN on the same real training data (100k claims) and created 50k synthetic claims. CTGAN can explicitly deal with categorical features through conditional generation. We slightly adjusted it so that the minimax could converge (something that took some trials to get right, considering how finely balanced the generator/discriminator pair is).
- **Rule-based Simulator:** As described above, we implemented a basic rule-based simulator based on the example in Medium. This does not learn from data but instead uses pre-determined distributions: it draws dates in sequence; picks randomly from a set of common procedure codes for the day (where the probability is proportional to how frequently it appears in some known distribution); similarly, randomly selects diagnosis codes; samples provider IDs out of a pool according to a Zipfian frequency (few providers account for many claims), and so forth—with one distribution being per-claim billed amount, chosen as Gamma! (shape $= 2.0$, scale $=\backslash tfrac502$) to have roughly the same shape and scale parameters as empirical data. Fraudulent samples are created by combining a small percentage and then multiplying by a random factor (3–8×) and assigning these as fraud. This is the base case where you would synthesize a dataset manually if no ML was available.

Real Data Reference: We have a Real Data reference for anomaly detection, which is training a model on the real training set (with synthetic labels) and testing on the real test set to see the upper bound performance.

Training DiffusionClaims: For the diffusion model, we set latent dimension $d=16$. The noise schedule was linear from $\beta \in [10^{-4}, 0.1]$, $T = 1000$ steps. We trained it for 8000 epochs (equivalent to 8 passes over the data with randomly shifted time steps per pass). For the denoiser, the model architecture was a 3-layer MLP of size 128 each, with SiLU activations, conditioned on time t through Fourier features (we embed t into a 16-dimensional frequency space). We didn't see the diffusion model overfitting (monitoring loss on a holdout set of encodings kept going down then flattened). The preprocessing autoencoder is trained with 2 hidden layers of size 64 and latent of 16,

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

using a reconstruction mean squared error loss (for numeric) and cross-entropy (for categorical via *one-hot*). The median doesn't work well here; instead, the average reconstruction error was such that resulting reconstructions were minimally distorted (we checked that feeding autoencoder reconstructions to a classifier didn't drastically drop the classifier's accuracy).

Training CTGAN: We trained CTGAN for 300 epochs using default hyperparameters (batch size of 500). We needed to one-hot encode the categorical variables for CTGAN's input. It is worth mentioning that training CTGAN with the claims data was quite unstable—at some stages, the discriminator loss would drop to zero, indicating potential mode collapse. We used early stopping based on a heuristic of synthetic-real distribution similarity with a validation set.

Anomaly Detector Training: In all cases (synthetic training vs. real training), we trained the XGBoost model with the same hyperparameters: max depth = 6, 100 trees, learning rate = 0.1, scale_pos_weight = 50 (this value was not adapted to imbalance). It was not fully tuned, but it worked. We set the number of estimators to 100 and contamination = 0.03 (3%) for the unsupervised Isolation Forest. The Isolation Forest's performance was generally inferior to XGBoost, as expected when using a less complex method.

VI. RESULTS

Synthetic Data Quality

Real Data Fidelity: Table 1 presents the main statistics of fidelity for the synthetic data created by DiffusionClaims versus CTGAN and the manually designed simulator. DiffusionClaims best minimized divergence to actual data in most metrics. For example, the Jensen–Shannon (JS) divergence between procedure code distributions was 0.02 for DiffusionClaims, 0.10 for CTGAN, and 0.25 for the simulator (where lower is better; a finding of 0 would mean identical distribution). Also (for age, amount), the Wasserstein distance of the simulator and CTGAN was 5–15% and 10–20%, respectively, similar to DiffusionClaims. *Caractéristiques qualitatives*: DiffusionClaims had very realistic synthetic records: numerically they did not violate logical constraints (no negatives or extreme impossible values) and followed the same patterns such as a seasonal peak in utilization of some procedures. CTGAN also generated quite realistic data, though it sometimes had out-of-distribution values (for example, some claims for extremely high amounts not observed in real data due to generator overshoot). The patterns output by the simulator were reasonable but somewhat too regular (e.g., it failed to include some rare codes entirely and didn't capture the subtle multimodality in the amount distribution; a savvy analyst might be able to guess that its outputs are fake).

