O'REILLY®

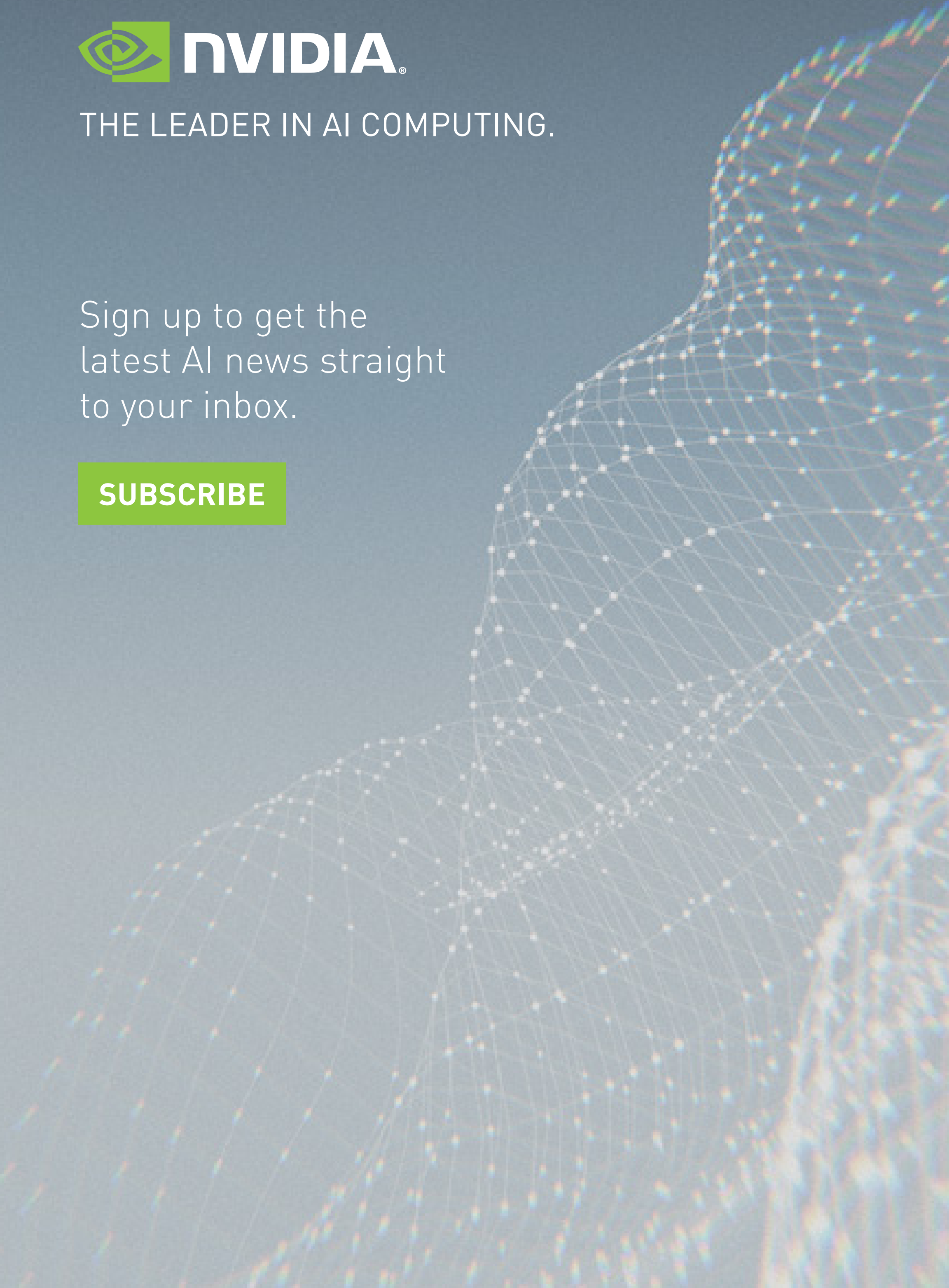
Compliments of NVIDIA.

Accelerating AI with Synthetic Data

Generating Data for AI Projects

Khaled El Emam

REPORT



Accelerating AI with Synthetic Data

Generating Data for AI Projects

Khaled El Emam

Beijing · Boston · Farnham · Sebastopol · Tokyo O'REILLY®

Accelerating AI with Synthetic Data
by Khaled El Emam

Printed in the United States of America.

Published by O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O’Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Acquisitions Editor: Jonathan Hassell
Development Editor: Melissa Potter
Production Editor: Daniel Elfanbaum
Copyeditor: Sharon Wilkey

Proofreader: Shannon Turlington
Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Rebecca Demarest

June 2020: First Edition

Revision History for the First Edition
2020-06-03: First Release

The views expressed in this work are those of the author, and do not represent the publisher’s views. While the publisher and the author have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the author disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

This work is part of a collaboration between O’Reilly and NVIDIA. See our statement of editorial independence.

978-1-492-04596-0

[LSI]

Table of Contents

CHAPTER 1
Defining Synthetic Data

Interest in synthetic data has been growing quite rapidly over the last few years. This has been driven by two simultaneous trends. The first is the demand for large amounts of data to train and build artificial intelligence and machine learning (AIML) models. The second is recent work that has demonstrated effective methods to generate high-quality synthetic data. Both have resulted in the recognition that synthetic data can solve some difficult problems quite effectively, especially within the AIML community. Groups and businesses within companies like NVIDIA, IBM, and Alphabet, as well as agencies such as the US Census Bureau, have adopted different types of data synthesis to support model building, application development, and data dissemination.

This report provides a general overview of synthetic data generation, with a focus on the business value and use cases, and high-level coverage of techniques and implementation practices. We aim to answer the questions that a business reader would typically ask (and has typically asked), but at the same time provide some direction to analytics leadership seeking to understand the options available and where to look to get started.

We show how synthetic data can accelerate AIML projects. Some problems that can be tackled by using synthetic data would be too costly or dangerous (e.g., in the case of training models controlling autonomous vehicles) to solve using more traditional methods, or simply cannot be done otherwise.

AIML projects run in different industries, and the multiple industry use cases that we include in this report are intended to give you a flavor of the broad applications of data synthesis. We define an AIML project quite broadly as well, to include, for example, the development of software applications that have AIML components.

The report is divided into four chapters. This introductory chapter covers basic concepts and presents the case for synthetic data. Chapter 2 presents the data synthesis process and pipelines, scaling implementation in the enterprise, and best practices. A series of industry-specific case studies follow in Chapter 3. Chapter 4 is forward-looking and considers where this technology is headed.

In this chapter, we start by defining the types of synthetic data. This is followed by a description of the benefits of using synthetic data—the types of problems that data synthesis can solve. Given the recent adoption of this approach into practice, building trust in analysis results from synthetic data is important. We therefore also present examples supporting the utility of synthetic data and discuss methods to build trust.

Alternatives to data synthesis exist, and we present these next with an assessment of strengths and weaknesses. This chapter then closes with an overview of methods for synthetic data generation.

What Is Synthetic Data?

At a conceptual level, *synthetic data* is not real data but is data that has been generated from real data and that has the same statistical properties as the real data. This means that an analyst who works with a synthetic dataset should get analysis results that are similar to those they would get with real data. The degree to which a synthetic dataset is an accurate proxy for real data is a measure of *utility*. Furthermore, we refer to the process of generating synthetic data as *synthesis*.

Data in this context can mean different things. For example, data can be *structured* data (i.e., rows and columns), as one would see in a relational database. Data can also be *unstructured* text, such as doctors' notes, transcripts of conversations among people or with digital assistants, or online interactions by email or chat. Furthermore, images, videos, audio, and virtual environments are also types of data that can be synthesized. We have seen examples of fake images in the machine learning literature; for instance, realistic faces of people who do not exist in the real world can be created, and you can view the results online.

Synthetic data is divided into two types, based on whether it is generated from actual datasets or not.

The first type *is* synthesized from real datasets. The analyst will have some real datasets and then build a model to capture the distributions and structure of that real data. Here, *structure* means the multivariate relationships and interactions in the data. Then the synthetic data is sampled or generated from that model. If the model is a good representation of the real data, the synthetic data will have similar statistical properties as the real data.

For example, a data science group specializing in understanding customer behaviors would need large amounts of data to build its models. But because of privacy or other concerns, the process for getting access to that customer data is slow and does not provide good enough data when it does arrive because of extensive masking and redaction of information. Instead, a synthetic version of the production datasets can be provided to the analysts for building their models. The synthesized data will have fewer constraints put on its use and would allow them to progress more rapidly.

The second type of synthetic data *is not* generated from real data. It is created by using existing models or by using background knowledge of the analyst. These existing models can be statistical models of a process (for example, developed through surveys or other data collection mechanisms) or they can be simulations. Simulations can be created, for instance, by gaming engines that create simulated (and synthetic) images of scenes or objects, or by simulation engines that generate shopper data with particular characteristics (say, age and gender) of people who walk past the site of a prospective store at different times of the day.

Background knowledge can be, for example, a model of how a financial market behaves based on textbook descriptions or based on the behaviors of stock prices under various historical conditions, or it can be knowledge of the statistical distribution of human traffic in a store based on years of experience. In such a case, it is relatively straightforward to create a model and sample from it to generate synthetic data. If the analyst's knowledge of the process is accurate, the synthetic data will behave in a manner that is consistent with real-world data. Of course, this works only when the phenomenon of interest is truly well understood.

As a final example, when a process is new or not well understood by the analyst and there is no real historical data to use, an analyst can make some simple assumptions about the distributions and correlations among the variables involved in the process. For example, the analyst can make a simplifying assumption that the variables have normal distributions and "medium" correlations among them, and create data that way. This type of data will likely not have the same properties as real data but can still be useful for some purposes, such as debugging an R data analysis program or for some types of performance testing of software applications.

For some use cases, having high utility will matter quite a bit. In other cases, medium or even low utility may be acceptable. For example, if the objective is to build AIML models to predict customer behavior and make marketing decisions based on that, high utility will be important. On the other hand, if the objective is to see if your software can handle a large volume of transactions, the data utility expectations will be considerably less. Therefore, understanding what data, models, simulators, and knowledge exist as well as the requirements for data utility will drive the specific approach to use for generating the synthetic data.

Table 1-1 provides a summary of the synthetic data types.

Table 1-1. Types of data synthesis with their utility and privacy implications

Type of synthetic data	Utility
Generated from real (nonpublic) datasets	Can be quite high
Generated from real public data	Can be high, although limitations exist because public data tends to be de-identified or aggregated
Generated from an existing model of a process, which can also be represented in a simulation engine	Will depend on the fidelity of the existing generating model
Based on analyst knowledge	Will depend on how well the analyst knows the domain and the complexity of the phenomenon
Generated from generic assumptions not specific to the phenomenon	Will likely be low

Now that you have an understanding of the types of synthetic data, we will look at the benefits of data synthesis overall and for some of these data types specifically.

The Benefits of Synthetic Data

In this section, we present several ways that data synthesis can solve practical problems with AIML projects. The benefits of synthetic data can be dramatic. It can make impossible projects doable, significantly accelerate AIML initiatives, or result in material improvement in the outcomes of AIML projects.

Improving Data Access

Data access is critical to AIML projects. The data is needed to train and validate models. More broadly, data is also needed for evaluating AIML technologies that have been developed by others, as well as for testing AIML software applications or applications that incorporate AIML models.

Typically, data is collected for a particular purpose with the consent of the individual; for example, for participating in a webinar or for participating in a clinical research study. If you want to use that same data for a different purpose, such as for building a model to predict what kind of person is likely to sign up for a webinar or who would participate in a study, then that is considered a *secondary purpose*.

