DIFFERENTIALLY PRIVATE NOISE ADDITION ON SMART METER DATA FOR EFFECTIVE PRIVACY RESEARCH

by
MOHAMED ZEINA

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of Master of Science

Sabancı University
December 2023

ABSTRACT

DIFFERENTIALLY PRIVATE NOISE ADDITION ON SMART METER DATA FOR EFFECTIVE PRIVACY RESEARCH

MOHAMED ZEINA

COMPUTER SCIENCE AND ENGINEERING M.S. THESIS, DECEMBER 2023

Thesis Supervisor: Prof. Albert Levi

Keywords: Smart Meters, Differential Privacy, GAN, SMOTE

Smart meters measure utility consumption, like electricity, gas, or water. Utility providers publish smart meter data to contribute to research and innovation by performing analysis on the data. Data owners utilize limited privacy techniques when publishing smart meter data, such as anonymization, which is susceptible to linkage attacks that allow for the re-identification of individuals. As a result, making smart meter data publicly available raises privacy concerns. Smart meter data could be misused to reveal personal information about daily routines, activities, and private characteristics of households. Differential privacy is a framework that balances the conflicting goals of data utilization and individual privacy. In this thesis, we aim to show to what extent differential privacy can effectively balance household privacy while providing efficient data utilization and information extraction. For this purpose, we use household electricity consumption data. The data set was unbalanced, so Synthetic Minority Oversampling Technique (SMOTE) was used to balance it. Moreover, since the data set was small, Generative Adversarial Network (GAN) technique was used to generate synthetic data based on the real data. Using IBM's diffprivlib library, conducted various experiments for adding noise to the data and performed machine-learning-based classification over noisy data. We evaluated various noise levels to determine the optimal one that gives a similar classification performance as the original data. It has been determined that the Gaussian Naive Bayes model with differential privacy provides a better differential privacy level (smaller ε) than the Logistic Regression model with differential privacy. Furthermore, it has been shown that the Gaussian noise addition mechanism is the best among the other mechanisms for achieving differential privacy.

ÖZET

AKILLI SAYAÇ VERILERINDE ETKIN MAHREMIYET ARAŞTIRMALARI IÇIN DIFERANSIYEL GIZLI GÜRÜLTÜ EKLEME

MOHAMED ZEINA

BILGISAYAR BILIMI VE MÜHENDISLIĞI YÜKSEK LİSANS TEZİ, ARALIK 2023

Tez Danışmanı: : Prof. Dr. Albert Levi

Anahtar Kelimeler: Akıllı sayaçlar, Diferansiyel Mahremiyet, Üretken Rekabetçi Ağ (GAN), Sentetik Azınlık Örneklem Artırma Tekniği (SMOTE)

Akıllı sayaçlar, elektrik, gaz veya su gibi hizmetlerin tüketimini ölçer. Hizmet sağlayıcıları, veri üzerinde analiz yaparak araştırma ve inovasyona katkıda bulunmak amacıyla akıllı sayaç verilerini yayınlamaktadırlar. Veri sahipleri, akıllı sayaç verilerini yayınlarken, bireylerin yeniden tanımlanmasına olanak tanıyan bağlantı saldırılarına karşı hassas olan anonimleştirme gibi, sınırlı mahremiyet tekniklerinden yararlanmaktadır. Sonuç olarak akıllı sayaç verilerinin kamuya açık hale getirilmesi mahremiyet endişelerini artırmaktadır. Akıllı sayaç verileri, hane halkının günlük rutinleri, faaliyetleri ve mahrem özellikleri hakkındaki kişisel bilgileri ortaya çıkarmak için kötüye kullanılabilir. Diferansiyel mahremiyet, verinin kullanışlılığı ile bireysel mahremiyetin çatışan hedeflerini dengeleyen bir çerçevedir. Bu tezde, diferansiyel mahremiyetin, etkin veri kullanımı ve bilgi çıkarımı sağlanırken, ev halkının mahremiyetini ne derece etkili bir şekilde dengeleyebileceğinin gösterimini amaçlanmaktadır. Bu amaçla ev elektriği tüketim verileri kullanılmıştır. Veri seti dengesiz olduğundan Sentetik Azınlık Örneklem Artırma Tekniği (SMOTE) kullanılarak dengeleme yapıldı. Öte yandan, veri seti küçük olduğundan, gerçek verilere dayalı sentetik veriler üretmek için Üretken Rekabetçi Ağ (GAN) tekniği kullanıldı. IBM'nin diffprivlib kütüphanesi kullanılarak veriye gürültü eklemek için çeşitli deneyler yapıldı ve gürültülü veri üzerinde makine öğrenimine dayalı sınıflandırma gerçekleştirildi. Orijinal verilere benzer sınıflandırma performansı sağlayan en uygun olanı belirlemek için çeşitli gürültü seviyelerinin değerlendirilmesi yapılmıştır. Diferansiyel

mahremiyete sahip Gaussian Naive Bayes modelinin, diferansiyel mahremiyete sahip Lojistik Regresyon modeline göre daha iyi bir diferansiyel mahremiyet düzeyi (daha küçük ε) sağladığı belirlendi. Ayrıca, Gaussian gürültü ekleme mekanizmasının, diferansiyel mahremiyet elde etmek için diğer mekanizmalar arasında en iyisi olduğu gösterilmiştir.

ACKNOWLEDGEMENTS

I want to express my heartfelt gratitude to all those who supported me in achieving this goal. A special thanks to Prof. Albert Levi for his unwavering support, constant encouragement, and endless patience throughout this project. I want to thank Prof. Yücel Saygın and Assoc. Prof. Ali İnan for honoring me by judging my work. I am also grateful to my family members for their constant support. Their constant presence has been my source of strength. I would also like to express my gratitude to all the mentors I have had during my academic years. Each one of them has taught me something invaluable, and I am thankful for their guidance.

This is dedicated to all my family members.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

There has been a rapid increase in data collection across various domains in this digital age. It is expected that the volume of data collected will reach more than 180 zettabytes by the year 2025 Taylor (2022). The rise in the number of digital devices and the widespread use of the internet in numerous aspects of our daily lives are among the factors behind this growth in data collection.

E-commerce is an online activity considered a crucial aspect of business operations Hua (2016). The exponential increase in online activity, such as mobile app usage and e-commerce transactions, is a primary factor contributing to the growth of data collection. Organizations collect customer data for analysis, including click, search, and purchase information.

Technological advancements in data processing and cloud computing have made it possible to store massive amounts of data. This provides organizations with the ability to extract valuable insights from the data they collect, including identifying patterns about their customers. The Internet of Things (IoT) has made it even easier to collect data, as there is an interconnected network of devices with sensors that are all connected to the Internet. The devices generate real-time data that can be used for various applications.

While there are many advantages to the increase in data collection, significant privacy concerns also need to be acknowledged. Data collection includes vast amounts of personal and sensitive information, such as individual preferences, locations, and interactions. The utilization of personal information has practical applications, such as recommendation systems in e-commerce, entertainment, and search queries Kumar & Reddy (2014). However, handling personal and sensitive information raises serious privacy concerns, which can hinder data disclosure.

Technological advancement in the energy sector is being made with the deployment of smart meters. Smart meters are devices that measure and monitor utility consumption, such as electricity, gas, or water. The number of residential smart meters deployed has increased significantly in recent years, which has resulted in a

notable increase in the amount of data collected by smart meters. In contrast to conventional meters, smart meters have data transmission capability that allows for two-way communication between the utility company and the meter. This feature makes it possible to gather data in almost real-time, giving utility providers access to up-to-date and accurate information. Utility providers can use smart meters' data to forecast future usage patterns and identify trends Jain, Babu, Nair & Sawle (2021).

Utility providers publish data from smart meters for several reasons. One of the main reasons is to contribute to research in the energy sector by providing access to smart meter data sets. Hofmann & Siebenbrunner (2023) conducted a study and published a data set that included hourly electricity consumption data of Norwegian households and answers to three surveys about household characteristics. Researchers and utility companies can use this data to analyze energy usage patterns, identify areas where improvements can be made to energy efficiency, and manage the energy grid more effectively. Cook, Schmitter-Edgecombe, Crandall, Sanders & Thomas (2009) discussed the importance of creating public data sets of smart home meter data to improve technology evaluation.

Smart meter data is often made public while using limited privacy techniques like anonymization. Anonymization involves removing or altering personally identifiable information, such as names and addresses, from the data set, as explained by Marques & Bernardino (2020). Hamza, Hefny & others (2013) have demonstrated how linkage attacks can still be carried out on data that has undergone anonymization to re-identify individuals. As a result, making smart meter data public raises some privacy concerns. Cook (2012) stated how individuals are hesitant to use sensing technologies in their homes due to privacy concerns

Smart meter data provides detailed information about a household's energy consumption, which could be used to reveal personal information about daily routines, activities, and private characteristics of a household. The problem is how to preserve the privacy of individuals while extracting useful information about the underlying population from the published data Dwork (2006). This problem is known as the *privacy-preserving analysis of data* problem.

Differential privacy emerged as a concept, offering a framework to balance the conflicting goals of data utility and individual privacy. It provides a mathematical guarantee of privacy and thus offers a more rigorous and formalized approach to privacy preservation. Unlike common privacy-preserving methods that rely on anonymization and data masking, differential privacy adds noise to the data analysis process, ensuring that the addition or removal of a single data point does not affect the

outcome of the analysis Desfontaines, Mohammadi, Krahmer & Basin (2019).

Smart meter data is often made public while using limited privacy techniques. We use differential privacy as a privacy mechanism by applying noise to the data. This thesis aims to experiment to see if we can publish smart meter data with some noise while still having good data utilization. Based on the extracted electricity consumption data, various supervised models were used to classify whether a household was occupied by a single individual or not, with each feature representing a distinct query. In other tests, different supervised models were used to attempt to classify the number of people inside the household. Differential privacy was achieved by applying noise to the data to analyze its effect on performance metrics, such as the accuracy and F1 score of the models. The data set used only had 4,232 households' consumption data. The Generative Adversarial Network (GAN) technique was used to expand the data set to generate synthetic data. In addition, the data sets used to train the models were unbalanced, so the Synthetic Minority Oversampling Technique (SMOTE) was used to balance the data sets.