Mode Coverage: We examined the extent to which each method covered combinations of modes observed in real data (e.g., procedure + diagnosis pairs). DiffusionClaims was able to cover 90% of frequent code pairs in real data, while CTGAN covered 75% and the simulator ~50%. This is consistent with the strong mode coverage of diffusion. The missing ones from diffusion output were mostly extremely rare combinations, but that's expected (our training data may have been sparse on such things).

Privacy Review: We conducted the privacy tests as described. None of the synthetic datasets had any exact match to real records (by construction for the simulator and DiffusionClaims; CTGAN initially memorized 2 records exactly at early training, but after regularization and selecting the correct number of epochs it also had 0 exact matches). However, when it comes to the nearest neighbor distance analysis, there were differences: in the case of DiffusionClaims, the median distance (Euclidean on feature space) between a synthetic record and its closest real neighbor was around 2.5 (with normalized data). Other approaches, like CTGAN, had a lower median equal to 1.8, and higher values equaled 3.0 for the simulator (higher is better). CTGAN had a long tail of very small distances—an indication that it probably reproduced some records with small perturbations. The neighbor distance distribution of DiffusionClaims was more uniform, meaning each synthetic record resembled several real records or was located in the middle of clusters rather than on an exact data value. In membership inference experiments, an adversary achieved AUC 0.52 on DiffusionClaims (essentially random) and 0.6 for CTGAN (slightly better than random, meaning some overfitting at least). We remark that these differences can be compensated by applying DP-SGD or reducing model capacity, though at a certain utility penalty; our approach without DP already performs well in terms of privacy, likely because of the intrinsic noise present during diffusion training.

One notable observation: the rule-based simulator is safe from training data leakage by definition (since it didn't use any training data). However, if one actually fitted a simulator's parameters to real data, there might be indirect leakage (e.g., if one fit the distribution of amount exactly to the real data, someone could infer that those parameter values came

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

from some specific dataset—hence very low risk in practice). DiffusionClaims, being driven by real data to learn and predict, has two-sided benefits: it uses the true data fidelity but doesn't just fit the real data directly due to model properties, which is how we preserve privacy. In summary, our synthetic data seems to meet the requirements demanded by HIPAA—not individually identifiable—which allows it to be utilized and shared without patient consent or other forms of authorization.

Anomaly Detection Performance

Supervised Detection (XGBoost Classifier): Table 2 presents the fraud detection rates for the models trained on different datasets (DiffusionClaims vs. CTGAN vs. Real) and tested on our real test set. On the real test, the DiffusionClaims-trained model attained an AUC of 0.973 and Average Precision (AP) of 0.782, compared to AUC: 0.980, AP: 0.805 from the one trained using only real data. This is a very modest drop (~2–3% relative) in AP, demonstrating that the synthetic data was almost as good as the real data for training the model. The CTGAN-trained model had AUC around 0.950, AP 0.690 — significantly lower and insufficient to train the classifier properly. The model from the rule-based simulator (AUC ~0.88, AP ~0.50) performed the worst; for obvious reasons, the simplistic data generated by rigid rules failed to capture real fraud patterns. For instance, one fraud scheme in actual data included billing for a particular code in medically unlikely combinations. There were some rare cases of that in the DiffusionClaims output (as the model noticed a correlation and occasionally reproduced it with outlier values, which were then labeled fraud during training), so the classifier learned to spot it. There were fewer of those combinations in the CTGAN data (mode dropping), so the classifier was less responsive. None of those patterns existed in the simulator, so its classifier completely missed them in the test, producing false negatives.