Access to data for secondary analysis is becoming problematic. The US Government Accountability Office[1] and the McKinsey Global Institute[2] both note that accessing data for building and testing AIML models is a challenge for their adoption more broadly. A Deloitte analysis concluded that data access issues are ranked in the top three challenges faced by companies when implementing AI.[3] A recent survey from MIT Technology Review reported that almost

[1] Government Accountability Office, "Artificial Intelligence: Emerging Opportunities, Challenges, and Implications," GAO-18-142SP (March 2018). *https://oreil.ly/Cpyli*.

[2] McKinsey Global Institute, "Artificial Intelligence: The Next Digital Frontier?" (June 2017). *https://oreil.ly/zJ8oZ*.

[3] Deloitte Insights, "State of AI in the Enterprise, 2nd Edition" (2018). *https://oreil.ly/l07tJ*.

half of the respondents identified data availability as a constraint to the use of AI with their company.[4] At the same time, the public is getting uneasy about how their data is used and shared, and privacy laws are becoming more strict. A recent survey by O'Reilly highlighted the privacy concerns of companies adopting machine learning models, with more than half of companies experienced with AIML checking for privacy issues.[5] In the same MIT survey mentioned previously, 64% of respondents note that "changes in regulation or greater regulatory clarity on data sharing" is a development that would be most likely to lead to more data sharing.

Contemporary privacy regulations, such as the US Health Insurance Portability and Accountability Act (HIPAA) and the General Data Protection Regulation (GDPR) in Europe, impose constraints or requirements to using personal data for a secondary purpose. An example is a requirement to get an additional consent or authorization from individuals. In many cases, this is not practical and can introduce bias into the data because consenters and nonconsenters differ in important characteristics.[6]

Data synthesis can give the analyst, rather efficiently and at scale, realistic data to work with. Given that synthetic data would not be considered identifiable personal data, privacy regulations would not apply, and obligations of additional consent to use the data for secondary purposes would not be required.[7]

Improving Data Quality

Given the difficulty in getting access to data, many analysts try to just use open source or public datasets. These can be a good starting point, but they lack diversity and are often not well matched to the problems that the models are intended to solve. Furthermore, open

[4] MIT Technology Review Insights, "The Global AI Agenda: Promise, Reality, and a Future of Data Sharing" (March 2020). *https://oreil.ly/FHg87*

[5] Ben Lorica and Paco Nathan, *The State of Machine Learning Adoption in the Enterprise* (O'Reilly).

[6] Khaled El Emam, et al., "A Review of Evidence on Consent Bias in Research," *American Journal of Bioethics* 13, no. 4 (2013): 42–44. *https://oreil.ly/SiG2N*.

[7] However, one should follow good practices, such as providing notice to individuals about how the data is used and disclosed, and having ethics oversight on the uses of data and AIML models.

data may lack sufficient heterogeneity for robust training of models. For example, they may not capture rare cases well enough.

Sometimes the real data that exists is not labeled. Labeling a large number of examples for supervised learning tasks can be time-consuming, and manual labeling is error prone. Again, synthetic labeled data can be generated to accelerate model development. The synthesis process can ensure high accuracy in the labeling.

Using Synthetic Data for Exploratory Analysis

Analysts can use synthetic data models to validate their assumptions and demonstrate the kind of results that can be obtained with their models. In this way, the synthetic data can be used in an exploratory manner. Knowing that they have interesting and useful results, the analysts can then go through the more complex process of getting the real data (either raw or de-identified) to build the final versions of their models.

For example, an analyst who is a researcher could use their exploratory models on synthetic data to then apply for funding to get access to the real data, which may require a full protocol and multiple levels of approvals. In such an instance, work with synthetic data that does not produce good models or actionable results would still be beneficial because analysts would have avoided the extra effort required to get access to the real data for a potentially futile analysis.

Another valuable use of synthetic data is for training an initial model before the real data is accessible. Then when the analyst gets the real data, they can use the trained model as a starting point for training with the real data. This can significantly expedite the convergence of the real data model (hence reducing compute time), and can potentially result in a more accurate model. This is an example of using synthetic data for transfer learning.

Using Synthetic Data for Full Analysis

A validation server can be deployed to run the analysis code that worked on the synthetic data on the real data. An analyst would perform all of their analysis on the synthetic data, and then submit the code that worked on the synthetic data to a secure validation server that has access to the real data, as illustrated in Figure 1-1. Because the synthetic data would be structured in the same way as the original data, the code that worked on the synthetic data should work directly on the real data. The results are then sent back to the analyst to confirm their models.

This is not intended to be an interactive system. The output from the validation server needs to be checked to ensure that no revealing information is being sent out by the code output. Therefore, it is intended to be used once or twice by the analyst at the very end of their analysis. It does provide a way to provide assurance to the analysts that the synthesis results are replicable on the real data.

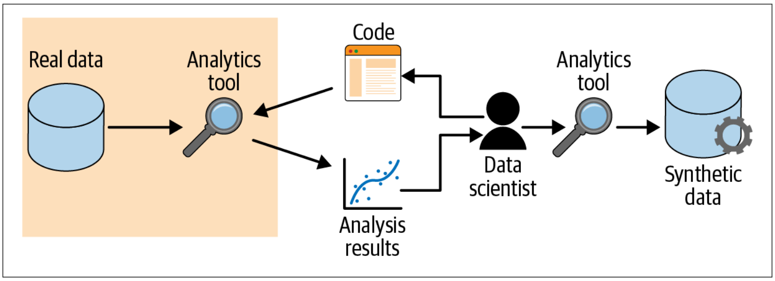


Figure 1-1. The setup for a validation server used to execute final code that produced results on the synthetic data (adapted from Replica Analytics Ltd., with permission)

When the utility of the synthetic data is high enough, the analysts can get similar results with the synthetic data as they would have with the real data, and no validation server is required. In such a case, the synthetic data plays the role of a proxy for the real data. This scenario is playing out in more and more use cases: as synthesis methods improve over time, this proxy outcome is going to become more common.

Replacing Real Data That Does Not Exist

In some situations, real data may not exist. The analyst may be trying to model something completely new, or the creation or collection of a real dataset from scratch may be cost prohibitive or impractical. Synthesized data can cover edge or rare cases that are difficult, impractical, or unethical to collect in the real world.

Synthetic data can also be used to increase the heterogeneity of a training dataset, which can result in a more robust AIML model. For example, unusual cases in which data does not exist or is difficult to collect can be synthesized and included in the training dataset. In that case, the utility of the synthetic data is measured in the robustness increment it gives to the AIML models.

We have seen that synthetic data can play a key role in solving a series of practical problems. One critical factor for the adoption of data synthesis, however, is trust in the generated data. It has long been recognized that high data utility will be needed for the broad adoption of data synthesis methods.[8] This is the topic we turn to next.

Learning to Trust Synthetic Data

Initial interest in synthetic data started in the early ’90s with proposals to use multiple imputation methods to generate synthetic data. Imputation in general is the process of replacing missing data values with estimates. Missing data can occur, for example, in a survey if some respondents do not complete a questionnaire.

Accurate imputed data requires the analyst to build a model of the phenomenon of interest by using the available data and then use that model to estimate what the imputed value should be. To build a valid imputation model, the analyst needs to know how the data will be eventually used. With multiple imputation, you create multiple imputed values to capture the uncertainty in these estimated values. This process can work reasonably well if you know how the data will be used.

In the context of using imputation for data synthesis, the real data is augmented with synthetic data by using the same type of imputation techniques. In such a case, the real data is used to build an imputation model that is then used to synthesize new data.

The challenge is that if your imputation models are different from the eventual uses of the data, the imputed values may not be very reflective of the real values, and this will introduce errors in the data. This risk of building the wrong synthesis model has led to historic caution in the application of synthetic data.

More recently, statistical machine learning models have been used for data synthesis. The advantage of these models is that they can capture the distributions and complex relationships among the

[8] Jerome P. Reiter, “New Approaches to Data Dissemination: A Glimpse into the Future (?),” *CHANCE* 17, no. 3 (June 2004): 11–15. *https://oreil.ly/x89Vd*.

variables quite well. In effect, they discover the underlying model in the data rather than having that model prespecified by the analyst. And now with deep learning data synthesis, these models can be quite accurate in that they can capture much of the signal in the data —even subtle signals.

Therefore, we are getting closer to the point where the generative models available today are producing datasets that are becoming quite good proxies for real data. There are also ways to assess the utility of synthetic data more objectively.

For example, we can compare the analysis results from synthetic data with the analysis results from the real data. If we do not know what analysis will be performed on the synthetic data, a range of possible analysis can be tried based on known examples of uses of that data. Or an "all models" evaluation can be performed in which all possible models are built from the real and synthetic datasets and compared.[9]

The US Census Bureau has, at the time of writing, decided to leverage synthetic data for some of its most heavily used public datasets, the 2020 decennial census data. For its tabular data disseminations, the agency will create a synthetic dataset from the collected individual-level census data and then produce the public tabulations from that synthetic dataset. A mixture of formal and nonformal methods will be used in the synthesis process.[10] We provide an overview of the synthesis process in Chapter 2. This, arguably, demonstrates the large-scale adoption of data synthesis for one of the most critical and heavily used datasets available today.

As organizations build trust in synthetic data, they will move from exploratory analysis use cases, to the use of a validation server, and then to using synthetic data as a proxy for real data.

A legitimate question is what are the other approaches that are available today to access data for AIML purposes, in addition to data

[9] A review of utility assessment approaches can be found in Khaled El Emam, "Seven Ways to Evaluate the Utility of Synthetic Data," *IEEE Security and Privacy* (July/August 2020).

[10] Aref N. Dajani, et al., "The Modernization of Statistical Disclosure Limitation at the U.S. Census Bureau," Census Scientific Advisory Committee Meeting (2017). *https://oreil.ly/OL4Oe*.

synthesis? We discuss these approaches, as well as their advantages and disadvantages relative to data synthesis in the following section.

Other Approaches to Accessing Data

When real data exists, two practical approaches, in addition to data synthesis, are available today that can be used to get access to the data. The first is de-identification. The second is secure multiparty computation.

Practical risk-based *de-identification* involves applying transformations to the data and putting in place additional controls (security, privacy, and contractual) to manage overall re-identification risks. A transformation can be, for example, generalizing a date of birth to a year of birth or a five-year range. Another transformation to data is to add noise to dates of events. Examples of controls include access controls to data and systems, performing background checks and training of analysts on privacy, and the use of encryption for data in transit and at rest. This process has worked well historically with clearly defined methodologies.[11]

As the complexity of datasets that are being analyzed increases, more emphasis is being put on the use of controls to manage the risk. The reason is that additional transformation would reduce the value of the data. Therefore, to ensure that the overall risk is acceptable, more controls are being put in place. This makes the economics of this kind of approach more challenging.

Data synthesis requires less manual intervention than de-identification, and there is no hard requirement for additional controls to be implemented by the synthetic data users.

The second approach that can be applied to get access to the data is to use *secure multiparty computation*. This technology allows computations to be performed on encrypted or garbled data; typically, multiple independent entities perform the computation collaboratively without sharing or leaking any raw data among themselves. There are multiple ways to do this, such as using *secret sharing techniques* (the data is randomly split among the collaborating entities)

[11] Khaled El Emam and Luk Arbuckle, *Anonymizing Health Data: Case Studies and Methods to Get You Started* (O'Reilly 2014).

or *homomorphic encryption techniques* (the data is encrypted, and computations are performed on the encrypted values).

In general, to use secure computation techniques, the analytics that will be applied need to be known in advance, and the security properties of each analysis protocol must be validated. A good example is in public health surveillance: the rate of infections in long-term care homes was aggregated without revealing any individual home's rate.[12] This works well in the surveillance case where the analysis is well defined and static, but setting up secure multiparty computation protocols in practice is complex.

Perhaps more of an issue is that few people understand the secure computation technology, the methods underlying many of these techniques, and can perform these security proofs. This creates key dependencies on very few skilled resources.

Once you have made a decision to generate and use synthetic data, you can turn to the next section for an overview of specific techniques to do so.