Performance evaluation was carried out using values of optimal epsilon per feature. Optimal epsilon per feature values are the smallest possible values of epsilon per feature at which the performance metrics, such as the F1 score or the accuracy of the classifier, reach their default values (i.e., values without any noise addition). It has been determined that the Gaussian Naive Bayes model with differential privacy provides a better level of differential privacy than the Logistic Regression model with differential privacy. This has been concluded since, for both accuracy and F1 score, the Gaussian Naive Bayes model had a smaller optimal epsilon per feature (ε). Additionally, it has been shown that the Gaussian noise-addition mechanism is the best among the other mechanisms for achieving differential privacy. This has been concluded since, for both performance metrics, the Gaussian NB model being used for classification was able to reach its default metric values with smaller optimal values of ε per feature and with $\delta = 1$ than when using other noise addition mechanisms. It was also discovered that the smart meter data can be utilized for information extraction while using differential privacy with small values of ε to achieve better privacy.

The remainder of the thesis is organized as follows: Related work is reviewed in Chapter 2. Next, in Chapter 3, we present the data set and methodology. The results are presented in Chapter 4, and Chapter 5 concludes the results.

2. RELATED WORK

Due to the increasing availability of electricity consumption data, researchers have extensively used machine-learning models and data mining techniques to analyze electricity consumption data in recent years. There has been a significant amount of literature dedicated to non-intrusive load monitoring (NILM). NILM is a method that identifies the appliances being used in a household and their corresponding energy consumption levels. It works by analyzing fluctuations in voltage and current in consumption data Revuelta Herrero, Lozano Murciego, López Barriuso, Hernández de la Iglesia, Villarrubia González, Corchado Rodríguez & Carreira (2018). The process can be challenging because each appliance has a unique energy signature. Firth, Lomas, Wright & Wall (2008), Chang (2012), and Tina & Amenta (2014) all used NILM for appliance recognition. Armel, Gupta, Shrimali & Albert (2013) explained the various benefits of obtaining appliance-level data, including consumer benefits, research and development, and utilities and policies. Zeifman & Roth (2011) and Zoha, Gluhak, Imran & Rajasegarar (2012) both conducted studies and concluded that there is no definitive set of features that can be used to detect and classify appliances with complete accuracy. However, Sadeghianpourhamami, Ruyssinck, Deschrijver, Dhaene & Develder (2017) later proposed a feature elimination process for NILM (Non-intrusive Load Monitoring) which can identify the best subset of features to be used by a model to classify appliances. This thesis differs from NILM in that it analyzes private household characteristics and electricity consumption data. This thesis differs from NILM in that it analyzes private household characteristics along with electricity consumption data.

Molina-Markham, Shenoy, Fu, Cecchet & Irwin (2010) demonstrated that usage patterns could be identified from smart meter data using statistical techniques, even in the absence of prior training or knowledge of household activities. They were able to demonstrate the potential for power consumption patterns to reveal a range of private information, such as how many people are in the home, sleeping routines, and eating routines. Their analysis used two months of data from three homes. McLoughlin, Duffy & Conlon (2012) conducted a study to investigate the

relationship between household characteristics and electricity consumption, using multiple linear regression analysis to model electricity consumption. Their findings concluded that there was a strong correlation between the characteristics of electricity consumption (such as time of use, maximum demand, load factor, and total consumption) and various household attributes (such as number of bedrooms, water heating, and cooking type). Their study further highlighted that electricity consumption data can potentially reveal private information about households.

Beckel, Sadamori, Staake & Santini (2014) also looked into the possibility of inferring household characteristics from electricity consumption data. They created a system that estimates some private characteristics of a household based on its electricity consumption by using supervised machine-learning techniques. They were able to demonstrate that eight private characteristics could be inferred with an accuracy ranging from 72% to 82%. Pekey, Çelebi, Anıl & Levi (2021) extracted features from 30-minute electricity consumption data and classified specific household private information using those features. They also concluded that private information about a household can be obtained from electricity consumption data. Beckel et al. (2014), McLoughlin et al. (2012), and Pekey et al. (2021) used the same data set as this thesis to analyze personal privacy. However, this thesis is different as we focus on applying differential privacy to examine whether it can effectively balance household privacy with efficient data utilization, rather than analyzing personal privacy like these prior studies.

A self-organizing map was employed in a different study by Beckel, Sadamori & Santini (2012) to examine electricity consumption traces. Using standard classification methods, they identified household properties that are likely to be inferred. They demonstrated that properties that are likely to be detectable using an automatic classification system are the size of a household and the income of its members.

3. OUR METHODOLOGY

In this chapter, we provide a detailed description of our contributions and the methods used. In Section 3.1, we discuss the data set used and how we processed it to extract meaningful data. In Section 3.2, we explain the features we extracted from the data and the methods used to extract them. In Section 3.3, we discuss how we used multilayer perceptron neural networks to create a General Adversarial Network (GAN) for synthetic data generation. In section 3.4, we explain the method we used to remove data imbalance in the data set, which was Synthetic Minority Oversampling Technique (SMOTE). We utilized a data set that had previously been used for analyzing personal privacy in other studies (discussed in Chapter 2). In Section 3.5, we showcase how the IBM Differential Privacy Library (diffprivlib) was used to evaluate model performance metrics under differential privacy. This was done to determine if differential privacy can effectively balance household privacy with efficient data utilization and information extraction.

3.1 Data Set & Data Processing

In a study conducted in Ireland by the Commission for Energy Regulation, Irish Social Science Data Archive (Irish Social Science Data Archive), smart meters were used to measure household electricity consumption. At 30-minute intervals, they gathered electricity consumption data from 4,232 households in Ireland. Data spanning 75 weeks (July 2009 – December 2010) was gathered, and kilowatt-hour (kWh) was used to calculate electricity consumption. Private information regarding the households, including the number of occupants, their income, and other private characteristics, was gathered through surveys.

The electricity consumption data from 4,232 households was provided in six CSV files. The CSV files have three columns. The first column shows the meter ID

of a household. The second column contains five-digit codes that represent two different things. The first three digits represent the day code, where day 1 corresponds to January 1st, 2009. The last two digits represent the time code and can have values between 1 and 48, each representing a 30-minute interval starting from 00:00:00 to 00:29:59. The last column contains the electricity consumed during the 30-minute interval (in kWh). In summary, each row shows the amount of electricity used by a particular household meter throughout a 30-minute interval on a given day. A reading of 0.140 kWh was recorded for meter ID 1392 on July 14, 2009, at 01:00:00–01:29:59, 195 days after January 1, 2009. This data is displayed in the first row of Table 3.1.

Table 3.1 Sample Rows from Electricity Consumption Data

Meter ID	**Date Time Code**	**Consumption (kWh)**
1392	19503	0.140
1392	19504	0.138
1392	19505	0.140
1392	19506	0.145

The data from the six CSV files was combined into a single data frame. After combining the CSV files, 157,992,996 rows were obtained in total. Duplicate rows were removed, along with rows that contained null readings. To extract useful dates and times from the five-digit codes, the column containing the codes was divided into two columns: one for the day code and another for the time code. Table 3.2 shows the same sample rows from Table 3.1 after splitting the five-digit codes.

Table 3.2 Electricity Consumption Data After Splitting the Five-Digit Codes

Meter ID	**Day Code**	**Time Code**	**Consumption (kWh)**
1392	195	3	0.140
1392	195	4	0.138
1392	195	5	0.140
1392	195	6	0.145

Time codes were mapped to extract times, while dates were obtained by adding the date code to December 31st, 2008. Table 3.3 shows the same sample rows from Table 3.2 after extracting dates and times.

Table 3.3 Electricity Consumption Data After After Extracting Dates and Times

Meter ID	**Date**	**Time**	**Consumption (kWh)**
1392	07/14/2009	1:29:59	0.140
1392	07/14/2009	1:59:59	0.138
1392	07/14/2009	2:29:59	0.140
1392	07/14/2009	2:59:59	0.145

3.2 Feature Extraction

Feature extraction was carried out using all 75 weeks of data. Beckel et al. (2014) and Pekey et al. (2021) used different categories of features, including consumption figures, ratios of consumption figures, temporal properties, statistical properties, and principal components. In this thesis, consumption figures were the only category of features extracted. The extracted features are as follows:

- Average Daily Electricity Consumption
- Average Electricity Consumption During Working Hours
- Maximum Electricity Consumption During Working Hours
- Average Electricity Consumption During Weekdays
- Average Electricity Consumption During Weekends
- Average Electricity Consumption During The Morning
- Average Electricity Consumption During The Afternoon
- Average Electricity Consumption During the Evening

To measure electricity consumption during working hours, we used data collected between 9 AM and 5:30 p.m. These hours correspond to the typical working hours in Ireland. For weekdays, we used data collected from Monday to Friday, while for weekends, we used data collected on Saturday and Sunday. Additionally, we measured electricity consumption during the morning hours between 5 AM and 12 PM, during the afternoon between 12 PM and 5 PM, and in the evening between 5 PM and 9 PM. All of the features are measured in kWh. For feature extraction., we utilized three Python libraries: NumPy, Pandas, and Sklearn Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg & others (2011). Jupyter Notebook was used as the coding environment.

The household characteristics obtained from the surveys did not require any data cleaning. Our focus was solely on the number of people living in each household, which was categorized into seven groups: 1, 2, 3, 4, 5, 6, or 7+. Category 7+ indicated that more than seven people were living in the household. Using the meter IDs, we combined the features extracted from the consumption data with the number of people living in each household to create a single data frame. We

derived the information on whether someone was living alone or not from the labels containing the number of people living in the household. A label equal to 1 indicates living alone, and 0 indicates not. Sample rows representing different households from the data frame that the models used are displayed in Tables 3.4 and 3.5.

Table 3.4 Electricity Consumption Data After Feature Extraction

Average Daily	**Average During Working Hours**	**Maximum During Working Hours**	**Average During Weekdays**	**Average During Weekends**
9.987	4.100	18.686	10.506	8.674
30.421	9.975	36.335	30.226	30.91
47.251	15.059	45.584	46.029	50.338
29.968	11.659	43.398	29.212	31.879
35.155	12.222	43.083	34.391	37.098

Table 3.5 Electricity Consumption Data After Feature Extraction

Average During Morning	**Average During Afternoon**	**Average During Evening**	**Number of People**	**Living Alone or Not**
3.071	1.872	2.125	1	1
7.179	5.993	8.181	3	0
7.541	9.580	14.655	4	0
8.271	6.822	7.761	2	0
7.861	7.537	8.445	4	0

3.3 Synthetic Data Generation

We used a data set of only 4,232 households' consumption data. To expand the data set, a Generative Adversarial Network (GAN) was used to generate synthetic data. Generative adversarial networks use deep learning to generate models, employing methods such as convolutional neural networks. GANs operate on the principle of a two-player zero-sum game, where the total gains of both players are zero. This means that any gain or loss of utility by one player is exactly balanced by the loss or gain of utility by the other player. In the case of GANs, the two players are two models: the generator and the discriminator. The generator model attempts to produce new samples by capturing the distribution of the real samples. On the other hand, the discriminator model is used to categorize samples as authentic (from the domain) or fake (generated) Wang, Gou, Duan, Lin, Zheng & Wang (2017).