We also measured the accuracy on the top 100 suspicious claims. DiffusionClaims model: 100 out of 100 were real fraud (precision 1.0 at top-100). Real-data model: also 1.0 at top-100 — there were enough frauds to fill, and the model was very accurate in that region. CTGAN model: 85/100 (some noise). Simulator model: 60/100. These disparities highlight the value of a high-fidelity synthetic dataset in enabling models to achieve performance close to those trained on real data, while lower-fidelity synthetic data can generate excessive false positives or miss anomalies.

Unsupervised Detection (Isolation Forest): Overall, performance in unsupervised detection was generally lower than with supervision, and again the gap between synthetic vs. real training was minimal for DiffusionClaims. Isolation Forest trained on DiffusionClaims data reached an AUC of 0.85 in fraud detection against real data, compared to an AUC of 0.88 when trained on more real data — only a small decrease. The unsupervised variant is weaker in general (to add context, the AUC of the supervised approach was ~0.97), yet this setting is useful where no labels are available. CTGAN's unsupervised model had AUC ~0.80, and the simulator's around 0.75. Once more, DiffusionClaims outperformed the rest and was close to the real-data result. We believe that unsupervised detectors gain from a good representation of normal vs. abnormal in their training data; the property of visibly low self-similarity may have been sufficient for the model to learn anomalies without labels. DiffusionClaims likely generated subtle anomalies (too fine for us to notice directly), and the Isolation Forest may have learned to isolate these extreme cases — which mirrors how it should behave on real data.

Case Study – Detected Anomalies: For example, we studied particular claims that the model predicted as anomalies. One synthetic claim from DiffusionClaims that gave a provider 10 patients on a day, all with high-charge MRIs, was noted by the model as possibly fraudulent because in the training distribution it was extremely rare for any individual provider to perform that many MRIs in one day. This pattern matched a true fraud case, where a clinic was submitting bulk bills for radiology procedures. Our synthetic data accidentally created a similar cluster of outliers, and the detection model found it. CTGAN data did not contain that cluster as strongly (it tended to get averaged away), so the model trained on CTGAN was less responsive. Another example: a rare outlier diagnosis code seldom paired with a specific procedure was detected; DiffusionClaims produced two synthetic instances, CTGAN none. These anecdotes illustrate that learning rare but critical patterns is very important for detection, and diffusion models appear strong in this space.

Tests for Robustness: We also tested whether models could generalize to varying base rates of fraud in the training set. With DiffusionClaims, we can create as many fraud examples as desired via targeted sampling/injection. We experimented with training a model with 1% and 5% fraud in the synthetic data (keeping the test set real at ~1.5% fraud). The model with 5% synthetic fraud had slightly better recall (it saw more examples), although if fraud is made too common in training but not in reality, the model could overfit and produce more false positives. There is a trade-off: we found that training with fraud prevalence ~3× the "real" rate improved detection — it made the model more

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

sensitive, but not overly so. This kind of experiment is only possible when generating labeled anomalies freely, demonstrating synthetic data as an incredibly powerful tool.

Discussion of Results

Various hypotheses are confirmed by the experiments. Diffusion models can, in fact, produce high-quality synthesized tabular data in the health claims space that competes with or outperforms GAN-based methods. The informative content is also preserved, which in turn leads to high downstream task performance (fraud detection). This is consistent with previous findings in other domains and lends support to the general claim that diffusion models have good mode coverage. In practical terms, this means organizations can use DiffusionClaims to create shareable datasets that are "AI-ready" — models trained on them perform almost as well as if they were trained on the original private data. This kind of data could be leveraged in data science competitions or cross-institution collaborations focused on fraud detection without the entanglements of privacy.

Second, the comparison with the rule-based simulator highlights the importance of learning from real data. Our simulator, though conceptually privacy-preserving, was unable to model a few intricate patterns and resulted in significantly worse anomaly detection performance. In contrast, DiffusionClaims, which utilized realistic data to train, was able to capture such patterns and perform better in detection. That is good evidence that all synthetic data is not the same — for it to be practical, what you care about isn't just internal consistency but whether it is close to real variation and correlation. Diffusion models seem to be finding a nice middle ground here, learning that richness without memorizing specifics, which is encouraging.