Generating Synthetic Data from Real Data

In this section, we consider methods for generating synthetic data from real data. Other approaches—for instance, using simulators—are discussed in Chapter 3 since they are more specific to the application domain.

At a general level, two classes of methods generate synthetic data from real data. Both have a generation component followed by a discrimination component. The *generation component* builds a model of the real data and generates synthetic data from that model. The *discrimination component* compares the generated data with the real data. If this comparison concludes that the generated data is very different from the real data, the generation parameters are adjusted and then new synthetic data is generated. The process iterates until acceptable synthetic data is produced.

[12] Khaled El Emam, et al., "Secure Surveillance of Antimicrobial Resistant Organism Colonization or Infection in Ontario Long Term Care Homes," *PLOS ONE* 9, no. 4 (2014). *https://oreil.ly/9dzJ4*.

An acceptable synthetic dataset is largely indistinguishable from the real data. However, we must be careful not to build a model that exactly replicates the original data. Such overfitting can create its own set of problems—the key problem being that the synthetic data can have nontrivial privacy problems.

The first approach to generating synthetic data is illustrated in Figure 1-2. Here the input to synthesis is real data. Various techniques can be used for the generator.

One set of techniques fits the distributions of all the variables in the real data (such as the type of distribution, the mean, and variance), and computes the correlations among the variables. With that information, it is then possible to sample synthetic data by using Monte Carlo simulation techniques while inducing the empirically observed correlations.

There are more advanced techniques that consider more complex interactions among the variables than just pairwise correlations (such as multiway interactions). For example, some studies have compared parametric, nonparametric, and artificial neural network techniques for data synthesis.[13] These empirical evaluations generate many synthetic datasets and evaluate the data utility of these to determine the extent to which the synthetic data produces analytics results that are comparable to the real data.

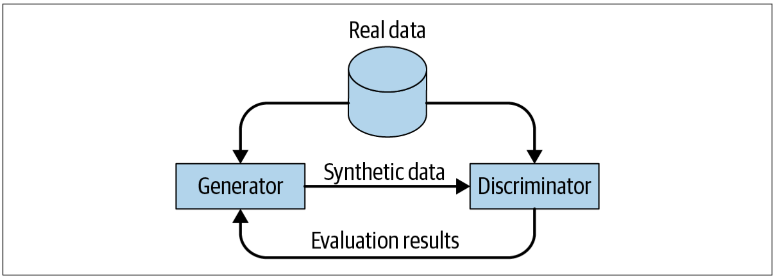


Figure 1-2. In this general scheme for generating data, the primary input is real data

[13] Jörg Drechsler and Jerome P. Reiter, “An Empirical Evaluation of Easily Implemented, Nonparametric Methods for Generating Synthetic Datasets,” *Computational Statistics & Data Analysis* 55, no. 12 (December 2011): 3232–3243. *https://oreil.ly/qHQK8;* Ashish Dandekar, et al., “A Comparative Study of Synthetic Dataset Generation Techniques,” National University of Singapore, TRA6/18 (June 2018). *https://oreil.ly/qLh0b*.

These evaluations have generally concluded that overall nonparametric statistical machine learning methods, such as decision trees, produce the best results. They are also simple to use and tune.

Deep learning synthesis techniques, such as autoencoders, have not been rigorously compared to nonparametric generators. However, they would be a good alternative to decision trees and can also work well in practice for data synthesis.

Other iterative techniques have been utilized, such as iterative proportional fitting (which is discussed in Chapter 3). These are suitable for certain types of real data, such as when the source consists of aggregate statistics rather than only individual-level or transactional data.

The second approach to generating synthetic data is illustrated in Figure 1-3. Here, instead of real data being the input to the generator, random data is provided as input. This is the configuration of generative adversarial networks and similar architectures. The model learns how to convert the random input into an acceptable synthetic dataset that passes the discriminator test.

Things start to get quite interesting when some of these methods are combined; for example, by creating ensembles to generate the synthetic data or by using the output of one method as the input to another method. An ensemble would have more than one data generation method, and, for example, would select the best synthesized records to be retained. Opportunities certainly exist for further experimentation and innovation in data synthesis methodologies.

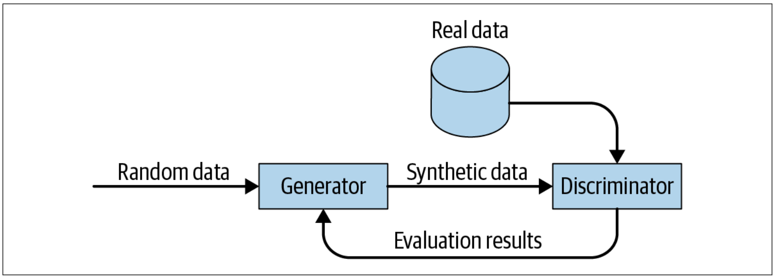


Figure 1-3. In this general scheme for generating data, the primary input is random data

Conclusions

This chapter provided an overview of what synthetic data is, its benefits, and how to generate it, as well as some of the trends driving the need for synthetic data. Both businesses and government alike are utilizing synthetic data, as you'll see in the use cases later in the report. In the next chapter, we look at the processes, data pipelines, and structure within an enterprise for data synthesis.

CHAPTER 2

The Synthesis Process

In the previous chapter, we defined the types of synthetic data, its benefits, and how to generate them. This chapter examines the practical implementation of data synthesis in the enterprise.

The implementation of data synthesis at the enterprise level has two key components: the process and the structure. The *process* consists of the key steps that indicate how to integrate synthesis into a data pipeline. The *structure* is typically operationalized through a Synthesis Center of Excellence. This would be a new entity within the organization that provides support throughout the enterprise in terms of process, technology, and governance for data synthesis implementations. This chapter describes the process and structure in some detail to provide guidance and present critical success factors.

In practice, the data synthesis capabilities described here may be deployed by large organizations as well as solo practitioners in many possible scenarios. Therefore, the following descriptions will need to be tailored to accommodate specific circumstances.

Data Synthesis Projects

Data synthesis projects have some processes that are focused on the generation of data and the validation of outputs, and other processes that prepare real data so that it can be synthesized. *Validation* includes both the evaluation of data utility and privacy assurance. In this section, we describe these processes and provide guidance on their application.

Data Synthesis Steps

A general data synthesis process is shown in Figure 2-1. This illustrates the complete process, although in certain situations and use cases not all of the steps would be needed. We will go through each of the steps next.

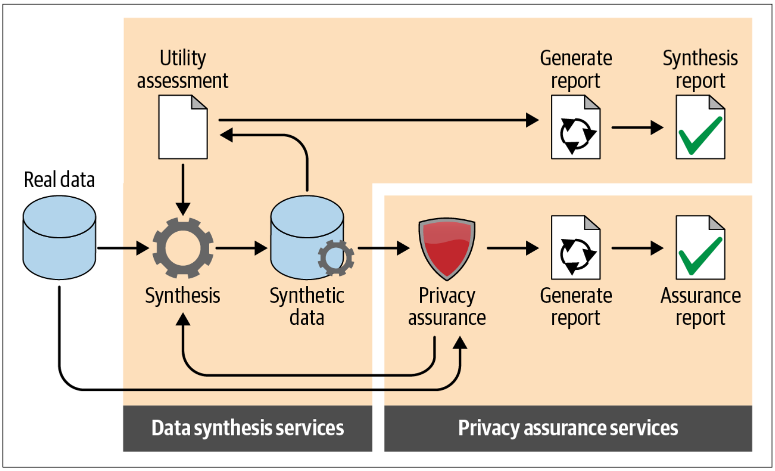


Figure 2-1. The overall data synthesis process (adapted from Replica Analytics Ltd., with permission)

When synthetic data is generated from real data, we need to start from the real data. The real data may be (a) individual-level datasets (or household-level datasets, depending on the context), (b) aggregate data with summaries and cross-tabulations characterizing the population, or (c) a combination of disaggregated and aggregate data. The real data may be open data or nonpublic data coming from production systems, for example.

The synthesis process itself can be performed using various techniques. We described some of these already in "Generating Synthetic Data from Real Data" on page 12, such as decision trees, deep learning techniques, and iterative proportional fitting. If real data does not exist, existing models or simulations can be used for data synthesis. The exact choice will be driven by the specific problem that needs to be solved and the level of data utility that is desired.

In many situations, a utility assessment needs to be done. This provides assurance to the data consumers that the data utility is acceptable, and helps with building trust in the synthesized data. These utility comparisons can be formalized using various similarity metrics so that they are repeatable and automated.

The utility assessment has two stages. The first stage consists of general-purpose comparisons of parameters calculated from the real and synthetic data; for example, comparisons of distributions and bivariate correlations. These act as a “smoke test” of the synthesis process. The second stage provides more workload-aware utility assessments.

Workload-aware utility assessments involve doing analyses on the synthetic data that are similar to the types of analyses that would be performed on the real data if that was available. For example, if the real data would be used to build multivariate prediction models, utility assessment would examine the relative accuracy of the prediction models built on synthetic datasets.

When the synthetic data pertains to individuals and potential privacy concerns exist, a privacy assurance assessment should also be performed. Privacy assurance evaluates the extent to which real people can be matched to records in the synthetic data and how easy it would be to learn something new if these matches were correct. Some frameworks have been developed to assess this risk empirically.

If the privacy assurance assessment demonstrates that the privacy risks are elevated, it would be necessary to revisit the synthesis process and change some of the parameters. For example, the stopping criterion for training the generative model may need to be adjusted because it was overfit, causing the synthetic records to be quite similar to the real records.

The utility assessment needs to be documented to provide the evidence that the level of utility is acceptable. Data analysts will likely want that utility confidence for the data that they are working on. And more importantly for compliance reasons, privacy assurance assessments must also be documented.

In practice, data generation would include utility assessment every time, and therefore they are bundled together as part of the Generate Report component in Figure 2-1. Privacy assurance can be performed across multiple synthesis projects since the results are expected to hold across similar datasets and would apply to the whole generation methodology. Hence that is bundled into a separate Privacy Assurance component in Figure 2-1.

The activities described here assume that the input real data is ready to be synthesized. In practice, data preparation will be required before real data can be synthesized. Data preparation is not unique to synthesis projects; however, it is an important step that we need to emphasize.

Data Preparation

When generating synthetic data from real data, as with any data analysis project that starts with real data, there will be a need for data preparation. This should be accounted for as part of the overall process.

Data preparation includes the following:

- Data cleaning to remove errors in the data.
- Data standardization to ensure that all of the fields are using consistent coding schemes.
- Data harmonization to ensure the data from multiple sources is mapped to the same data dictionary (for example, all the Age fields in the data, regardless of the field name and type, are recognized as an Age field).
- Linking of data from multiple sources. It is not possible to link synthetic data because the generated data does not match real people, so all linking has to happen in advance.

With data synthesis, the generated data will reflect any data quality challenges of the input data. Data analysis requires clean data, and it is easier to clean the data before the synthesis process. Messy data can distort the utility assessment process and cause convergence of the synthesis models to take longer. Furthermore, as we discuss in the next section with respect to pipelines, data synthesis may happen multiple times for the same real dataset, and therefore it is much easier to have data quality issues addressed once up front *before synthesis*.

Real data will have certain deterministic characteristics, such as structural zeros (these are zero values in the data that must be zero because it wouldn't make sense for them to be nonzero; i.e., the zero is not a data collection artifact). For example, five-year-olds cannot have PhDs and get pregnant, so the "pregnancy?" count for someone who is five will always be zero. Also body mass index (BMI) is a deterministic derived calculation from height and weight. This means that there is no uncertainty in deriving the BMI from height and weight. The data synthesis process needs to capture these characteristics and address them. They can either be specified a priori as a series of rules to be satisfied or as edits applied to the synthetic data after the fact. This way, the synthesized data will maintain high logical consistency.

A key consideration when implementing data synthesis is how to integrate it within a data architecture or pipeline. In the next section, we address this issue and provide some common pipelines.