A sequential multilayer perceptron neural network was used to create the discriminator model. The input to the neural network was nine-dimensional, consisting of eight features extracted from the consumption data and the number of people living in the household. The network had three hidden layers, all of which had ReLU activation functions. The first layer consisted of 256 neurons, while the second and third hidden layers consisted of 128 and 64 neurons, respectively. The output was a single neuron with a sigmoidal activation function, representing a probability.

The generator model was created using a sequential multilayer perceptron neural network. The network had two hidden layers, both with ReLU activation functions Radford, Metz & Chintala (2015). The first layer had 16 neurons, and the second layer had 32 neurons. The output consisted of 9 neurons with a linear activation function. The model was trained for 400 epochs, which means it was trained for 400 repetitions on the whole training set. The learning rate used was 0.001, which is the suggested learning rate. The binary cross-entropy function was used as the loss function to train the models. This function is suitable for training the discriminator because it considers a binary classification task. The Adam optimizer function was used to train both the discriminator and generator models Kingma & Ba (2014).

The GAN model was trained on 4096 data points, resulting in the generation of another 4096 data points. The synthetic data was then combined with the original data set, resulting in a combined data set containing information for 8,322 households after processing. The distribution of synthetic data was compared to the original data set for each feature.

The GAN model successfully replicated the general pattern of the original data for the average daily electricity consumption. However, it was unable to generate some of the unusual data points that are present in the original data set. The results are shown in Figure 3.1.

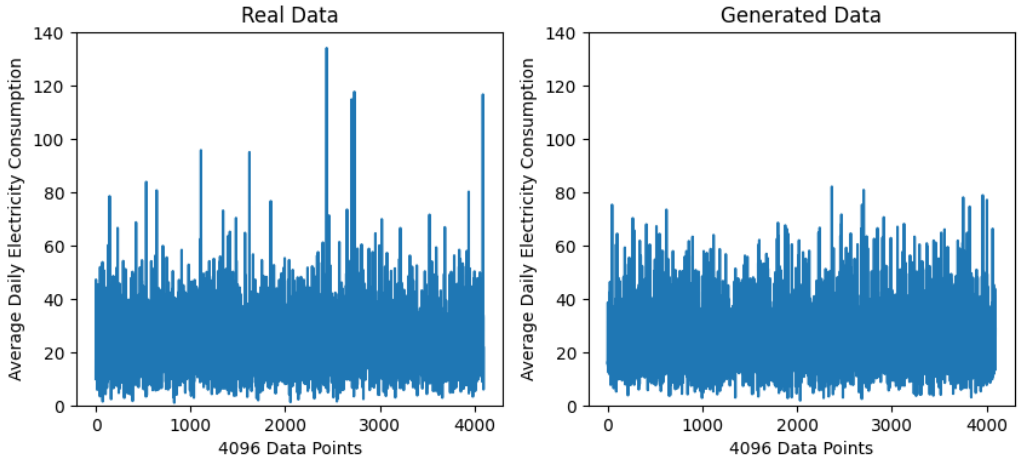

Figure 3.1 4096 Real Data Points vs 4096 Generated Data Points for Average Daily Electricity Consumption

Figure 3.2 compares the data distributions of real and generated data for the average daily electricity consumption, and it can be seen that both distributions are very similar.

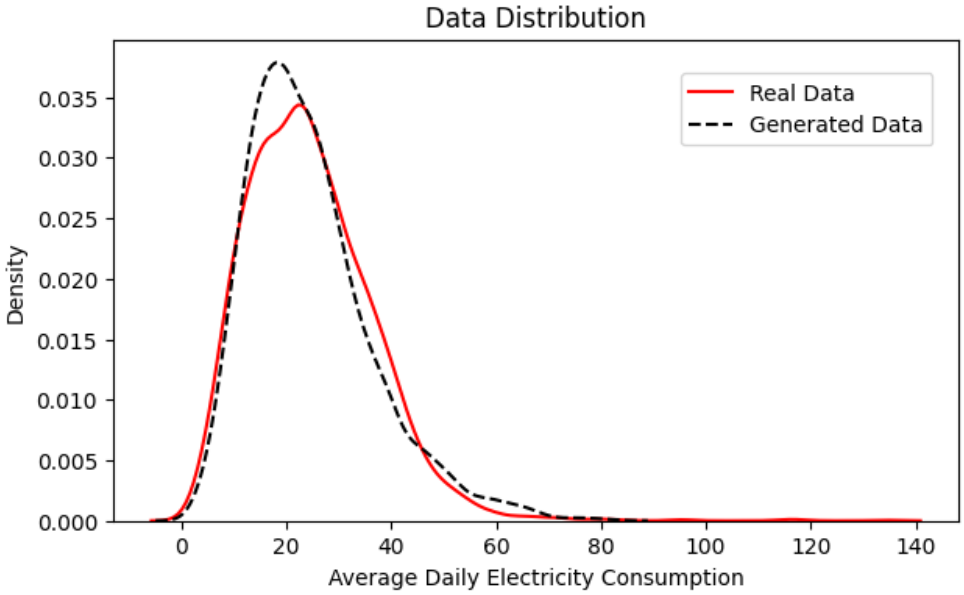

Figure 3.2 Data Density of 4096 Real Data Points vs 4096 Generated Data Points for Average Daily Electricity Consumption

The general trend of average electricity consumption during working hours in the original data was successfully reproduced by the GAN. However, the GAN was unable to generate some of the unusual data points that were present in the original data set, as demonstrated in Figure 3.3.

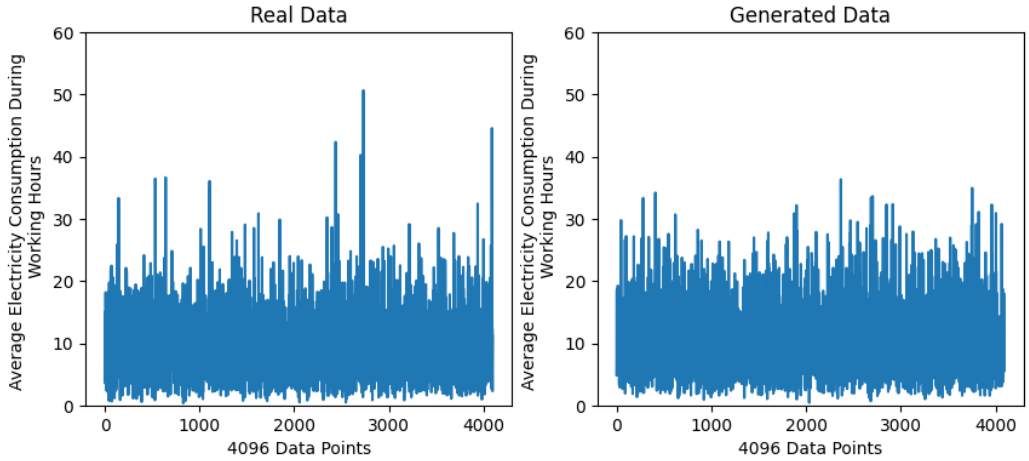

Figure 3.3 4096 Real Data Points vs 4096 Generated Data Points for Average Electricity Consumption During Working Hours

The data distributions of real and generated data for the average electricity consumption during working hours are compared, and Figure 3.4 shows that both distributions are very similar.

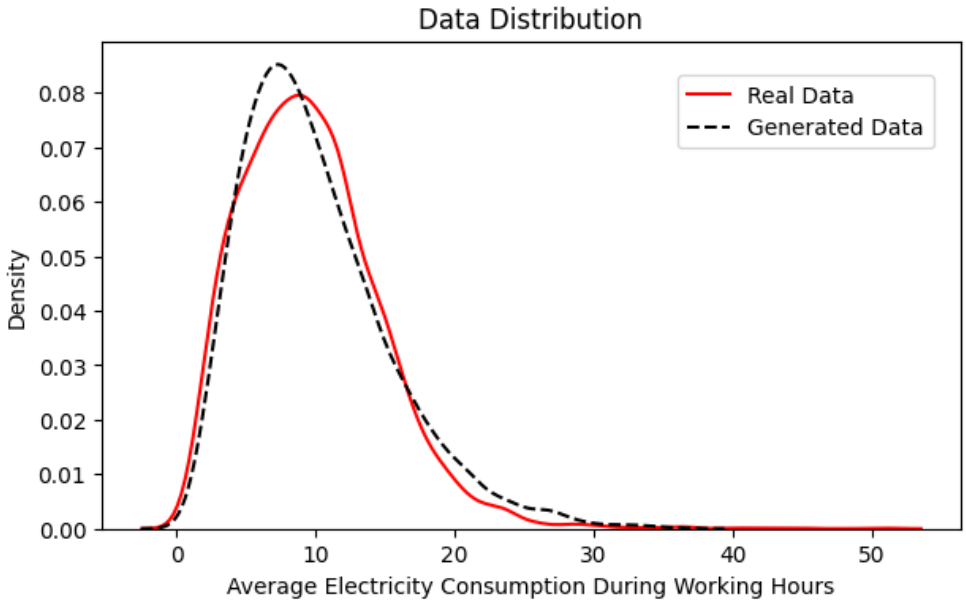

Figure 3.4 Data Density of 4096 Real Data Points vs 4096 Generated Data Points for Average Electricity Consumption During Working Hours

The GAN model was able to replicate the general pattern of maximum electricity consumption during working hours in the original data. It was also able to generate some of the unusual data points that are present in the original data set, and this can be seen in Figure 3.5.