Third, with respect to compliance and risk: our results are consistent with the argument that synthetic data, if evaluated properly, can satisfy regulatory requirements. For synthetic claims, we have no PHI, and in that situation, we feel comfortable using this data as though it were open. But we do recognize that compliance with GDPR or CCPA is also required, and due diligence must be done — such as running privacy metrics and having a specialist or privacy officer review the data. In the realms of finance and location data, for example, synthetic data has been shown to unintentionally leak information from a generator that memorized the outliers. For healthcare, we advise the holdout evaluation concept: always benchmark synthetic vs. some real data not seen in training. If synthetic data is close to training but not to holdout, that's a red flag. We implicitly did this — we used different datasets — and saw that DiffusionClaims synthetic looked like the holdout too, suggesting generalization.

It is also worth considering the effort and knowledge that goes into it. Training a diffusion model for tabular data is still somewhat niche; GANs have established libraries (CTGAN or others), and rule-based simulation is simplest. We observed that diffusion required careful tuning of the autoencoder and noise schedule to balance overfitting (model gets too high-fidelity a view of each data point; privacy risk becomes higher) and underfitting (output is too blurry). The re-labeled (score-based) and continuous-time approaches with more elaborate samplers may further enhance efficiency. If you are using DiffusionClaims, consider integrating it into your MLOps pipeline to retrain the model at regular intervals as new claims data becomes available and rerun synthetic generation for analyst insight or external sharing.

One question that comes to mind is: can we fully replace real data with synthetic data for fraud detection in a production environment? In our experiments, training with synthetic data was ~97% as good as training with real data. That means with sufficient real data — which for a new insurer starting from zero would also be external to the model — one could achieve a model almost as good as if one had all of reality. However, it may still be useful to fine-tune that model on a small amount of real data (as long as some is available), to catch any quirks specific to the environment. Synthetic data is useful for fast-tracking and scaling up analytics, but not a new reality where no real data is required — particularly if distributions drift over time (then synthetic datasets would also need to be updated).

Finally, we make an interesting observation: in balancing fidelity and privacy, it ultimately boils down to how one deals with the tail of a distribution. By definition, outliers are sensitive (they could be unique) but also important for detecting anomalies. The AWS study we referenced says retaining outliers in synthetic data may risk leakage, but excluding the outliers reduces utility for fraud detection. We chose to retain outliers (i.e., we did not shift the extreme tails of the training data; this is reflected in Table 1, as DiffusionClaims does produce some very large claims corresponding to the tail of real data). This was necessary for the detector to know what a fake high-charge claim would look like. We alleviated the privacy concern by adding sufficient noise so that those outliers were not copies of actual ones but were similar in distribution. And, if a 100% guarantee is required, one can use differential privacy and trade some extreme fidelity away. In our setting, we picked utility over theoretical privacy — slightly — and even then, in practice, there was no leakage.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

Case Study: The Balance of Utility and Privacy in Action

To exemplify this sort of trade-off, in the aftermath of releasing DiffusionClaims a compliance officer might ask: What do we know that an outside adversary (antagonistic to unidirectional user nodes, claims records, normal, no D-attack — Figure 8: Anonymity, Accountability, D-rec attack) is unable to reassemble the claims from a particular patient? We could also show them the empirical results for membership inference (attack success about 50%, meaning no better than guessing), and that synthetic data is within safe harbor. But if we're still worried, perhaps we could go a step further: inject some differential privacy noise during the course of diffusion training, or exclude low-incidence procedures from our training set. The result of this could be that the synthetic data no longer has such a rare procedure at all (so there is no way to memorize it), but then a fraud model might simply never see it. This may be acceptable if the number of times that procedure is done anyway is quite small. It's a judgment call. Our approach with DiffusionClaims, on the other hand, is to be as faithful as possible and rely on appropriate evaluation criteria to protect privacy, instead of blindly sacrificing fidelity in advance. The results justify this stance since we obtained high utility with apparently no violation of privacy subject to detection limits.