The Data Synthesis Pipeline

Understanding the data flows that are bringing in data to analysts for their AIML projects is important for deciding where data preparation and data synthesis should be implemented in that data flow. It is easiest to explain this through a few examples. All of these examples represent actual situations that we have seen in a variety of industries (such as health care and financial services).

One relatively noncomplex setting is a single production dataset or a single data source. In that case, the data flows are simple, as illustrated in Figure 2-2. The analysts receiving the synthetic data can then work on that data internally or share it with external parties.

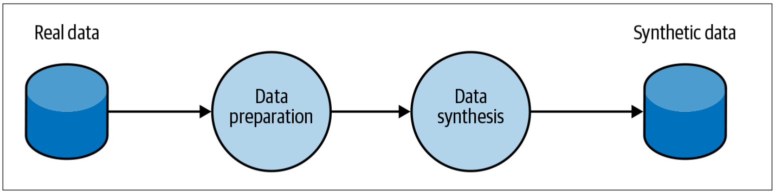


Figure 2-2. Synthesizing data from a production environment

In a more complex situation, the data source is in a different organization. For example, the data may be coming from a financial institution to an analytics consultancy or analytics vendor. This is illustrated in the data flows in Figure 2-3.

Under these data flows, the data analysts/data consumers are not performing the data synthesis because they do not have authority or the controls to process the real data (which may be, for example, personally identifying financial information). Under contemporary data protection regulations, such as the European GDPR, the obligations and risks to process personally identifying information are not trivial. Therefore, if the data analyst/data consumer can avoid these obligations by having the data supplier or a trusted third party perform the data synthesis, that would be preferable.

This data flow has three common scenarios. In scenario (a), the data preparation and data synthesis both happen at the data supplier. In scenario (b), a trusted third party performs both tasks. In scenario (c), the data supplier performs the data preparation, and the trusted third party performs the data synthesis. In this context, a trusted third party would be an independent entity that has the authority and controls in place to process the real data on behalf of the data supplier.

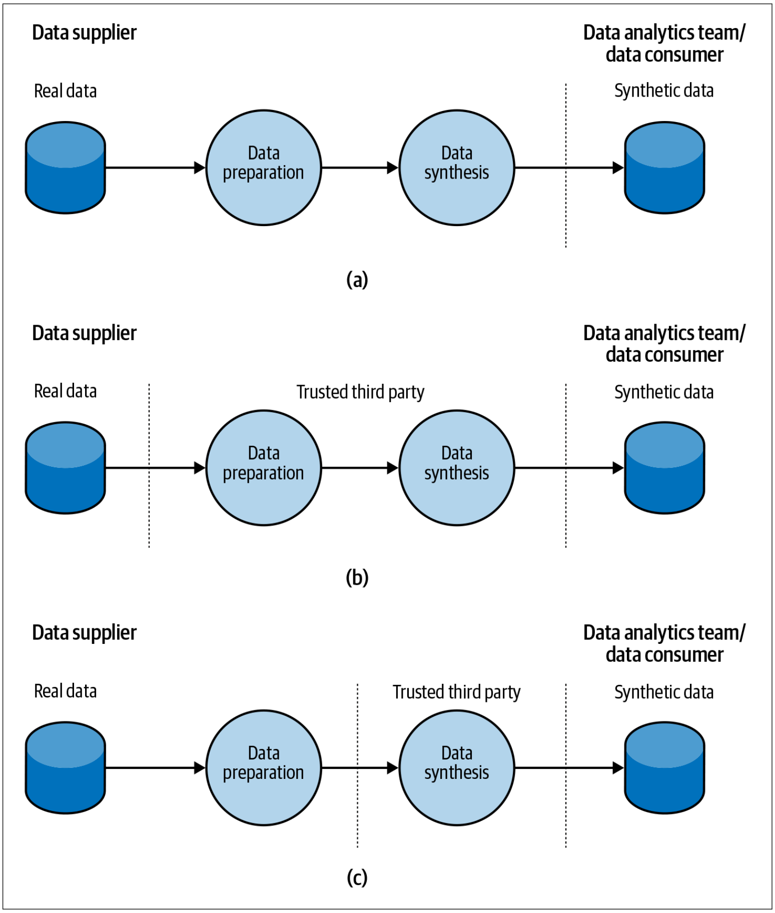


Figure 2-3. Synthesizing data coming from an external data supplier

The last set of examples of data flows that we will look at has many data sources. These are extensions of the examples in Figure 2-3. In the first data flow shown in Figure 2-4, the data is synthesized at the source by each of multiple data suppliers. For example, the suppliers may be different banks or different pharmacies sending the synthesized data to an analytics company to be pooled and to build models on. Or a medical software developer may be collecting data centrally from all of their deployed customers, with the synthesis performed at the source within their software. Once the synthesized data reaches the data analysts, they can build AIML models without the security and privacy obligations of working with real data.

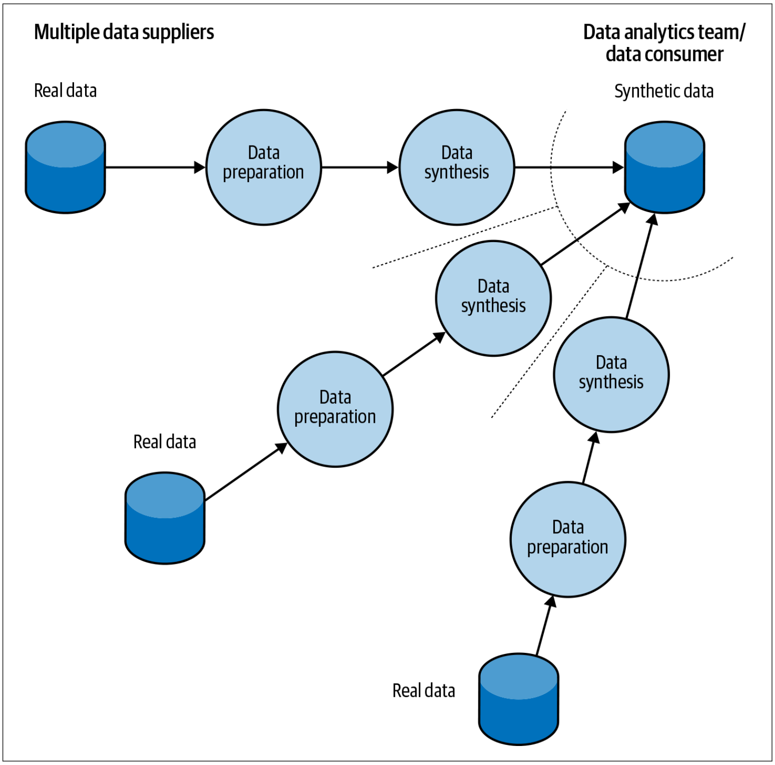


Figure 2-4. Synthesizing data from multiple external data suppliers

Another data flow with multiple data sources involves using a trusted third party who prepares and synthesizes the data on behalf of all of them. The synthesis may be performed on each individual data supplier's data, or the data may be pooled first and then the synthesis performed on the pooled data. The exact setup will depend on the characteristics of the data and the intervals that the data are arriving at the third party. This is illustrated in Figure 2-5.

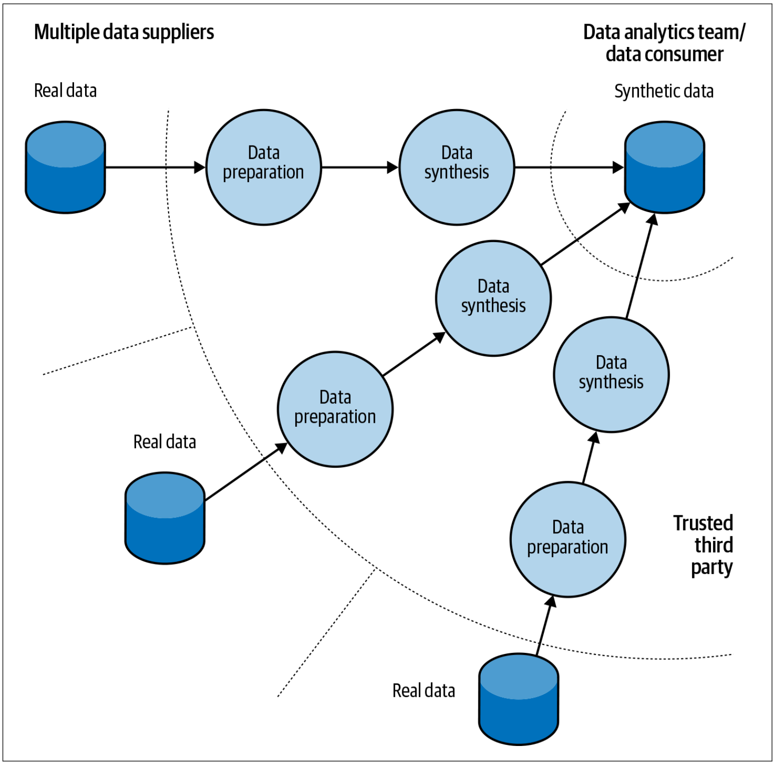


Figure 2-5. Synthesizing data coming from multiple external data suppliers going through a single trusted third party that performs data preparation and synthesis

The final data flow that we will consider is a variant of the one we examined earlier. Here, the data preparation is performed at the data source before being sent to the trusted third party, as illustrated in Figure 2-6.

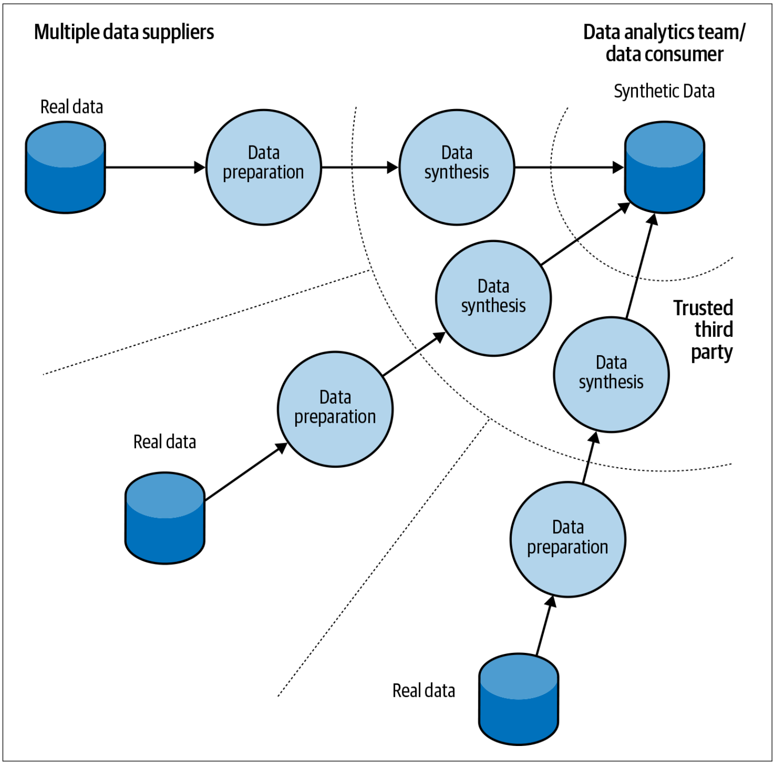


Figure 2-6. Synthesizing data from multiple external data suppliers going through a single trusted third party that performs only synthesis

The exact data flow used in a particular situation will depend on multiple factors: (a) the number of data sources, (b) the costs and readiness of the data analyst/data consumer to process real data and meet any regulatory obligations, (c) the availability of qualified trusted third parties to perform these tasks, and (d) the ability of data suppliers to implement automated data preparation and data synthesis processes. In this section, we have provided a set of common pipelines that can be implemented, given the combinations of the preceding factors.

In large organizations, data synthesis needs to be part of a broader structure that is scalable and that can serve multiple business units and client needs. We present the concept of program management, which supports such scalability, in the next section.