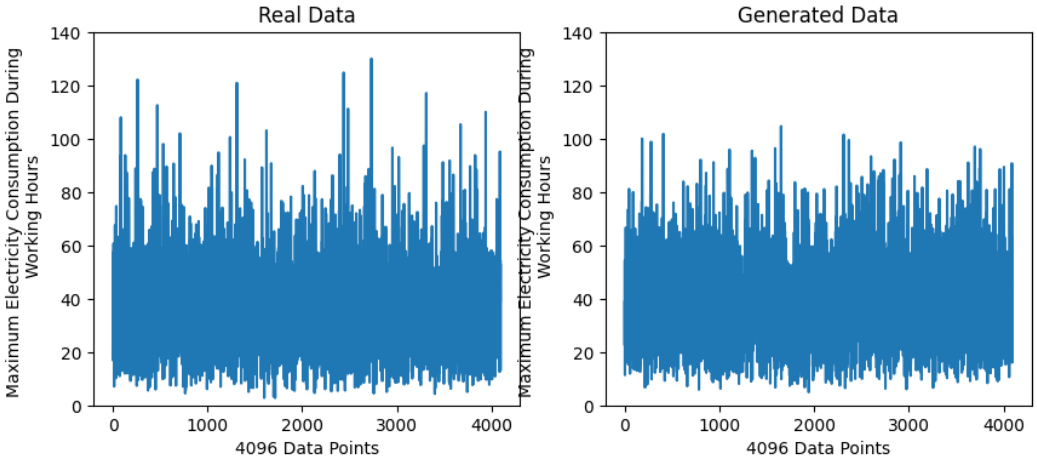

Figure 3.5 4096 Real Data Points vs 4096 Generated Data Points for Maximum Electricity Consumption During Working Hours

A comparison between the data distributions of real and generated data for the maximum electricity consumption during working hours was made, and it can be seen that both distributions are very similar, as shown in Figure 3.6.

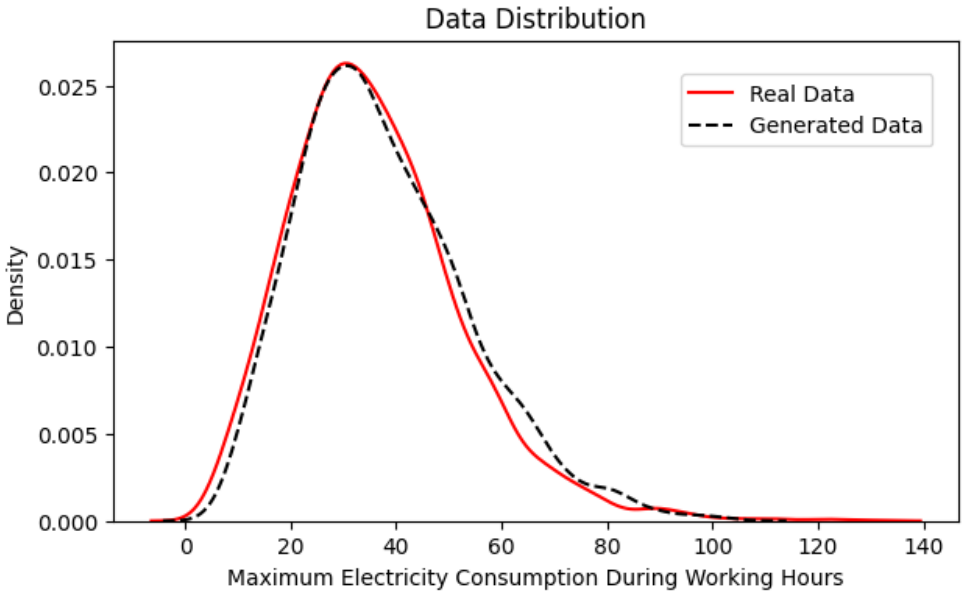

Figure 3.6 Data Density of 4096 Real Data Points vs 4096 Generated Data Points for Maximum Electricity Consumption During Working Hours

The GAN model was able to mimic the general trend of average electricity consumption during the weekdays in the original data. It followed the same behavior as for most features where it was unable to generate outliers, and this can be seen in Figure 3.7.

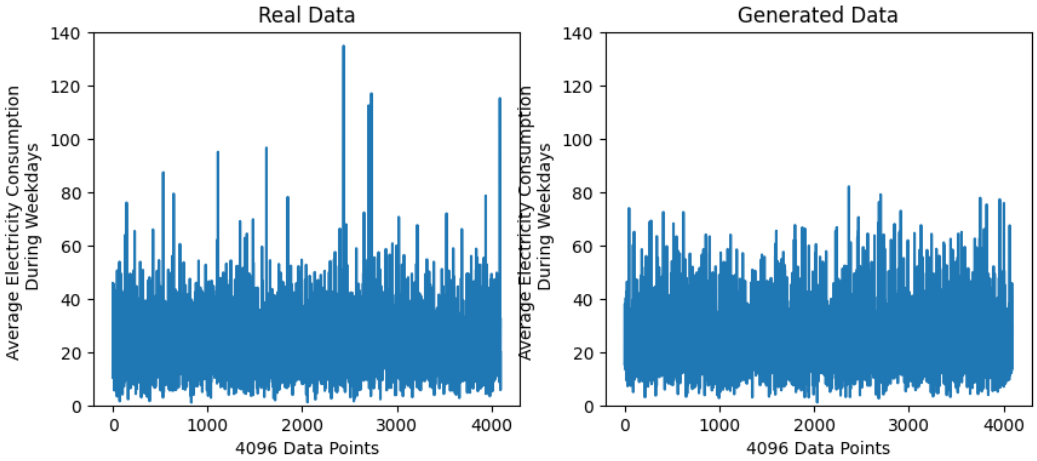

Figure 3.7 4096 Real Data Points vs 4096 Generated Data Points for Average Electricity Consumption During the Weekdays

For average consumption during the weekdays, the data distributions of real and generated data were compared. The results of the analysis indicated that the distributions are highly similar, as evidenced in Figure 3.6

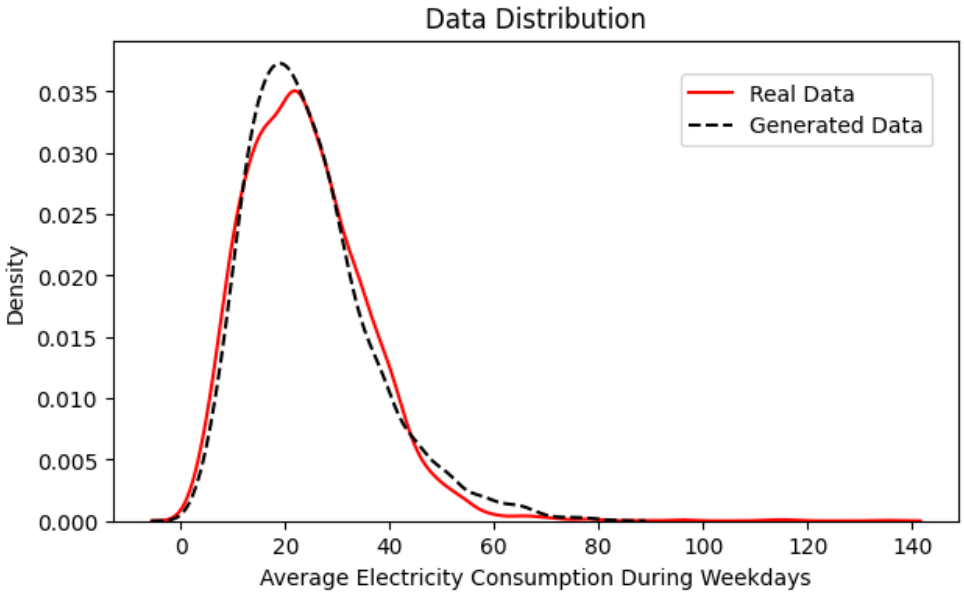

Figure 3.8 Data Density of 4096 Real Data Points vs 4096 Generated Data Points for Average Electricity Consumption During the Weekdays

Figure 3.9 shows that the GAN model was able to reproduce the general pattern of average electricity consumption during the weekends in the original data. Additionally, it was unable to generate outliers seen in the original data set.

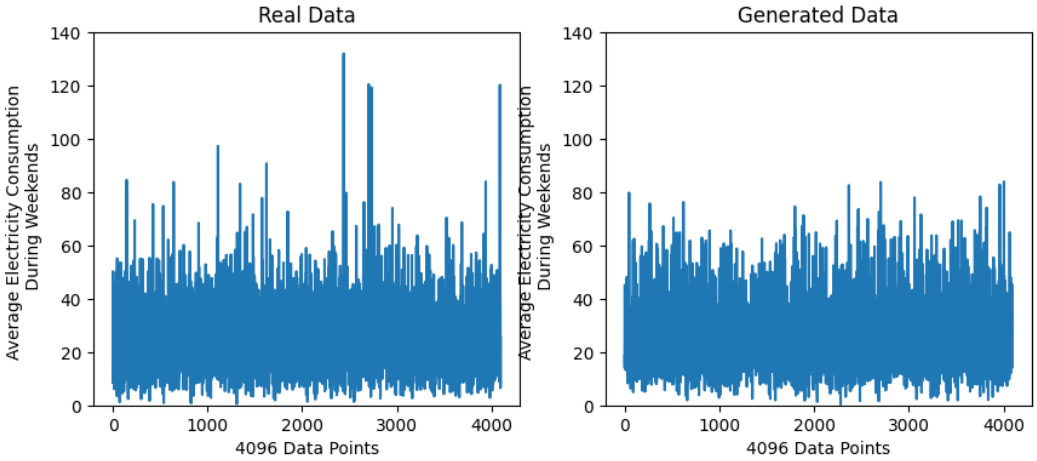

Figure 3.9 4096 Real Data Points vs 4096 Generated Data Points for Average Electricity Consumption During the Weekends

The data distributions of real and generated data were compared for average daily consumption during the weekends. The data distributions are very similar, as displayed in Figure 3.10

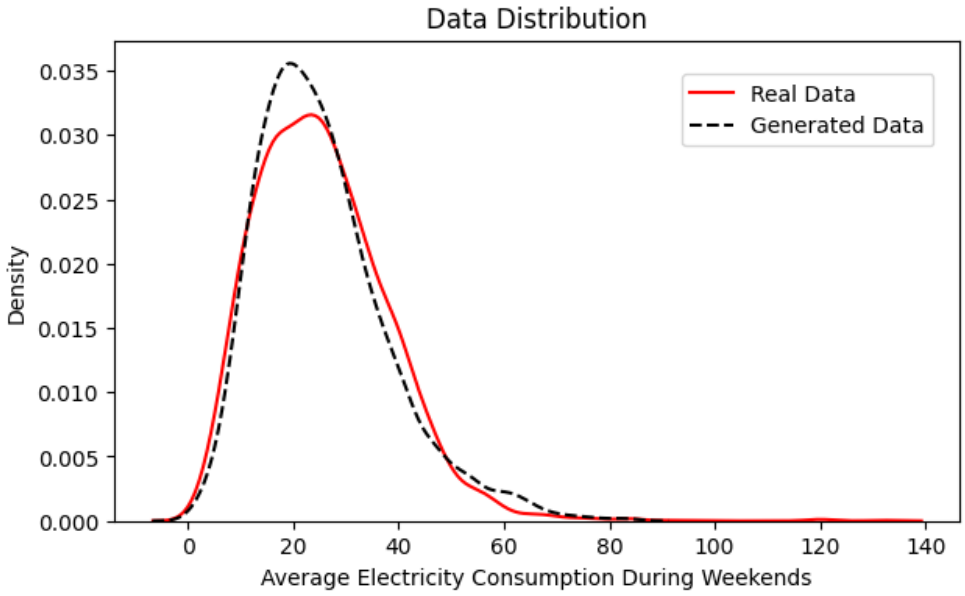

Figure 3.10 Data Density of 4096 Real Data Points vs 4096 Generated Data Points for Average Electricity Consumption During the Weekends

We can see from Figure 3.11 that the GAN model successfully replicated the overall pattern of average electricity consumption during the morning as observed in the original data set. Moreover, it was observed that the GAN model did not produce any outliers that are present in the original data set.