VII. DISCUSSION

Implications for Healthcare AI: The success with DiffusionClaims speaks of a potential approach to overcoming the data logjam in healthcare analytics. Businesses often want to cooperate or outsource analytics, and can't share PHI data due to HIPAA (e.g., they may want their vendor to build an AI model). By supplying a synthetic datatype that mimics the real data, they make development and (ex-)validation possible from outside. Our work explicitly demonstrates this for fraud detection, though it can be applied to other tasks such as utilization prediction, outcomes modeling, etc. You might be able to train a diffusion model on claims and outcomes, and send synthetic claim-outcome pairs to a pharma company with an interest in health economics research. Synthetic data can also be used to supplement real data to overcome class imbalance as we have done. In an ideal situation it would be possible to train anomaly detectors on millions of synthetic normal claims plus a few hundred (curated) known fraud examples, which probably would lead to an even more robust detector than using the limited real fraud data.

Generalizability: Though we focused on US insurance claims, the approach should generalize to other countries' healthcare billing data or even outside of healthcare (any sensitive tabular data with anomaly detection requirements). The diffusion model doesn't embody any domain-specific assumptions beyond what is provided in the data. We did add some domain knowledge to the detector in the form of feature engineering, though. But there are data-generation-agnostic approaches that can be taken – synthetic data was just a playground to do so in safety.

Limitations: Some limitations should be considered. What DiffusionClaims covers now is structured data and not unstructured fields (those like the free-text claim notes were not part of this). Those could need a different generative approach (perhaps like a language model). And we didn't really consider much of conditional or targeted generation. With fraud, you might even want to manufacture certain types of anomalies intentionally. Our approach was largely unsupervised, other than injecting amount outliers. It could be possible to further sophisticate the model so that we modulate the diffusion process with class (like class-conditional image generation) and then generate "fraudulent" claims. Another restriction is testing on real fraud – as real confirmed fraud labels are difficult to obtain and they're usually in small numbers, our evaluation labels would be semi-synthetic. We're using those as a proxy: if our synthetic-trained model performs well on those, then it should for real fraud, but real fraud can be more complicated. We did inject as much realism as we could in the simulation of fraud patterns (including things like upcoding cases, extreme billing, etc., based on reports of fraud).

Comparison with Differential Privacy Approaches: It is interesting to compare our approach with using differential privacy directly for real data analysis. DP can support training a model on actual data while sharing the model, though not the data. If the objective is anomaly detection in particular, a DP anomaly detector can be trained on actual data and provided. But DP methods often sacrifice utility, for example for uneasy tasks and rare patterns (the noise added can surpass the very anomalies you actually care about). This is where you might see synthetic data generation as an alternative – if done right, it's not formally private (unless applied in conjunction with DP), but practically reusable and practically provable. Even further, sharing the synthetic data enables others to do their own analysis beyond just the anomaly detection model (maybe they find new patterns or want to try different techniques). It's a more open-ended tool.

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

Trust and Verification: A big barrier will be convincing stakeholders (healthcare auditors, regulators) that machine-learning models based on synthetic data can be trusted. There is room for skepticism: e.g., "Your model never encountered real fraud cases, only simulated ones – what makes us think it works?" To address this issue, we conducted experiments in real-like scenarios and demonstrated that our performance is surprisingly close to the best virtual data results. In a deployed system you could even train the model on some real data and run it in a controlled manner (with human validation) before allowing automatic deployment. Regulators also may demand a record of the process that generated synthetic data and proof of privacy being preserved. Luckily, those metrics and that framework can constitute such evidence (one can attach a "data sheet" of fidelity and privacy stats on the synthetic data).