Synthesis Program Management

As data synthesis becomes a core part of an organization's data pipeline, an enterprise-wide structure is needed to ensure that the activities are repeatable and scalable. *Scale* here can mean data synthesis being used by multiple internal business units or as a capability used by multiple clients. This can be supported at a programmatic level by a center of excellence (CoE).

A *Synthesis CoE* is a mechanism that allows an organization to centralize expertise and technology for the generation of synthetic data. In large organizations, such centralization is beneficial because it ensures learning over time (a shorter feedback loop), standardizes methodologies across projects and datasets, and provides some economies of scale with respect to the technologies and computational capacity that may be needed.

The CoE can serve a single organization or, for that matter, can serve a consortium of companies operating in the same space. The end users of the synthetic data can be internal, or the CoE can support clients in implementing, say, analytics tools by making appropriate synthetic data available to them.

Those operating the CoE need both technical skills, to generate synthetic data and perform privacy assurance, and business analysis skills, to be able to understand user requirements and translate those into synthesis specifications. More importantly, change management is key since transitioning analysts to using synthetic data will require them to provide some education and possibly a series of utility assessments.

CoE for an Analytics Service Provider

ConsultingCo provides management consulting services to a broad spectrum of clients. Some years ago, the company created a data analytics business that supports clients by helping them build data analysis capacity (for example, finding, organizing, and cleaning the data and building AIML models to inform the business lines) and to do actual model building for them. One of the big challenges was getting data early in the process.

Early on in these engagements, the clients often did not have a full accounting of their data assets or the quality of that data. There were also questions about the lawful basis for performing secondary analysis on that data. Complicating matters was the internal reluctance by business lines to share data or to invest in making data available for analytics before the value of the analytics was demonstrated.

The data synthesis team at ConsultingCo provides synthetic data early in these engagements to enable analysts to demonstrate the value of using the data that is available to the clients and how models that can be built would inform business decisions. The synthetic data can be generated without real data or based on small samples of real data.

The ability to demonstrate value early in the process greatly facilitates getting buy-in for acquiring, cleaning, and using the data within the organization. The synthesis CoE gives ConsultingCo a competitive advantage in that the likelihood of success of these engagements increases.

Data synthesis will be a new methodology for many organizations. While the introduction of any data analytics method and technology involves organizational change, data synthesis introduces specific considerations during implementation. In the next section, best practices for the implementation of data synthesis are discussed to help increase your likelihood of a smooth adoption of this approach.

Best Practices for Implementing Data Synthesis

The success of a synthetic data generation project depends on a set of technical and change management factors. *Change management* is used here to refer to the activities that are needed to support the analysts and analytics leadership in changing their practices to embed the use of synthetic data into their work. The practices we cover in this section can have an oversized influence on the outcome of implementing data synthesis.

While the amount of manual effort to synthesize data is relatively small, many data synthesis methods are computationally intensive. Therefore, we first discuss the importance of computing capacity. We next consider the situation in which analysts need to work with only cohorts rather than full datasets. The section closes with a discussion of the importance of validation studies, initially and continuously, to get and maintain buy-in of data analysts and data users.

Having Sufficient Computing Capacity

One of the critical requirements for synthetic data generation is computing capacity. This is especially true for large datasets, with many variables and many transactions. In practice, synthesis models require tuning, and being able to efficiently iterate can have a nontrivial impact on the speed of getting the data ready for subsequent uses.

Even relatively straightforward decision tree data synthesis methods can be computationally demanding. For example, most classification decision tree algorithms will test all possible combinations when performing their tree splits. This becomes a significant combinatorial problem for variables with hundreds of categories, such as those often seen in health datasets.

Moving to deep learning models for data synthesis, the computing needs to build generative models for large datasets is not trivial. For example, large structured data can have many variables or many records—both demand computing capacity. The generation of a large number of heterogeneous images through simulations or virtual environments also can require significant computation.

The next section discusses data quality and where the management of data quality issues falls within the data synthesis process.

Synthesizing Cohorts Versus Full Datasets

As a practical matter, many data analyses and AIML models are performed or developed respectively on specific cohorts, or subsets of the full dataset. For example, only a subset of consumers within a specific age range may be of interest, or the analysis may be performed on only a subset of the variables. Then that cohort is extracted from the master dataset and sent to the analysts.

For data synthesis, it is much easier to synthesize the full dataset rather than synthesize each cohort as it is extracted. The advantage over synthesizing individual cohorts is that the synthesis would be done once, rather than every time a data extract is needed.

Given this argument, it is recommended that data be synthesized as it is coming in rather than as it is going out. For example, if an organization has a data lake and is extracting cohorts from that for specific analyses, the data synthesis should be performed when the data is going into the data lake such that the data lake consists of only synthetic data.

The final best practice that we cover in the next section pertains to how to build trust and get buy-in for the deployment of synthetic data on a sustainable basis.

Performing Validation Studies to Get Buy-In

Perhaps the key factor in the success of data synthesis projects is getting the buy-in of the data users and data analysts. In many instances, the use of synthetic data is new for data analysts, for example. Including validation steps in the process of deploying data synthesis will be important, and we have included that explicitly in the process of Figure 2-1. *Validation* means that case studies are performed to demonstrate the utility of the synthetic data for the task at hand. Even if case studies exist in other organizations, demonstrations on their own data can be much more impactful for the data analysts using the synthetic data.

A validation means showing that the results from the synthetic data are similar to the results from the real data. The extent of the similarity will depend on the specific use case. For example, if the use case is to use synthetic data for software testing, the criteria for similarity would be less stringent than if the data will be used to build an AIML model to identify high-risk insurance claims.

Such validation studies should be chosen to be representative of the datasets and situations that are likely going to be encountered in practice. Choosing the most challenging dataset or context for a validation is not going to be very informative and increases the chances of unsuccessful outcomes. Going in the other direction and choosing the simplest scenarios may not be convincing for the eventual users of the synthetic data.

Conclusions

The adoption of synthetic data generation has been rapid over the last couple of years. This is an indicator that, as a technology, it does solve a real problem. The synthesis methods are evolving quite rapidly and are being applied to increasingly complex datasets. It is therefore recommended that applied practitioners in this area keep abreast of research work in the field. Unfortunately, the community is dispersed among multiple disciplines and not centralized. This makes the case for dedicated expertise to monitor developments in this area in a CoE, to consolidate and disseminate them in a manner that practicing analysts can use.

The deployment of data synthesis methods in an organization can be scaled from discrete projects to a continuous synthesis flow. In all cases, an understanding of the data flow and the actors in it helps determine where synthesis should be implemented. More generally, we have covered implementation best practices in this chapter related to having enough computing capacity, dealing with cohorts, and generating buy-in. While these are not an exhaustive list of practices that need to be executed correctly, they are critical for the success of a data synthesis implementation effort.

CHAPTER 3

Synthetic Data Case Studies

While the technical concepts behind the generation of synthetic data have been around for a few decades, their practical use has picked up only quite recently. One reason is that this type of data solves some challenging problems that were quite hard to solve before, or solves them in a more cost-effective way. All of these problems pertain to data access: sometimes it is just hard to get access to real data.

In this chapter, we present a few application examples from various industries. These examples are not intended to be exhaustive but rather illustrative. Also, the same problem may exist in multiple industries (for example, getting realistic data for software testing is a common problem that data synthesis can solve), and the applications of synthetic data to solve that problem will therefore be relevant in these multiple industries. Because we discuss software testing, say, under only one heading does not mean that it would not be relevant in another.

The first industry that we will examine is manufacturing and distribution. We then give examples from health care, financial services, and transportation. The industry examples span the types of synthetic data we discussed, from generating structured data from real individual-level and aggregate data, to using simulation engines to generate large volumes of synthetic data.

Manufacturing and Distribution

The use of AIML in industrial robots, coupled with improved sensor technology, is further enabling factory automation for more complex and varied tasks.[1] In the warehouse and on the factory floor, these systems are increasingly able to pick up arbitrary objects off shelves and conveyor belts, and to inspect, manipulate, and move them, as illustrated, for example, by the Amazon Picking Challenge.[2]

However, robust training of robots to perform complex tasks in the production line or warehouse can be challenging because of the need to obtain realistic training data covering multiple anticipated scenarios, as well as uncommon ones that are rarely seen in practice but are still plausible. For example, recognizing objects under different lighting conditions, with different textures, and in various positions requires training data that captures the variety and combinations of these situations. It is not trivial to generate such a training dataset.

Let's consider an illustrative example of how data synthesis can be used to train a robot to perform a complex task that requires a large dataset for training. Engineers at NVIDIA were trying to train a robot to play dominoes by using a deep learning model (see Figure 3-1). The training needed many heterogeneous images that capture the spectrum of situations that a robot may encounter in practice. Such a training dataset did not exist, and it would have been cost prohibitive and time-consuming to manually try to create these images.

[1] Jonathan Tilley, "Automation, Robotics, and the Factory of the Future," McKinsey (September 2017). *https://oreil.ly/qWTfW*.

[2] Lori Cameron, "Deep Learning: Our No. 1 Tech Trend for 2018 Is Set to Revolutionize Industrial Robotics," IEEE Computer Society. *https://oreil.ly/FeOkW*.

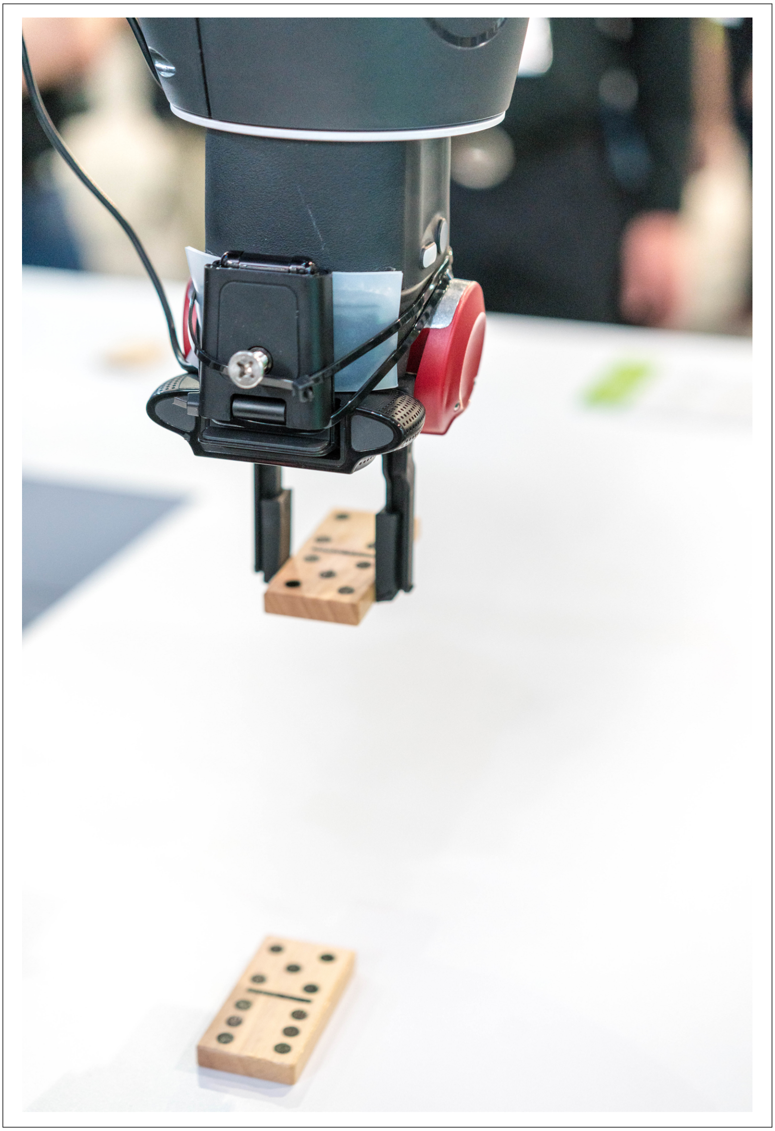


Figure 3-1. The domino-playing robot

The NVIDIA team used a graphics rendering engine from its gaming platform to create images of dominoes in different positions, with different textures, and under different lighting conditions (see

Figure 3-2).[3] No one actually manually set up dominoes and took pictures of them to train the model; the images that were created for training were simulated by the engine.