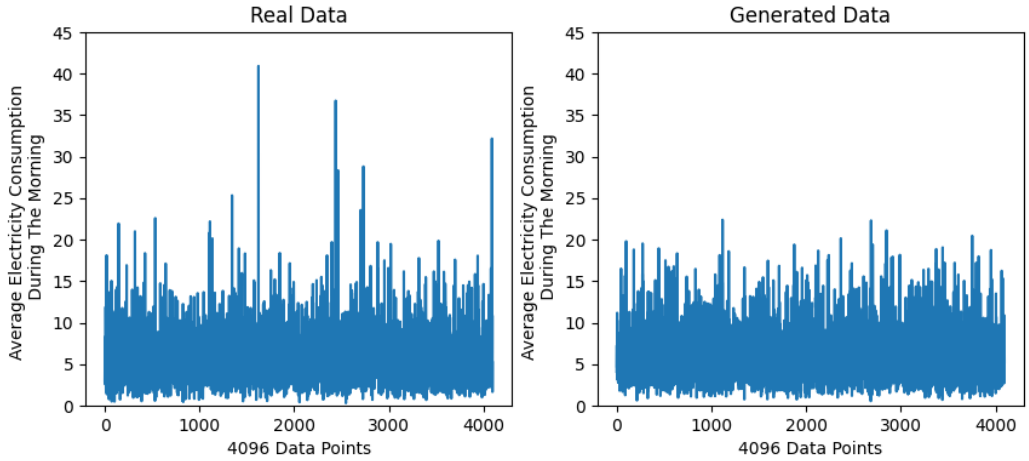

Figure 3.11 4096 Real Data Points vs 4096 Generated Data Points for Average Electricity Consumption During the Morning

Distributions of real and generated data for the average daily consumption of electricity during the morning. Figure 3.12 clearly shows that the distributions of both real and generated data are quite similar.

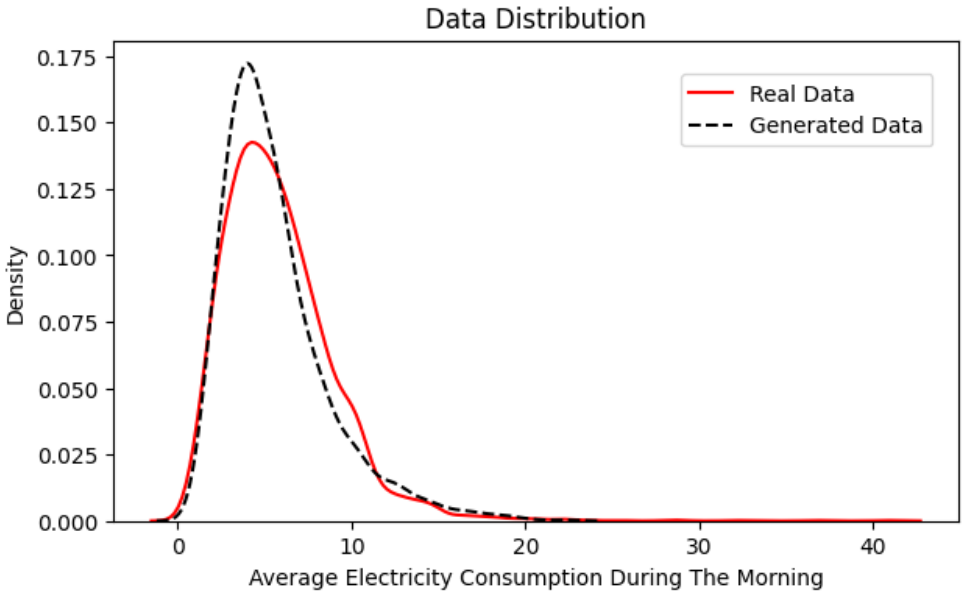

Figure 3.12 Data Density of 4096 Real Data Points vs 4096 Generated Data Points for Average Electricity Consumption During the Morning

In Figure 3.13, we can observe that the GAN model was able to successfully replicate the pattern of average electricity consumption during the afternoon, as seen in the original data set. Additionally, the GAN model did not produce any outliers that are present in the original data.

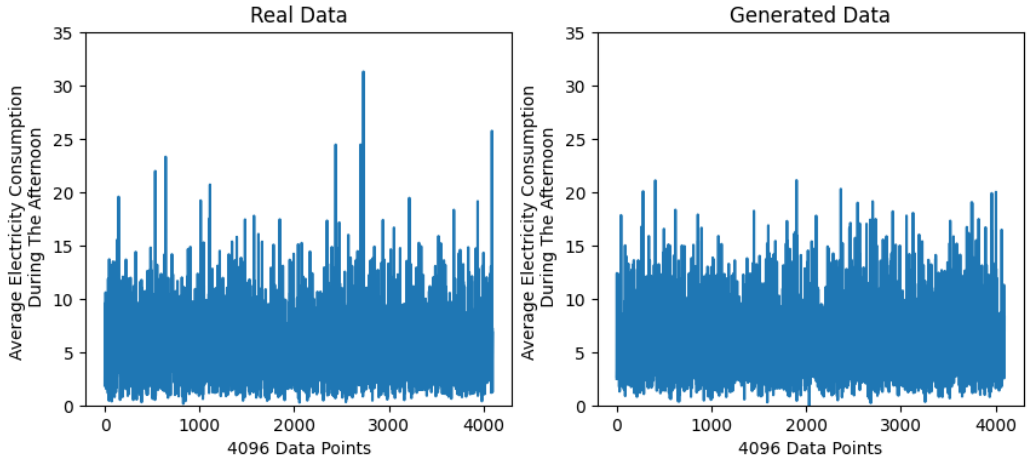

Figure 3.13 4096 Real Data Points vs 4096 Generated Data Points for Average Electricity Consumption During the Afternoon

Figure 3.14 compares the distributions of real and generated data for the average daily consumption of electricity during the afternoon. The distributions of both real and generated data in the figure are quite similar.

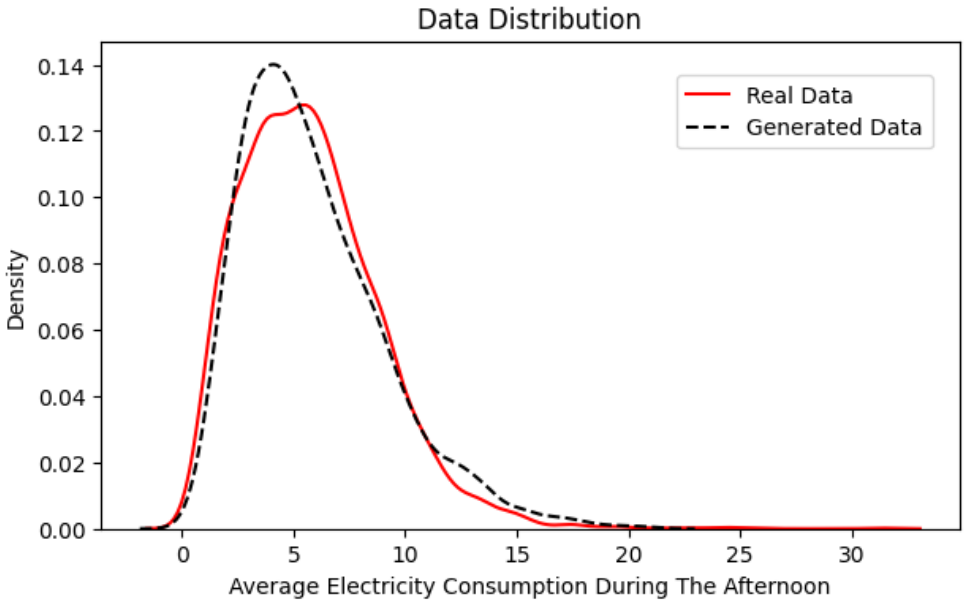

Figure 3.14 Data Density of 4096 Real Data Points vs 4096 Generated Data Points for Average Electricity Consumption During the Afternoon

The GAN model mimicked the general trend of average electricity consumption during the evening that was present in the original data but was unable to generate some of the outliers that are present in the original data set. The results are presented in Figure 3.15.