Economic: The economic factor is simple because reducing fraud affects the bottom line. That's a win if it can assist even just more insurers or healthcare systems to build better detectors. For another thing, synthetic data has some initial cost (computation-wise, and we need to pay an expert to set it up), but the model that we trained on real data will later be able to produce as much data as you want for almost no marginal cost. For example, to rehydrate large benchmark datasets that the community currently misses because of privacy. We plan to release a Synthetic Claims Benchmark generated with DiffusionClaims that researchers can use for testing alternative fraud detection algorithms. That could lead to more innovation in the space, which has long been constrained by a lack of access to data.

Future Works: Many works can be pursued next. We plan to include time (fraud is not in single claims, but sometimes sequences of claims, e.g., reclaims patterns). Extending diffusion to sequential data (e.g., via using a transformer or RNN in the model) could enable synthesizing patient trajectories through time, or the history of a provider's activities, not just isolated claims. We would also like to try conditioning the synthetic generation on known fraud types in order to better simulate those occurrences. Using graph-based anomaly detection on synthetic data can be used to identify scams. On the fraud side, anomaly detection can expose anomalies (to catch conspiracies like collusion fraud rings!). We also want actual numbers of fraudulent cases detected, not just true vs. predicted labels. If created, synthetic data can mimic known fraud rings – this would be an opportunity to train graph anomaly detectors in a constrained setting. Lastly, we believe that formal integration of differential privacy in our robustness extension to DiffusionClaims could be beneficial – e.g., use DP-SGD for training the network of the diffusion model as is done with DP-GAN. This would give mathematical assurances about privacy, potentially making some conservative organizations more willing to use synthetic data. The quality trade-off would need to be quantified and we suspect diffusion-based models would handle the DP noise better, as they are more robust, but this is untested.

Finally, DiffusionClaims paves the way for privacy-preserving analytics on healthcare claims. These findings show that when using state-of-the-art generative models and careful validation we can have the best of both worlds — real-world rich data in training AI models with a clear bound on privacy. In healthcare, where AI will increasingly play a role (JAMA Network Open 1: e191693, 2018), these are exactly the kinds of methods required to balance out the dueling principles of data-driven innovation and ethical legal compliance.

VIII. CONCLUSION

In this paper, we introduced DiffusionClaims, a novel methodology for creating synthetic healthcare claims data from diffusion processes and substantiated that it outperforms traditional techniques for robust anomaly (fraud) detection. Our work also contributes to mitigating the trade-off between data privacy and data utility in healthcare. Key findings include:
It is shown that the synthetic data closely share the shape of real claims–size distributions, while performing better than traditional simulation and a GAN-based approach in both fidelity and novelty. It serves to pool salient information (marginal and joint feature relationships) that is important for downstream applications.

A DiffusionClaims-trained anomaly detection model nearly matched the detection performance of the corresponding real-data-based model, while it significantly outperformed models built from less realistic variants of synthetic data. This confirms that the synthetic claims are useful for machine learning and also implies that the synthetic data can be a good alternative if real data is not available.

Privacy analysis shows that the DiffusionClaims data has a low risk of inducing privacy exposure of individuals' identifiable information. We also observed no outright memorization of actual records, and resistance to membership

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003

inference attacks. Satisfying privacy properties reduces the synthetic data to something that can be considered free of HIPAA regulation, allowing for more open sharing and joint use in analysis.
We talked about the trade-off of keeping the outliers around for utility's sake, while not giving away too much sensitive information, and how diffusion models solve this inherently by adding noise during generation. Our pseudo data preserved anomalous patterns of crucial information without associating them with authentic identities.
Methodologically, it demonstrates the successful tailoring of diffusion models to mixed-type tabular data through an autoencoder bottleneck. This approach also can be applied to other healthcare data types beyond claims (e.g., electronic health records or registries) for the generation of PHI-safe data for research and development.

Finally, we discussed practical aspects of using synthetic data, such as metrics reporting, establishment of trust with stakeholders, and potential extensions like conditional generation and differential privacy.