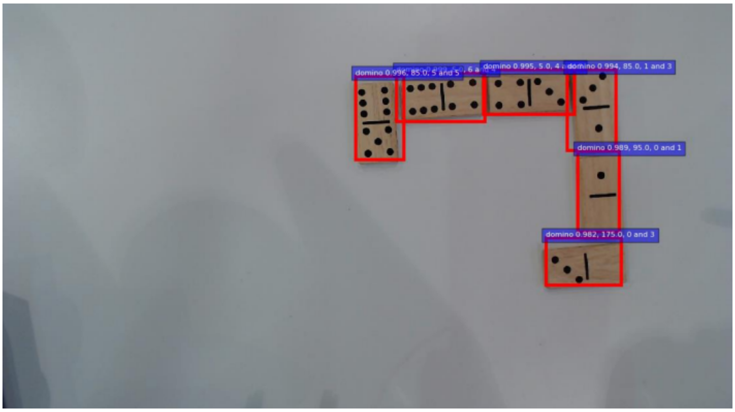

domino 0.996, 85.0, 5 and 5
domino 0.999, 5.0, 6 and 4
domino 0.995, 5.0, 4 a
domino 0.994, 85.0, 1 and 3
domino 0.989, 95.0, 0 and 1
domino 0.982, 175.0, 0 and 3

Figure 3-2. An example of a synthesized domino image

In this case, the image data did not exist. Creating a large enough dataset manually would have taken a lot of people a long time—a not very cost-effective option. The team used the simulation engine to create hundreds of thousands of images to train the robot. This is a good example of how synthetic data was used to train a robot to recognize, pick up, and manipulate objects in a heterogeneous environment—the same type of model building that would be needed for industrial robots.

Health Care

Getting access to data for building AIML models in the health industry is often difficult because of privacy regulations or because the data collection can be expensive. Health data is considered sensitive in many data protection regimes, and its use and disclosure for analytics purposes must meet multiple conditions. These conditions can be nontrivial to put in place (e.g., providing patients access to their own data, strong security controls around the retention and

[3] Rev Lebaredian, "Synthetic Data Will Drive the Next Wave of Business Applications," *GTC Silicon Valley* (2019). *https://oreil.ly/tGlWt*.

processing of the data, staff training, and others).[4] Also, the collection of health data anew for specific studies or analyses can be quite expensive. For instance, the collection of data from multiple sites in clinical trials is costly.

The following examples illustrate how synthetic data has solved the data access challenge in the health industry.

Data for Cancer Research

There are strong currents pushing governments and the pharmaceutical industry to make their health data more broadly available for secondary analysis. This is intended to solve the data access problem and encourage more innovative research to understand diseases and find treatments.

Regulators have required companies to make health data more broadly available. A good example is the European Medicines Agency, which has required pharmaceutical companies to make the information that they submit for their drug approval decisions publicly available.[5] Health Canada has also recently done so.[6]

In addition, medical journals are now strongly encouraging researchers who publish articles to make their data publicly available for other researchers to replicate the studies, and possibly lead to innovative analyses on that same data.[7]

In general, when that data contains personal information, it needs to be *de-identified*, or made nonpersonal, before it is made public (unless consent is obtained from the affected individuals beforehand, which is not the case here). However, in practice, it is difficult

[4] Mike Hintze and Khaled El Emam, "Comparing the Benefits of Pseudonymisation and Anonymisation Under the GDPR," *Journal of Data Protection & Privacy* 2, no. 1 (December 2018): 145–158. *https://oreil.ly/7g4DP*.

[5] European Medicines Agency, "External Guidance on the Implementation of the European Medicines Agency Policy on the Publication of Clinical Data for Medicinal Products for Human Use" (September 2017). *https://oreil.ly/ReUFR*.

[6] Health Canada, "Guidance Document on Public Release of Clinical Information," Government of Canada (April 1, 2019). *https://oreil.ly/Cun4r*.

[7] International Committee of Medical Journal Editors, "Data Sharing Statements for Clinical Trials: A Requirement of the International Committee of Medical Journal Editors," *The Lancet* (June 2017). *https://oreil.ly/T7D1I*.

to de-identify complex data for a public release.[8] There are a few reasons for this:

- Public data has few controls on it (e.g., the data users do not need to agree to terms of use and they do not need to reveal their identity, making it difficult to ensure that they are handling it securely). Therefore, the level of data transformations needed to ensure that the risk of re-identification is low can be extensive, ensuring that data utility has degraded significantly.
- Re-identification attacks on public data are getting more attention by the media and regulators, and are also getting more sophisticated. As a consequence, de-identification methods need to err on the conservative side, further eroding data utility.
- The complexity of datasets that need to be shared further amplifies the data utility problems because a lot of the information in the data would need to be transformed to manage the re-identification risk.

Synthetic data makes it feasible to have complex open data. *Complexity* here means the data has many variables and tables, with many transactions per individual. For example, data from an oncology electronic medical record would be considered complex. It would have information about, for instance, the patient, visits, treatments, drugs prescribed and administered, and laboratory tests.

Synthesis would simultaneously address the privacy problem and provide data that is of higher utility than the incumbent alternative. A good example is the synthetic cancer registry data that has been made publicly available by Public Health England. This synthetic cancer dataset is available for download and can be used to generate and test hypotheses and to do cost-effective and rapid feasibility evaluations for future cancer studies.

Beyond data for research, a digital revolution is happening in medicine.[9] For example, the large amounts of health data that exist with providers and payers contain many insights that can be detected by

[8] Khaled El Emam, "A De-identification Protocol for Open Data", IAPP Privacy Tech (2016). *https://oreil.ly/ZtEXe*.

[9] Neal Batra, et al., "The Future of Health," Deloitte Insights (April 2019). *https://oreil.ly/86nE1*.

the more powerful AIML techniques. New digital medical devices are adding more continuous data about patient health and behavior. Patient-reported outcome data provides patient assessments of function, quality of life, and pain. And, of course, genomic and other -omic data is at the core of personalized medicine. All this data needs to be integrated and converted into decisions and treatments that can be provided at the point of care. Innovations in AIML can be a facilitator of that.

In the next section, we examine how digital health and health technology companies can use synthetic data to tap into this innovation ecosystem. And note that, increasingly, drug and device companies are becoming digital health companies.

Evaluating Innovative Digital Health Technologies

Health technology companies are constantly looking for data-driven innovations coming from the outside. These can be innovations from start-up companies or from academic institutions. Typical examples include data analysis (statistical machine learning or deep learning models and tools), data wrangling (such as data standardization and harmonization tools and data cleaning tools), and data type detection tools (finding out where different types of data exist in the organization).

Because the adoption of new technologies takes resources and has opportunity cost, the decision to do so must be made somewhat carefully. These companies need a mechanism to evaluate these innovations in an efficient way to determine which ones really work in practice and, more importantly, which ones will work with their data. The best way to do that is to give these innovators some data and have them demonstrate their wares on that data.

Some large companies get approached by innovators at a significant pace—sometimes approaching multiple parts of an organization at the same time. The pitches are compelling, and the potential benefits to their business can be significant. They want to bring these innovations into their organizations. But experience has told them that, for instance, some of the start-ups are pitching ideas rather than mature products, and the academics are describing solutions that worked on only small problems or in situations unlike theirs. There is a need to test these innovations on their own problems and data.

In the pharmaceutical industry, providing data to external parties can be complex because much of the relevant data pertains to patients or health care providers. The processes needed to share that data would usually require extensive contracting and an audit of the security practices at the data recipient. Just these two tasks could take quite some time and investment.

Sometimes the pharmaceutical company is unable to share its data externally because of this complexity or of internal policies, and therefore asks the innovator to come in and install the software in the pharmaceutical company's environment (for an example, see the following sidebar). This creates significant complexity and delays because now the pharmaceutical company needs to audit the software, address compatibility issues, and figure out integration points. This makes technology evaluations quite expensive and uses up a lot of internal resources. In addition, this is not scalable to the (potentially) hundreds of innovations that the pharmaceutical company would want to test every year.

These companies have started to do two things to make this process more efficient and to enable them to bring in innovations. First, they have a standard set of synthetic datasets that are representative of their patient or provider data. For example, a pharmaceutical company would have a set of synthetic clinical trial datasets in various therapeutic areas. These datasets can be readily shared with innovators for pilots or quick proof-of-concept projects.

Rapid Technology Evaluation

Cambridge Semantics (CS), a Boston company developing a graph database and various analytics tools on top of that, was planning to do a pilot with a large prospect in the health space to demonstrate how its tools can be used to harmonize pooled clinical trial data. To be able to do this pilot, CS needed to get data from the prospect. That way, the company could demonstrate that its tools worked on real data that was relevant for the prospect—few things are more compelling than seeing a problem solved in an elegant way on your own data.

The initial challenge was that getting data from the prospect meant that CS would need to go through an audit to ensure that it had adequate security and privacy practices to handle personal health information. That process would have taken three to four months to complete.

An alternative that was considered was for CS to install its software on the prospect's private cloud and then to run it there using real data. However, the complexities of introducing new software into a regulated computing environment are not trivial. Furthermore, giving CS staff access to the internal computing environment would have required additional checks and processes. This also would have taken three to four months.

The team landed on a synthetic data solution whereby multiple synthetic datasets were created and given to CS to demonstrate how it would solve the specific problem. The pilot was completed in a few days. At the time of writing, CS was close to closing the deal.

The second process that is used is to run competitions. The basic idea is to define a problem that needs to be solved. Then invite innovators to solve that problem and provide entrants with synthetic data to demonstrate their solutions. These can be open or closed competitions. With the former, any start-up, individual, or institution can participate, such as by organizing public hackathons or datathons. With the latter, specific ones are invited to participate in the competition.

With public hackathons or datathons, entrants are invited to solve a given problem with a prize at the end for the winning individual or team. The main difference is that the innovators are not selected in advance, but rather participation tends to be more open. The diversity in these competitions means that many new ideas are generated and evaluated in a relatively short period of time. Synthetic data can be a key enabler under these circumstances by providing datasets that the entrants can access with minimal constraints.

A good example of an open competition was the Heritage Health Prize (HHP). The HHP was notable for the size of the prize and the size of the dataset that was made available to entrants. At that time, the availability of synthetic data was limited, and therefore a de-identified dataset was created.[10] Because of the challenges of de-

[10] Khaled El Emam, et al., "De-identification Methods for Open Health Data: The Case of the Heritage Health Prize Claims Dataset," *Journal of Medical Internet Research* 14, no. 1 (February 2012): e33. *https://oreil.ly/H7g8y*.

identifying open datasets noted earlier, it has been more common for health-related competitions to be closed. However, at the time of writing there is no compelling reason to maintain that restriction. Synthetic data is now being used to enable such competitions, as described in "Datathons Enabled by Synthetic Data" on page 42.

In practice, only a small percentage of innovators succeed when given a realistic dataset to work with. Those who make it through the evaluation or competition are then invited to go through the more involved process to get access to real data and do more detailed demonstrations, or the company may decide to license the innovation at that point. But at least the more costly investments in the technology evaluation are performed only on candidates that are known to have an innovation that works.

Datathons Enabled by Synthetic Data

The Vivli Microsoft Data Challenge was held in June 2019 in Boston. The goal of the competition was to propose innovative methods to facilitate the sharing of rare disease datasets, in a manner that maintains the analytic value of the data while safeguarding participant privacy. Rare disease datasets are particularly difficult to share while maintaining participant privacy, as these datasets often contain relatively few individuals, and individuals may be uniquely identified using only a handful of attributes.