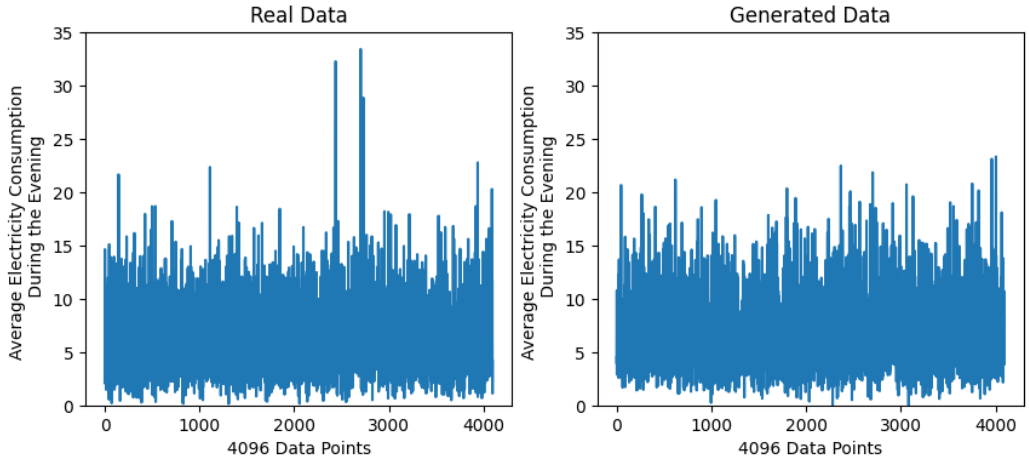

Figure 3.15 4096 Real Data Points vs 4096 Generated Data Points for Average Electricity Consumption During The Evening

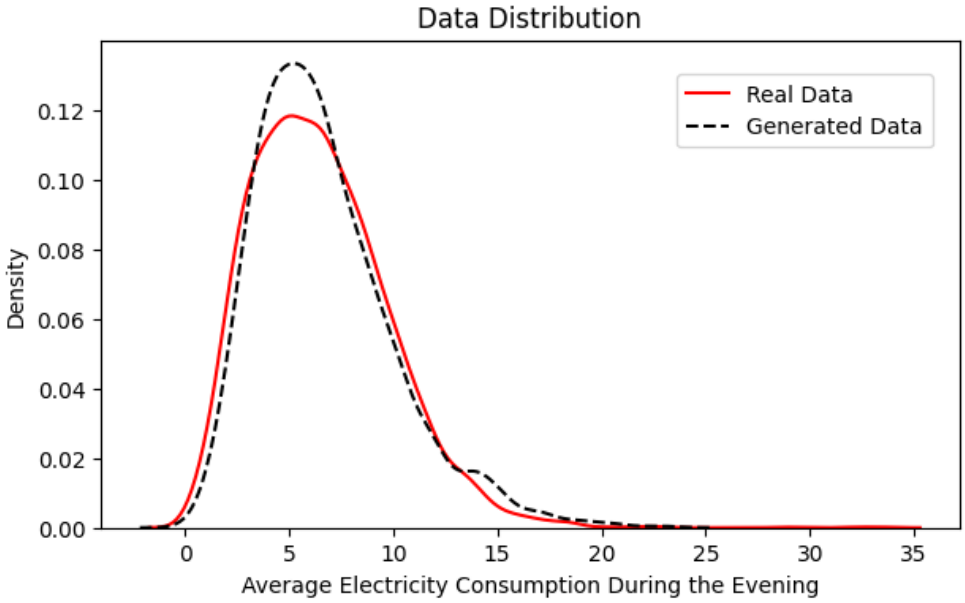

Figure 3.16 Data Density of 4096 Real Data Points vs 4096 Generated Data Points for Average Electricity Consumption During The Evening

The distributions of real and generated data for evening electricity consumption were compared, showing how similar they are in Figure 3.16.

The GAN model generated real numbers for the number of people, resulting in dissimilar data distributions as shown in Figure 3.17.

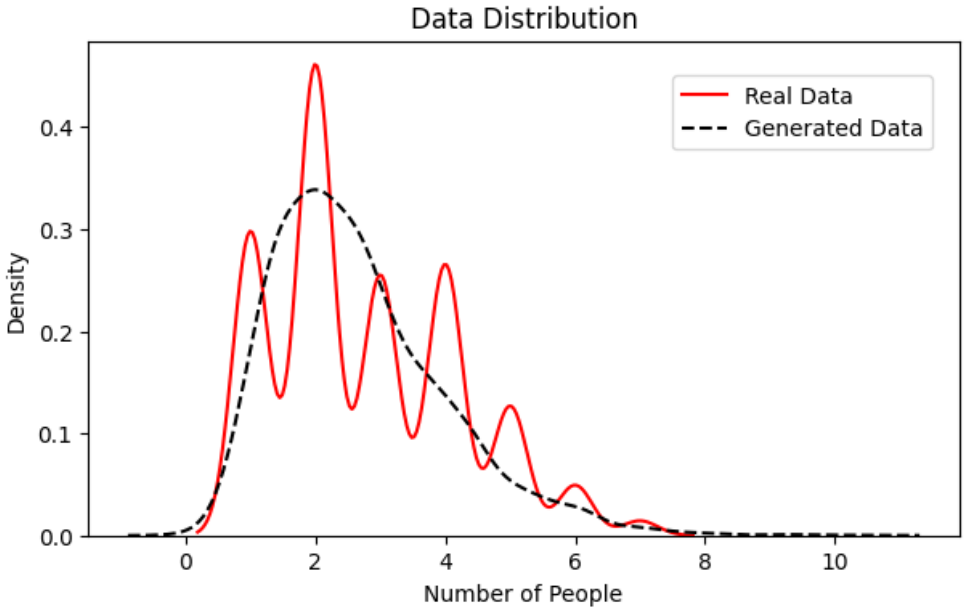

Figure 3.17 Data Density of 4096 Real Data Points vs 4096 Generated Data Points for the Number of People

After converting real numbers to integers, the distributions were compared once again. It was observed that the GAN model generated more data points for the class that contained three people in a household, as compared to the original dataset. The distributions appear to be quite similar, as displayed in Figure 3.18.

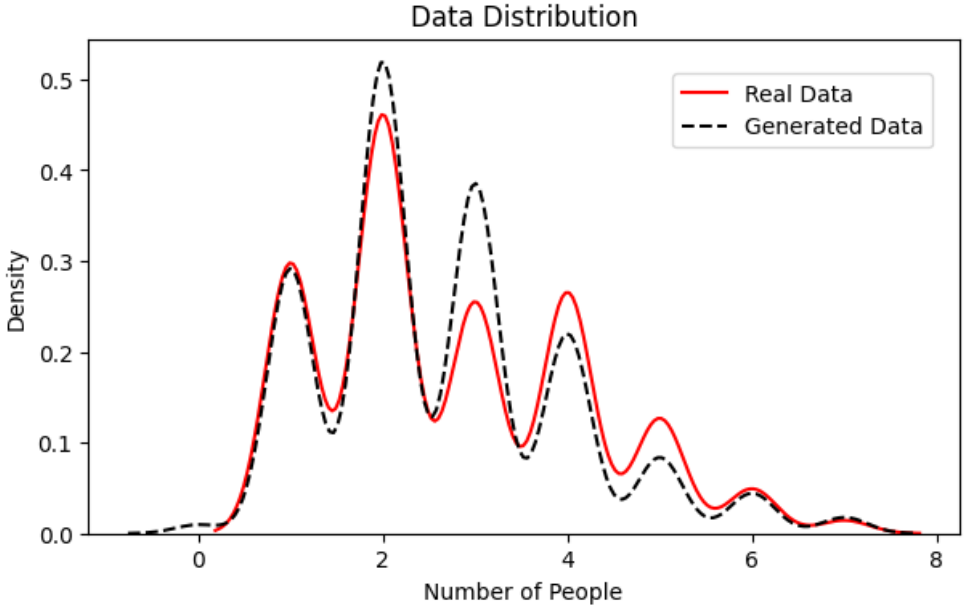

Figure 3.18 Data Density of 4096 Real Data Points vs 4096 Generated Data Points for the Number of People After Integer Conversion

3.4 Data Oversampling

Section 3.4.1 discusses the imbalance in data sets when models were used to classify whether someone is living alone or not in a household. The section also describes how the data imbalance was addressed by applying SMOTE. Similarly, Section 3.4.2 discusses the imbalance in data when models were used to classify the number of people in a household. This section also explains how the imbalance was eliminated after applying SMOTE. All performance metrics were compared before and after applying SMOTE. Please note that the performance metrics analyzed in this section do not take into account differential privacy. This means that no noise was applied to the data used by the machine-learning models.

3.4.1 Oversampling for Living Alone or Not Label

The original data set (without data generated by GAN) after processing contained data for 4,232 households, and it was unbalanced. For the label indicating whether someone lived alone or not, approximately 79.7% of individuals did not live alone, while only 20.3% of individuals lived alone, as shown in Figure 3.19. This resulted in a slight imbalance of approximately 4 to 1 in the original data set.

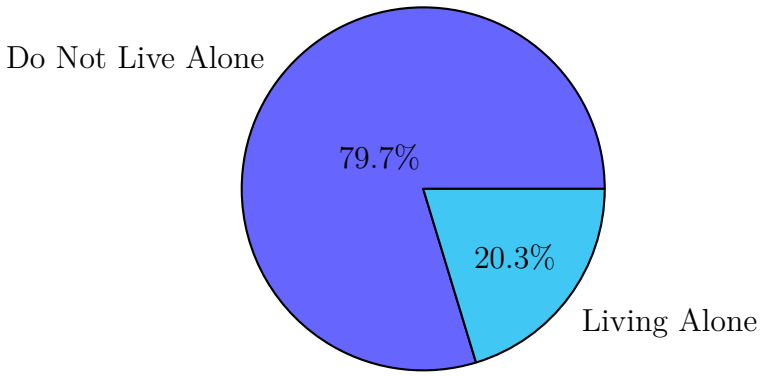

Figure 3.19 Living Alone or Not Label Distribution for 4,232 Households from the Original Data Set

The combined data set (original data set + data generated by GAN) after processing contained data for 8,322 households, and it was also unbalanced. For the label indicating whether someone lived alone or not, approximately 83% of individuals did not live alone, while only 17% of individuals lived alone, as demonstrated in

Figure 3.20. This resulted in a slight imbalance of approximately 5 to 1 in the combined data set.

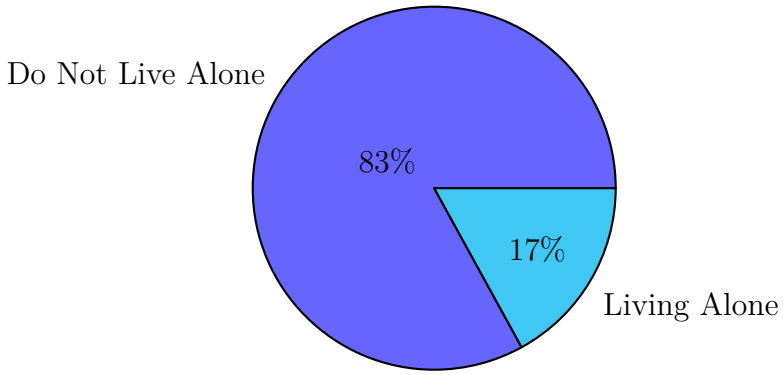

Figure 3.20 Living Alone or Not Label Distribution for 8,322 Households for the Combined Data Set

The Synthetic Minority Oversampling Technique (SMOTE) was used to address the class imbalance problem in the data sets. Unlike common oversampling techniques, SMOTE generates synthetic samples to over-sample the minority class instead of oversampling it with replacement. Synthetic samples are generated by operating in feature space rather than data space Chawla, Bowyer, Hall & Kegelmeyer (2002).