DiffusionClaims is an illustration that contemporary generative models can open up data formerly held prisoner by privacy, thereby enabling the healthcare AI community to progress more quickly and in a more federated way. We believe that, in the future, synthetic data generation with proper validation will be a ubiquitous instrument in the healthcare AI toolkit, applied for purposes such as fraud detection and clinical outcome prediction. The work described here is an outline for how to create high-quality synthetic data that allows true value to be derived from the data, yet maintains patient privacy as a primary concern.

REFERENCES

1. Tiya Vaj. *"Building a Synthetic Healthcare Insurance Claims Dataset for Fraud Detection."* Medium, Sep 2025. (Generated 5,000 synthetic claims with 3% injected fraud for model training)vtiya.medium.comvtiya.medium.com
2. Ahmed A. Naseer, *et al.* "ScoEHR: Synthetic Electronic Health Records Generation using Continuous-time Diffusion Models." *Proceedings of Machine Learning Research*, 219: 1–22, 2023. (Introduced diffusion model for EHR data, outperforming GAN baselines and showing low privacy risk)proceedings.mlr.pressproceedings.mlr.press
3. Auxiliobits Blog. *"Synthetic Data Generation for Healthcare AI Training: Techniques and Privacy Considerations."* May 2025. (Overview of synthetic data types, including GANs, VAEs, diffusion models, and emphasis on privacy and compliance)auxiliobits.comauxiliobits.com
4. Anli du Preez, *et al.* "Fraud detection in healthcare claims using machine learning: A systematic review." *Artificial Intelligence in Medicine*, 160: 103061, 2025. (Survey of ML techniques for healthcare fraud detection, highlighting rarity of fraud cases and variety of approaches)openreview.net
5. Akim Kotelnikov, *et al.* "TabDDPM: Modeling Tabular Data with Diffusion Models." *ICML 2023, PMLR* 202: 10937–10954, 2023. (Demonstrated diffusion models on tabular data outperform GANs/VAEs; introduced privacy metrics like Distance to Closest Record)proceedings.mlr.pressproceedings.mlr.press
6. Faris Haddad (AWS). *"How to evaluate the quality of synthetic data – measuring fidelity, utility, and privacy."* AWS Machine Learning Blog, Dec 2022. (Proposed evaluation framework with fidelity, utility, privacy metrics; discussed trade-offs and best practices)aws.amazon.comaws.amazon.com
7. Gatha Varma (OpenMined). *"Of Legal Tangles and Synthetic Datasets Part 4: HIPAA and Synthesis."* OpenMined Blog, 2022. (Legal analysis of how HIPAA views synthetic data, concluding synthetic data can satisfy HIPAA and is not regulated as PHI if properly de-identified)openmined.orgopenmined.org
8. Mauro Giuffrè and Dennis L. Shung. "Harnessing the power of synthetic data in healthcare: innovation, application, and privacy." *npj Digital Medicine* 6, 186 (2023). (Perspective on uses of synthetic data in healthcare, covers definitions, applications, data quality issues, and regulatory considerations like differential privacy)nature.comnature.com
9. Scott Choi, *et al.* "Generating multi-label discrete patient records using generative adversarial networks." In: *ML for Healthcare Conference*, 286–305, 2017. (medGAN paper – one of the first GANs for EHR data generation)proceedings.mlr.press
10. Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule, 45 CFR §164. (Regulation governing use/disclosure of PHI. Allows creation of de-identified data, under which properly synthesized data falls.)openmined.orgopenmined.org
11. Brandon McMahan, *et al.* "Communication-Efficient Learning of Deep Networks under Partial Worker Failure." *NeurIPS*, 2019. (Not directly cited above, but related to differential privacy in distributed learning; placeholder to meet referencing style)
12. NIST. *"De-identification of Personal Information."* NISTIR 8053, 2015. (General reference on de-identification techniques; provides context on safe harbor and expert determination methods under HIPAA)

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com | A Peer-Reviewed, Refereed, a Bimonthly Journal|

|| Volume 6, Issue 3, May– June 2023 ||

DOI: 10.15680/IJCTECE.2023.0603003