This event gathered 60 participants on 11 teams from universities, hospitals, and pharmaceutical, biotech, and software companies. Each team had five hours to plan and propose a solution, then five minutes to present the solution to the judges. The solutions developed combined new and existing technologies in interesting ways tailored for use in rare disease datasets. Unsurprisingly, the winning team proposed a solution built around the use of synthetic data.

Synthetic data was critical to this event's success, as it allowed all participants to "get their hands dirty" with realistic clinical trial data, without needing to use costly secure computational environments or other control mechanisms. The synthetic data grounded the competition in reality by providing participants with example data that their solutions would need to be able to accommodate. Groups that built demos of their solutions were also able to apply their methods to the synthetic data as a proof of concept.

Data challenges like this are dependent on providing high-quality data to participants, and synthetic data is a practical means to do so.

Another large consumer of synthetic data is the financial services industry. Part of the reason is that this industry has been an early user of AIML technology and data-driven decision-making, such as in fraud detection, claims processing, and consumer marketing. In the next section, we examine specific use cases where synthetic data has been applied in this sector.

Financial Services

Getting access to large volumes of historical market data in the financial services industry can be expensive. These types of data are needed, for example, for building models to drive trading decisions and for software testing. Also, using consumer financial transaction data for model building (say, in the context of marketing retail banking services) is not always easy because that requires the sharing of personal financial information with internal and external data analysts.

The following use cases illustrate how synthetic data has been used to solve some of these challenges.

Synthetic Data Benchmarks

When selecting software and hardware to process large volumes of data, financial services companies need to evaluate vendors and solutions in the market. Instead of having each company evaluate technologies from innovative vendors and academics one by one, it is common to create standardized data benchmarks.

A *data benchmark* consists of a dataset and a set of tests that will be performed on that dataset. Vendors and academics can then use their software and hardware to produce the outputs by using this data as inputs, and they can all be compared in a consistent manner. Creating a benchmark would make the most sense in situations where the market is large enough and the community can agree on a benchmark that is representative.

In competitive scenarios in which multiple vendors and academics can supply solutions to the same set of problems, the benchmarks must be constructed in a manner that ensures that no one can easily game the system. With a standard input dataset, the solutions can just be trained or configured to produce the correct output without performing the necessary analytic computations.

Synthetic data benchmarks are produced from the same underlying model, but each vendor or academic gets a unique and specific set of synthetic data generated from that model. In that way, each entity running the benchmark will need to produce different results to score well on the benchmark.

An example of that is the STAC-A2 benchmark for evaluating software and hardware used to model financial market risk. The benchmark has quality measures in the output that are assessed during the computation of option price sensitives for multiple assets using Monte Carlo simulation. A series of performance/scaling tests are also performed using the data.

When financial services companies wish to select a technology vendor, they can compare the solutions on the market by using a consistent benchmark that was executed on comparable data. This provides a neutral assessment of the strengths and weaknesses of available offerings without having to perform their own evaluations (which can be expensive and time-consuming), or relying on vendor-specific assessments (which may be biased toward that vendor).

Software Testing

Software testing is a classic use case for synthetic data. This includes functional and performance testing of software applications by the software developers. In some cases, large datasets are needed to benchmark software applications to ensure that they can perform at certain throughputs or with certain volumes. Extensions of the testing use case are datasets for running software demos by a sales team and for training users of software on realistic data.

Software testing is common across many industries, and the problems being addressed with synthetic data will be the same. The financial services sector provides two common use cases. The first is to test internal software applications (e.g., fraud detection) to ensure that they perform the intended functions and do not have bugs. To do so, realistic input data is needed, and this includes data covering edge cases or unusual combinations of inputs. The second is to test that these applications can scale their performance (for example, response times in automated trading applications are important) to handle large volumes of data that are likely to be met in practice. This testing must also simulate unusual situations; for example, when trading volumes spike because of an external political or environmental event.

In most software engineering groups, obtaining production data is not easy. This may be because of privacy concerns or the data contains client confidential business information. Therefore, there is reluctance to make that available to a large group of software developers. The same applies to making data available for demos and for training purposes. Furthermore, in some cases, the software is new and there is insufficient customer data to use for testing.

One alternative that has been used is to de-identify the production data before making it available to test teams. Because the need for test data is continuous, the de-identification must also be performed on a continuous basis. The cost-effectiveness of continuous de-identification versus synthetic data would have to be considered.

The data utility demands for software testing are not as high as they are for some of the other use cases that we looked at. It is possible to generate synthetic data from theoretical distributions and then use it for testing. Another approach that has been applied is to use public datasets (open data) and replicate those multiple times to create larger test datasets or resample with replacement.

More principled methods exist for the generation of synthetic data for testing, demos, and training. These involve the generation of synthetic data from real data by using the same approaches that are used to generate data for building and testing AIML models. This will ensure that the data is realistic, has correct statistical characteristics (e.g., a rare event in the real data will also be a rare event in the synthetic data), and that key properties are maintained if large synthetic datasets are generated.

The next industry that we consider is transportation. Under that heading, we will consider data synthesis for planning purposes through microsimulation models and data synthesis for training models in autonomous vehicles.

Transportation

The use of synthetic data in the transportation industry goes back a few decades. The main driver is the need to make very specific planning and policy decisions about infrastructure in a data-limited environment. Hence the use of microsimulation models became important to inform decision-making. This is the first example we consider. The second example is the use of gaming engines to synthesize virtual environments that are used to train AIML models, which are then embedded in autonomous vehicles.

Microsimulation Models

Microsimulation environments allow users to do "what-if" analyses and run novel scenarios. These simulation environments become attractive when no real data is available at all and therefore synthetic data needs to be created.

In the area of transportation planning, it is necessary to evaluate the impact of planned new infrastructure, such as a new bridge or a new mall. Activity-based travel demand models can use synthetic data to allow planners to do that.

A commonly used approach to creating synthetic data for these models combines aggregate summaries (for example, from the census) with sample individual-level data collected from surveys. Census data normally provides information like household composition, income, and number of children. The aggregate data normally covers the whole population of interest but may not have all the needed variables and not to the level of granularity desired. The survey data will cover a sample of the population but have very detailed and extensive variables.

Synthetic reconstruction then uses an iterative process such as iterative proportional fitting (IPF) to create synthetic individual-level data that plausibly generates the aggregate summaries and uses the sample data as the seed. The IPF procedure was developed in the 1940s,[11] and has more recently been applied to the data synthesis

[11] W. Edwards Deming and Frederick F. Stephan, "On a Least Squares Adjustment of a Sampled Frequency Table When the Expected Marginal Totals Are Known," *Annals of Mathematical Statistics* (1940): 427–444. *https://oreil.ly/Ldktz*.

problem.[12] IPF has some known disadvantages in the context of synthesis; for example, when the survey data does not cover rare situations. More robust techniques, such as combinatorial optimization, have been developed to address them.[13]

The next step is to use other data, collected also through surveys or directly from individuals' cell phones, characterizing their behaviors and movements. This data, such as the factors that influence the choice of mode of transportation taken by an individual, is used to build models.

By combining the synthetic data with the models, one can run microsimulations of what would happen under different scenarios. Note that the models can be cascaded in the simulation, describing a series of complex behaviors and outcomes. For example, the models can inform decisions like the impact on traffic, public transportation usage, bicycle trips, and car usage when having a new bridge or a new mall constructed in a particular location. These microsimulators can be validated to some extent by ensuring that they give outputs that are consistent with reality under known historical scenarios. But they can also be used to simulate novel scenarios to inform planning and policy making.

Let's now consider a very different use case for synthetic data in the context of developing AIML models for autonomous vehicles. Some of these models need to make decisions in real time and can have significant safety impacts. Therefore, the robustness of their training is quite critical.

Data Synthesis for Autonomous Vehicles

One of the critical functions of an autonomous vehicle (AV) is perception. An AV must be able to recognize stationary and dynamic objects in the vehicle's path and surrounding it. Camera, lidar, and radar systems provide the data feeds to an onboard AI

[12] Richard J. Beckman, et al., "Creating Synthetic Baseline Populations," *Transportation Research Part A* 30, no. 6 (1996): 415–429. *https://oreil.ly/Qm6Jc*.

[13] Zengyi Huang and Paul Williamson, "A Comparison of Synthetic Reconstruction and Combinatorial Optimization Approaches to the Creation of Small-Area Microdata," University of Liverpool (2002). *https://oreil.ly/ZXq0I*; Justin Ryan, et al., "Population Synthesis: Comparing the Major Techniques Using a Small, Complete Population of Firms," *Geographical Analysis* 41 (2009): 181–203. *https://oreil.ly/gUsDs*.

supercomputer to enable object identification, as well as speed and distance determination of these objects.

While real-world data is used to train the numerous deep neural networks that run inside an AV, synthetic data is essential to test and validate the AI models that process these signals. Real-world data cannot capture every edge case, or rare or dangerous scenarios—such as an animal darting into the vehicle's path or direct sunlight shining into a camera sensor—that an autonomous vehicle could encounter. Additionally, the recorded environment is fixed and cannot respond to changes in the system's behavior when it is run through the scenario multiple times.

The only way to address these gaps is to leverage synthetic data. By generating an extensive list of customizable scenarios, engineers can model real-world environments—as well as create entirely new ones—that can change and respond to different behaviors (see Figure 3-3). While driving in the real world provides a valuable tool for validation, it is not nearly exhaustive enough to prove a vehicle is capable of driving without a human at the wheel.

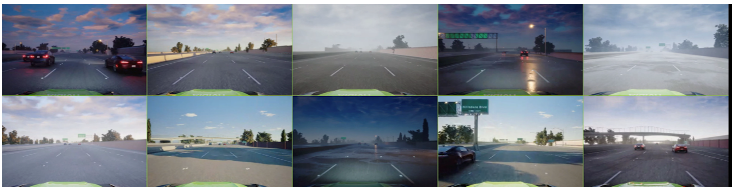


Figure 3-3. Simulation enables large-scale scenario modeling in different environments, weather conditions, and times of day

Generally, in autonomous vehicles, simulation can be broken into two main categories: postperception and end-to-end. *Postperception simulation* is used for development of planning and control algorithms. It simulates the world and provides a list of objects in the world with information about each to the planner. From the list of objects, the planning and control algorithms must decide what to do and when to execute.

The other category is *end-to-end simulation*, in which the simulator simulates raw sensor data from a 3D world. The sensor data is streamed to the perception networks in the AV stack, which must comprehend the world state. The perception networks then provide their output to the planning and control stacks. End-to-end simulation has the advantage of introducing realism into the simulation pipeline in the form of noise and errors in the perception stack as it would in a vehicle before it is passed to the planning and control algorithms.

Generating synthetic data detailed enough for autonomous vehicle testing requires a rigorous process. First, the environment must be created. It can either replicate a location in the real world, like New York City, using actual data, or be an entirely synthetic place. For end-to-end simulation, everything in the environment must accurately simulate the same material properties as the real world; for example, the reflection of light off metal or the surface of asphalt.

A wide range of weather conditions can be modeled, as well as lighting conditions based on the time of day. Even the sensor models themselves must replicate the output of the sensors being tested, requiring massive amounts of computing capacity.

Among the wide variety of simulation methods available for testing and validation (such as model-in-the-loop, software-in-the-loop, and object based), hardware-in-the-loop simulation provides one solution to achieving high-fidelity bit-accurate results. An example is the two-server NVIDIA DRIVE Constellation platform shown in Figure 3-4.