Initially, the original data set contained data for 4,232 households, with the majority class representing 3,373 households that had more than one person living in them. After applying SMOTE, the minority class samples were increased to 3,373 and matched the samples in the majority class, thus eliminating the imbalance in the data set. Moreover, SMOTE increased the overall number of samples from 4,232 to 6,746.

The combined data set included data for a total of 8,322 households. The majority of the households (6,908) had more than one person living in them. The minority class samples were increased to 6,908 after applying SMOTE, matching the majority class samples, and thus eliminating the imbalance in the data set. Additionally, SMOTE increased the overall number of samples from 8,322 to 13,816.

When using any oversampling method, the overall accuracy of the prediction model decreases. However, the model's ability to classify minority classes increases in accuracy. SMOTE generates synthetic samples, which may also create unrealistic samples and, as a result, reduce accuracy Fernández, García, Galar, Prati, Krawczyk & Herrera (2018).

For models that used the original data set and classified whether someone lived alone or not, the Gaussian NB model showed a decrease in accuracy from approxi-

mately 77.5% to around 73%. Meanwhile, the Logistic Regression model showed a decrease in accuracy from approximately 74% to around 58%. On the other hand, the Gaussian NB model showed an increase in the F1 score from approximately 40% to around 73%, while the Logistic Regression model showed an increase in the F1 score from approximately 26% to around 51%. This information is summarized in Table 3.6

Table 3.6 Performance Metrics for Living Alone or Not Classification Using Original Data (Pre-SMOTE vs Post-SMOTE)

Classifier	Accuracy		F1 Score	
	Pre SMOTE	Post SMOTE	Pre SMOTE	Post SMOTE
Gaussian NB	77.5%	73%	40%	73%
Logistic Regression	74%	58%	26%	51%

It was observed that the Gaussian NB model's accuracy decreased from around 80% to approximately 75% when classifying whether someone lived alone or not using the combined data set. Similarly, the Logistic Regression model's accuracy decreased from around 77.5% to approximately 61%. In contrast, the Gaussian NB model demonstrated a significant improvement in the F1 score, from around 45% to approximately 77%. Meanwhile, the Logistic Regression model only showed a slight increase in the F1 score, from approximately 55% to around 57%. You can find this data summarized in Table 3.7.

Table 3.7 Performance Metrics for Living Alone or Not Classification Using Combined Data (Pre-SMOTE vs Post-SMOTE)

Classifier	Accuracy		F1 Score	
	Pre SMOTE	Post SMOTE	Pre SMOTE	Post SMOTE
Gaussian NB	80%	75%	45%	77%
Logistic Regression	77.5%	61%	55%	57%

3.4.2 Oversampling for Number of People Label

For the label indicating the number of people living in a household in the original dataset, out of all the samples, approximately 20.3% of households had only one person, 31.7% had two people, 17.5% had three people, 17.7% had four people, 8.5% had five people, 3.3% had six people, and only 1% had seven or more people

living in the household. This information is represented in Figure 3.21. This resulted in a significant imbalance between the majority and minority classes, with an approximate ratio of 33 to 1.

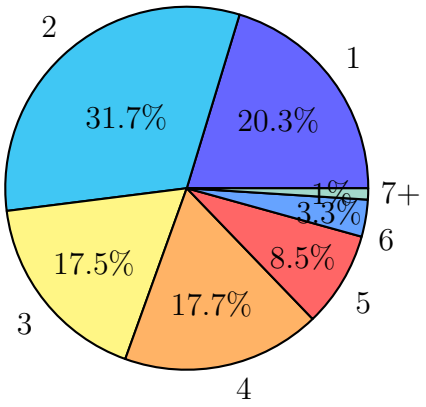

Figure 3.21 Number of People Label Distribution for 4,232 Households from the Original Data Set

Similarly, for the combined dataset, out of all the samples, approximately 17% of households had only one person, 30.6% had two people, 22.9% had three people, 17.6% had four people, 7.7% had five people, 3.2% had six people, and only 1% had seven or more people living in the household. This information is represented in Figure 3.22. This resulted in a significant imbalance between the majority and minority classes, with an approximate ratio of 33 to 1.

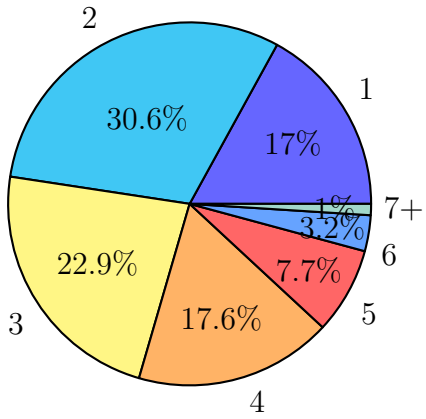

Figure 3.22 Number of People Label Distribution for 4,232 Households from the Combined Data Set

For the original data set, the majority of households had two people living in them, totaling up to 1340 households. After applying SMOTE, the number of samples in all classes increased except the majority class to 1340, making them equal. As a result, the overall number of samples in the data set increased from 4,232 to 9,380.

In the combined data set, it was found that most households had a total of two individuals living in them, making it a majority class with 2,545 households. To balance the data set, SMOTE was applied, resulting in an increase in the number of samples in all classes except for the majority class, which remained at 2,545. As a result, the total number of samples in the data set increased from 8,322 to 17,815.

When using the original data set to classify the number of people living in a household, the Gaussian NB model showed a decrease in accuracy from about 35% to around 26%. Similarly, the Logistic Regression model showed a decrease in accuracy from about 19% to around 16%. The F1 score for the Gaussian NB model slightly increased from approximately 18% to approximately 21.5%. Additionally, the Logistic Regression model has also shown a slight increase in the F1 score from around 11% to around 14%, as shown in Table 3.8

Table 3.8 Performance Metrics for Number of People Classification Using Original Data (Pre-SMOTE vs Post-SMOTE)

Classifier	Accuracy		F1 Score	
	Pre SMOTE	Post SMOTE	Pre SMOTE	Post SMOTE
Gaussian NB	35%	26%	18%	21.5%
Logistic Regression	19%	16%	11%	14%

When the combined data set was used to classify the number of people living in a household, the Gaussian NB model's accuracy decreased from 35% to approximately 28%. Similarly, the Logistic Regression model showed a decrease in accuracy, dropping from approximately 21% to 16%. The Gaussian NB model's F1 score slightly increased from approximately 22% to approximately 25%. Additionally, the Logistic Regression model also showed a slight increase in the F1 score from around 14% to around 15%, as demonstrated in Table 3.9.

Table 3.9 Performance Metrics for Number of People Classification Using Combined Data (Pre-SMOTE vs Post-SMOTE)

Classifier	Accuracy		F1 Score	
	Pre SMOTE	Post SMOTE	Pre SMOTE	Post SMOTE
Gaussian NB	35%	28%	22%	25%
Logistic Regression	21%	16%	14%	15%

3.5 Differential Privacy Models

As previously mentioned, we used features extracted from electricity consumption data as inputs to various supervised models. Each feature represents a different query. Some of these models were utilized to classify whether a household was occupied by a single person or not, while others were used to classify the number of people residing in a household. We added noise to the data during classification to ensure differential privacy and assess its impact on model performance, specifically accuracy, and F1 score. The IBM Differential Privacy Library (diffprivlib) was used to achieve this. Diffprivlib is a Python library that is open-source and designed for differential privacy. Unlike the generic models found in Scikit-learn, Diffprivlib provides machine-learning models with differential privacy. Additionally, Diffprivlib offers mechanisms that add various types of noise Holohan, Braghin, Mac Aonghusa & Levacher (2019).

Two sets of tools were used to test the impact of differential privacy on the performance metrics of the models. The first set of tools included hybrid differential privacy machine-learning models. They are hybrid since they can classify data while adding noise to the data they are classifying to achieve differential privacy. During initialization, the privacy parameter of the machine-learning model was set to apply noise addition to the data being classified. Gaussian Naive Bayes (Gaussian NB) and Logistic Regression differential privacy machine-learning models were used.

The second set of tools contained noise-addition mechanisms, which are independent of the machine-learning models used for classification. The data underwent noise addition using three different noise-addition mechanisms: Gaussian, Geometric, and Laplace. This was achieved by setting the privacy parameter and the sensitivity parameter of the mechanism used during initialization.

We used two sets of data to conduct tests: the original dataset, which did not include data generated by GAN, and the combined dataset (original dataset + GAN-generated data). For the analysis utilizing hybrid differential privacy machine-learning models, we used Logistic Regression and Gaussian NB models. These models use epsilon (ϵ) as their privacy parameter. The value of ϵ ranges from 0 to 1, as shown in Table 3.10.

A total of eight tests were conducted using hybrid differential privacy machine-learning models. Four of these tests were used to classify whether an individual

Table 3.10 Privacy Parameters of the Hybrid Differential Privacy Machine-Learning Models

Model	**Privacy Paramter**	**Range**
Gaussian NB	Epsilon (ϵ)	[0, 1]
Logistic Regression	Epsilon (ϵ)	[0,1]

was living alone or not, while the remaining four were used to classify the number of people living in a household. Different combinations of machine-learning models and performance metrics were used in each test, as shown in Table 3.11.

Table 3.11 Test Combinations Using the Hybrid Differential Privacy Machine-Learning Models

Model	**Performance Metric**	**Label**
Gaussian NB	Accuracy	Living Alone or Not
Logistic Regression	Accuracy	Living Alone or Not
Gaussian NB	F1 Score	Living Alone or Not
Logistic Regression	F1 Score	Living Alone or Not
Gaussian NB	Accuracy	Number of People
Logistic Regression	Accuracy	Number of People
Gaussian NB	F1 Score	Number of People
Logistic Regression	F1 Score	Number of People

For the analysis utilizing noise-addition mechanisms, we used three noise-addition mechanisms: Gaussian, Geometric, and Laplace. All noise-addition mechanisms use epsilon (ϵ) as their privacy parameter, with ϵ ranging from 0 to 1. However, the Gaussian mechanism differs from the other two mechanisms as it takes an extra privacy parameter called delta (δ), which represents the probability of information accidentally being leaked. This information is summarized in Table 3.12.