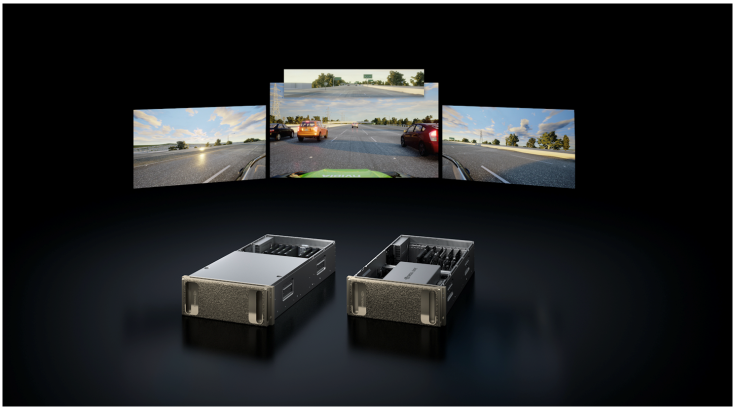


Figure 3-4. The NVIDIA DRIVE Constellation simulation platform uses one server to generate output from sensors, while the other contains the in-vehicle AV computer making driving decisions in real time

One server simulates the output from camera, radar, and lidar sensors. The in-vehicle AV computer inside the other server then receives the data as if it is coming from a real-world driving environment, runs the full vehicle software stack, makes decisions, and sends vehicle control commands back to the simulator. This closed-loop process enables bit-accurate, timing-accurate hardware-in-the-loop testing.

The work needed to perform hardware-in-the-loop testing is significant, both in terms of infrastructure as well as in the vehicle. Achieving the fidelity necessary for autonomous vehicle validation is incredibly compute-intensive. First, a detailed world has to be generated. Next, the sensor output must be simulated in a physically accurate way—which takes time and massive amounts of computing horsepower. Then the in-vehicle hardware and software can be fully assessed through an extensive suite of simulated scenarios.

Conclusions

This chapter provided examples of the applications of synthetic data in various industries. We have seen the adoption of synthetic data grow in these industries, as well as others, over the last couple of years. Because data access challenges are not likely to get any easier or go away anytime soon, applicability of data synthesis to more use cases is expected to grow.

Having said all this, more work still needs to be done to improve data synthesis technology and to make its adoption easier. The next chapter is more forward-looking in that it lays out the big things that would be very beneficial to develop to support the growth of synthesis capabilities in practice.

CHAPTER 4
The Future of Data Synthesis

While significant progress has been made over the last few years in making synthetic data generation practical and scalable, we need some additional requirements for future work and improvements to the current state of practice. This chapter is a summary of the key issues that need to be worked on. It does not present a research and development agenda, but rather a set of items to consider when developing such an agenda.

We cover four main issues. First, we need to develop a data utility framework. Such a framework would make it easier to benchmark various data synthesis techniques. The second issue, which is coming up more frequently, is the need to remove certain relationships from synthetic data for commercial or security reasons. Third, data watermarking will become increasingly important as more synthetic data is generated and shared. Finally, simulators that can generate different types of synthetic data would provide powerful capabilities.

Creating a Data Utility Framework

As discussed in Chapter 1, data utility is important for the adoption of synthetic data. The higher the data utility of synthetic data, the greater the number of use cases where it would be a good tool to accelerate AIML efforts, and the more likely that analysts will be comfortable using it.

In practice, we are seeing that a significant dataset, the 2020 decennial US census, is being shared as synthetic data and derivatives from synthetic data. The question of whether the utility of synthetic data is good enough or not may no longer be the right one to ask. We have entered the era of large-scale synthetic data, and the utility levels that are available today may be sufficient for many practical problems.

The question now is how do we demonstrate this data utility to analysts and data users so that they are confident and comfortable using synthetic data? The answer has two parts (at least):

1. Data utility is defined as the ability to get substantively similar results on synthetic data as on real data.
2. Data utility is defined relative to an alternative method of getting access to data, such as de-identified data.

It is good practice to perform a utility assessment for every synthesized dataset, and this is where a data utility framework would be of value. The availability of a validation server would be a plus. Over time, using synthetic data as a proxy for real data will become more accepted, especially as synthesis methods continue to improve.

De-identified data is generally considered a good proxy for real data. How does the utility of synthetic data compare to the utility of de-identified data? This remains an empirical question and, over time, evidence will be accumulated to inform this issue. However, we have argued earlier in this report that, practically, the economics of de-identification are potentially unfavorable compared to those of data synthesis.

The use cases that we discussed in this report can be expanded upon if it is possible to manipulate the synthetic data. This means that instead of generating data that has high fidelity to real data, we want to represent something different. In the next section, we consider the need to remove relationships or information from generated synthetic data.

Removing Information from Synthetic Data

Interesting applications emerge when we start looking at *hybrid synthetic data*. This data is generated from real data, but then is also manipulated to exhibit characteristics that were not in the original data. This section examines the removal of information from synthetic data to hide sensitive information.

In domains such as law enforcement and intelligence, there is a need to build AIML models, which means that there is a need to get access to data. These models can, for example, characterize determinants of crime and predict adversary activities. But the data owners may want to hide certain attributes or relationships to ensure that they are not exhibited in the generated data. These hidden attributes or relationships pertain to highly sensitive or classified information that should not be known more broadly; for example, those that reveal data surveillance capabilities or sources.

Another scenario requiring specific attributes or relationships to be hidden comes up in commercial settings. For example, a financial services company may want to create a synthetic version of a dataset but not reveal specific commercially sensitive information in that data. Therefore, there is a need to partially synthesize the data or mask parts of it after synthesis.

In the next section, we discuss how data watermarking can be a useful capability as the adoption of synthetic data grows. Watermarking of data has been used historically to establish data provenance; for example, in the case of a data breach. Establishing a synthetic data signature would be a new application of these capabilities.

Using Data Watermarking

Imagine a future whereby synthetic data is around every corner and is commonly used as a key component of the data analytics and secondary processing ecosystem. Some concerns that have been expressed include the ability to tell the difference between real data and synthetic data.

Data watermarking methods can address this concern. One type of watermark would be a unique data pattern that is deliberately embedded within the synthetic data and that is recoverable. Alternatively, a watermark can be computed algorithmically from the existing patterns in the data, effectively being a signature characterizing the data.

Whenever there is a question about the status of a dataset, it would be compared to known watermarks to determine whether it is real or synthetic. Given that synthetic data is generated through a stochastic process, every instance of a dataset will have a unique pattern to it.

The difficulty with practical data watermarks is that they need to be invariant to data subsets. For example, would the watermark still be detectable for a subset of the variables or for a subset of the rows in the dataset?

As our understanding of specific processes improves over time, it becomes easier to build plausible models and simulators of these processes. The simulators can act as data synthesizers as well. We discuss this topic in the next section.

Generating Synthesis from Simulators

Within the context of data synthesis, a *simulator* is a statistical or a machine learning model, or a set of rules that characterize a particular process embedded in a software application. When the application is executed, it generates data from these models or rules. We saw some of that in the context of gaming engines in Chapter 3, which are used to generate data for training robots and training and testing autonomous vehicle systems. In the same chapter, we looked at microsimulation as another example of a simulation capability. However, the concept can be implemented more broadly and in other domains.

Generating data from simulators raises the possibility of setting the desired heterogeneity of the synthetic data. For example, a simulator can effectively oversample rare events or catastrophic events to ensure that the trained models are robust against a larger domain of inputs. However, these events need to be somewhat plausible. For example, when generating images for training autonomous vehicles, we would not want to have scenes with cars on top of buildings or floating in air. Plus, how would one validate the trained models in practice since the real situations are unlikely to occur (or occur rarely in the real world)?

Some domains are more amenable to simulators than others. As our understanding of health systems and biological systems improve, they can plausibly be modeled more accurately, and these models can be used to generate data. This will start off being done at the macro level, but would increase in granularity over time.

In addition to being another source of synthetic data, simulators allow us to manipulate synthetic data. For example, if we want to test a new AIML technique to see if it can detect the genetic and other characteristics of patients who respond particularly well to a drug, we can use a simulator to create datasets with signals of different strengths.

The list of items for consideration as part of the future of data synthesis did not cover new techniques for data synthesis. However, that is also an area of active development, with innovations in genetic algorithms and deep learning models. A deep dive into specific algorithms for synthesis is a specialized topic for a different publication.

Conclusions

Synthetic data represents an exciting opportunity to solve some practical problems related to accessing realistic data for numerous significant use cases. The demand for data to drive AIML applications, the greater availability of large datasets, and the increasing difficulty in getting access to this data (because of data protection regulations and concerns about data sharing) have created a unique opening for data synthesis technologies to fill that gap.

As we discussed, data access problems span multiple industries, such as manufacturing and distribution, healthcare and health research, financial services, as well as transportation and urban planning (including autonomous vehicles). The techniques and methodologies that have been developed over the last few years have achieved substantial data utility milestones. The number of use cases for which data synthesis provides a good solution is increasing rapidly.

In this report, we have looked at industries in which synthetic data can be applied in practice to solve data access problems. Again, a characteristic of these use cases is their heterogeneity and the plethora of problems that synthesis can solve. They are not a comprehensive list of industries and applications, but do highlight what early users are doing and illustrate the potential.

While we did not discuss the privacy benefits of synthetic data much in this report, it is important to highlight that in our closing. The current evidence suggests that the risk of matching synthetic data to real people and learning something new from that matching is very small. This is an important factor when considering the adoption of data synthesis.

Once a decision has been made to adopt data synthesis, the implementation process must be considered. As data synthesis becomes more programmatic across the enterprise, a center of excellence becomes an appropriate organizational structure as opposed to running individual projects. Depending on whether the demand for data synthesis is discrete for specific datasets or a continuous dataset, an architectural decision needs to be made on the implementation of a pipeline and its integration within a data flow. A data pipeline architecture would help with synthesis technology implementation.

Exciting advances in synthetic data generation are in development today that will help with broader adoption of this approach and type of technology. It was already noted some time ago that the future of data sharing, data dissemination, and data access will utilize one of two methods: interactive analytics systems or synthetic data.[1]

[1] Jerome P. Reiter, "New Approaches to Data Dissemination: A Glimpse into the Future (?)," *CHANCE* 17, no. 3 (June 2004): 11–15. *https://oreil.ly/x89Vd*.

Acknowledgements

The preparation of this report benefited from a series of interviews with subject matter experts. I would like to thank the following individuals for making their time available to discuss and/or provide input on their experiences and thoughts on the synthetic data market and technology: Fernanda Foertter, Jim Karkanias, Alexei Pozdnoukhov, Rev Lebaredian, John Ashley, Rob Csongor, and Simson Garfinkel.

Furthermore, this work has benefited from discussions with and feedback from Lucy Mosquera and Richard Hoptroff.

Rob Csongor and his team provided the content for the section on autonomous vehicles.

About the Author

Dr. Khaled El Emam is a senior scientist at the Children's Hospital of Eastern Ontario (CHEO) Research Institute and leads the Electronic Health Information Laboratory, which conducts applied academic research on synthetic data generation methods and tools and re-identification risk measurement. He is also a professor in the Faculty of Medicine at the University of Ottawa, Canada.

Khaled currently invests in, advises, and sits on the boards of technology companies developing data protection technologies and building analytics tools to support improved health care delivery and drug discovery. He is also the cofounder of Replica Analytics, which develops data synthesis solutions for health data.

He has been performing data analysis since the early '90s, building statistical and machine learning models for prediction and evaluation. Since 2004, he has been developing technologies to facilitate the sharing of data for secondary analysis, from basic research on algorithms to applied solutions development that have been deployed globally. These technologies addressed problems in anonymization and pseudonymization, synthetic data, secure computation, and data watermarking.

He has written and edited multiple books on various privacy and software engineering topics. In 2003 and 2004, he was ranked as the top systems and software engineering scholar worldwide by the

Journal of Systems and Software, based on his research on measurement and quality evaluation and improvement.

Previously, Khaled was a senior research officer at the National Research Council of Canada. He also served as the head of the Quantitative Methods Group at the Fraunhofer Institute in Kaiserslautern, Germany. He held the Canada Research Chair in Electronic Health Information at the University of Ottawa from 2005 to 2015, and has a PhD from the Department of Electrical and Electronics Engineering, King's College, at the University of London, England.