Table 3.12 Privacy Parameters of the Noise-Addition Mechanisms

Mechanism	**Privacy Paramters**	**Range**
Gaussian	Epsilon (ϵ), Delta (δ)	(0, 1], (0, 1]
Geometric	Epsilon (ϵ)	(0, 1]
Laplace	Epsilon (ϵ)	(0, 1]

A total of 12 tests were performed using noise-addition mechanisms. The Gaussian NB model was used as the model for classification for all of the tests. For each mechanism, two tests were performed to classify whether an individual was living alone or not, and two were used to classify the number of people living in a household.

Eight features were used by the classifiers to do the classification. Each feature represented a different query. A total privacy budget of $\varepsilon = 8$ was used, and it was

divided equally among each feature. In each test, ε was varied using 50 different values in linear space. Since the models and noise-addition mechanisms are probabilistic, they produce different results each time they are run. Therefore, 50 runs were performed for each of the 50 different values of privacy parameters tested, and their averages were taken to obtain more stable results. When using noise-addition mechanisms, we set the sensitivity parameter of the mechanisms to 1 because the queries used were count queries. We divided the data sets used for analysis in all tests into 70% for training and 30% for testing the machine-learning models.

4. RESULTS

In this chapter, we present the results of the differential privacy analysis and provide the optimal privacy parameters for each machine-learning model and noise-addition mechanism. Optimal privacy parameters are the smallest values of privacy parameters at which performance metrics reach default values. A privacy budget of $\varepsilon = 8$ was equally divided among each feature. Each figure in this chapter shows performance metrics plotted against ϵ per feature. As mentioned in Section 3.5, we used two sets of tools to test the impact of differential privacy on the performance metrics of the models. All of the results displayed in this section show the classification done by models on the combined data after the application of SMOTE. Section 4.1 shows the results obtained when using hybrid differential privacy machine learning models, while Section 4.2 demonstrates the results obtained using noise-addition mechanisms. Section 4.3 discusses the findings.

4.1 Hybrid Differential Privacy Machine-Learning Models

The accuracy and F1 scores of the models were evaluated before adding any noise. In the case of predicting whether a person is living alone in a household, the Gaussian Naive Bayes model had an accuracy of around 74% and an F1 score of around 75%, while the Logistic Regression model had an accuracy of approximately 59% and an F1 score of approximately 57%. However, when predicting the number of people in a household, the Gaussian Naive Bayes model had an accuracy of around 28% and an F1 score of around 25%, whereas the Logistic Regression model had an accuracy of approximately 16% and an F1 score of approximately 15%.

Two different machine learning models, Gaussian NB, and Logistic Regression, were employed. 50 different values of ε were tested for both models in a linear space. For each of these 50 different values, the average values were computed across 50

different runs. Section 4.1.1 displays the results when models are classifying whether a person is living alone or not inside a household, while Section 4.1.2 displays the results when models are classifying the number of people inside a household.

4.1.1 Living Alone or Not Predictions

Figure 4.1 illustrates the accuracy of Gaussian NB against different values of ε per feature. The optimal value of ε per feature is approximately 0.36, at which point the accuracy converges to its default value (i.e., the accuracy without noise addition).

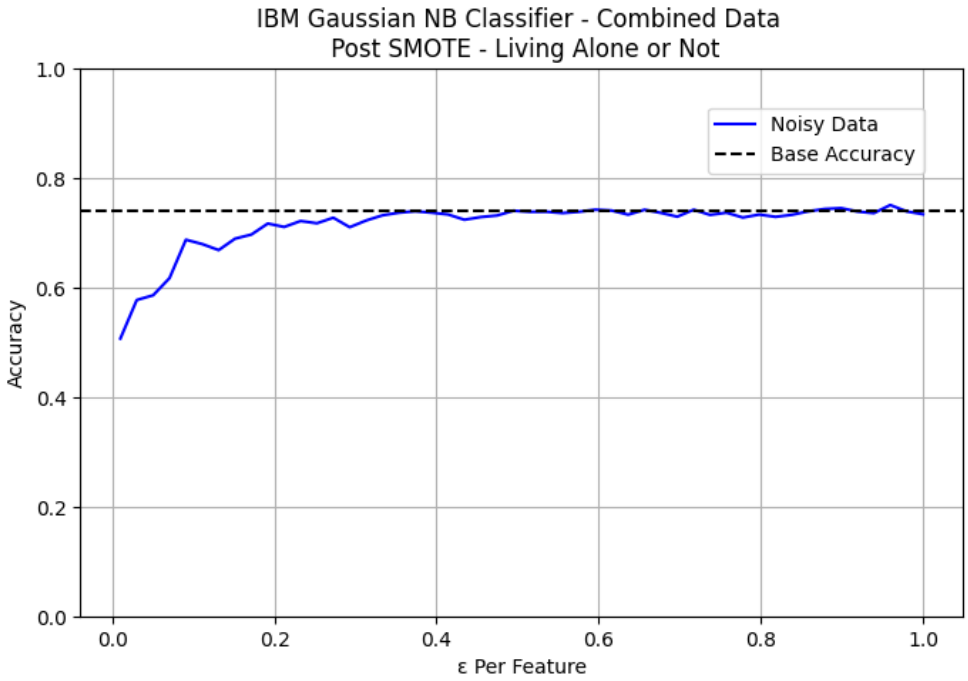

Figure 4.1 Gaussian NB: Living Alone or Not Predictions (Accuracy vs Epsilon Per Feature)

The optimal value of ε per feature is approximately 0.72, at which point the F1 score of Gaussian NB converges to its default value (i.e., the F1 score without noise addition), as can be seen in Figure 4.2.

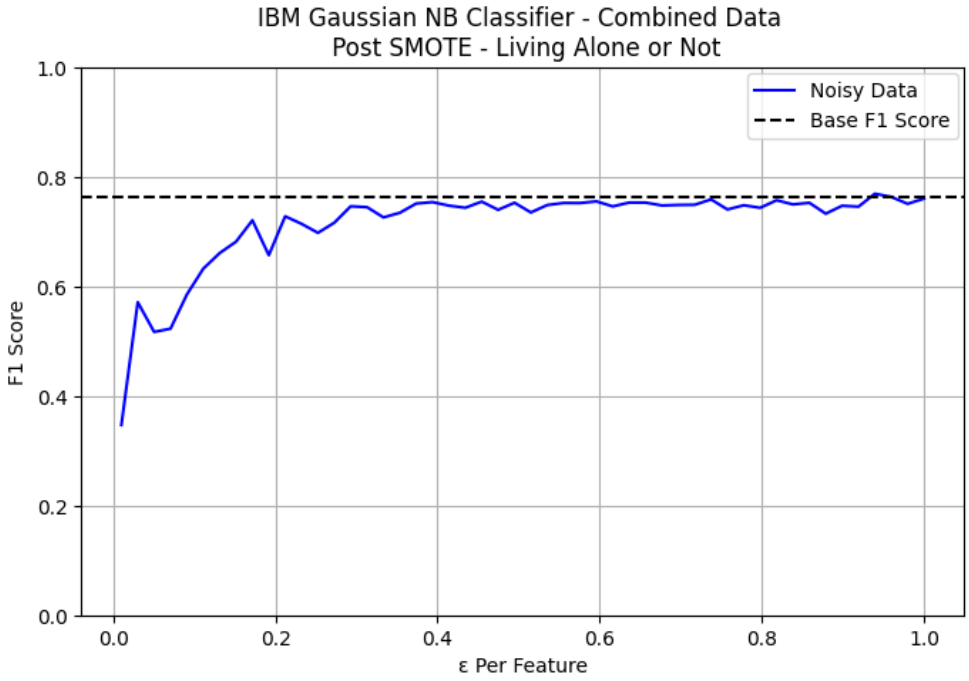

Figure 4.2 Gaussian NB: Living Alone or Not Predictions (F1 Score vs Epsilon Per Feature)

Figure 4.3 displays the accuracy of Logistic Regression for various values of ε per feature. The optimal ε per feature value is around 0.42, which results in the accuracy converging to its default value.

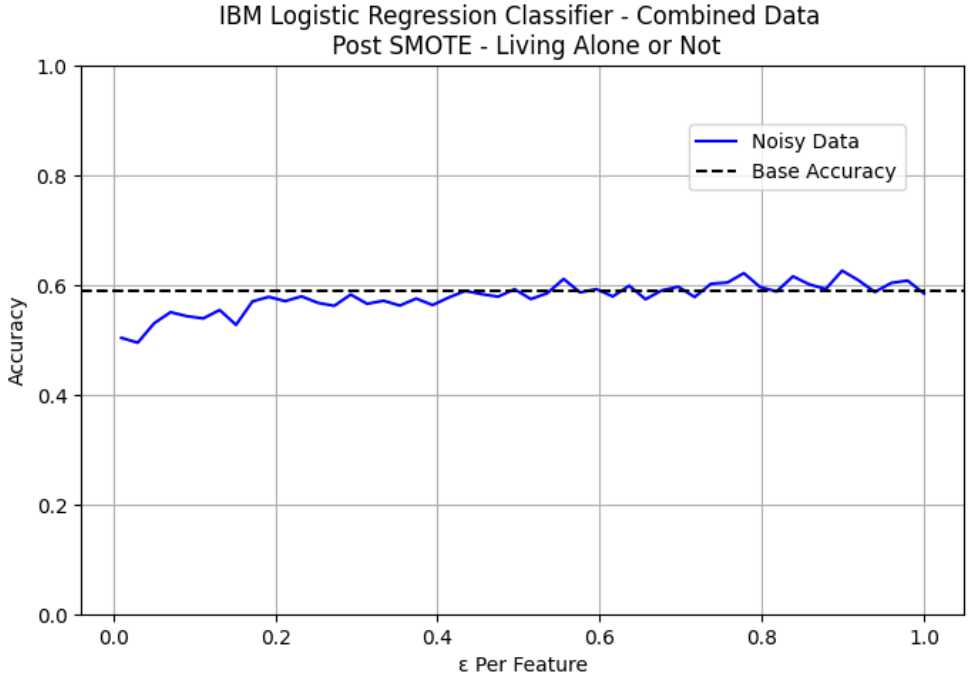

Figure 4.3 Logisitic Regression: Living Alone or Not Predictions (Accuracy vs Epsilon Per Feature)

The graph in Figure 4.4 shows the F1 score of Logistic Regression when tested against different values of ε per feature. It was discovered that when ε per feature is approximately 0.78, the F1 score converges to its default value, and this was taken as the optimal ϵ per feature value. Table 4.1 provides an overview of the results obtained when using hybrid differential privacy machine learning models when predicting whether an individual is living alone or not.

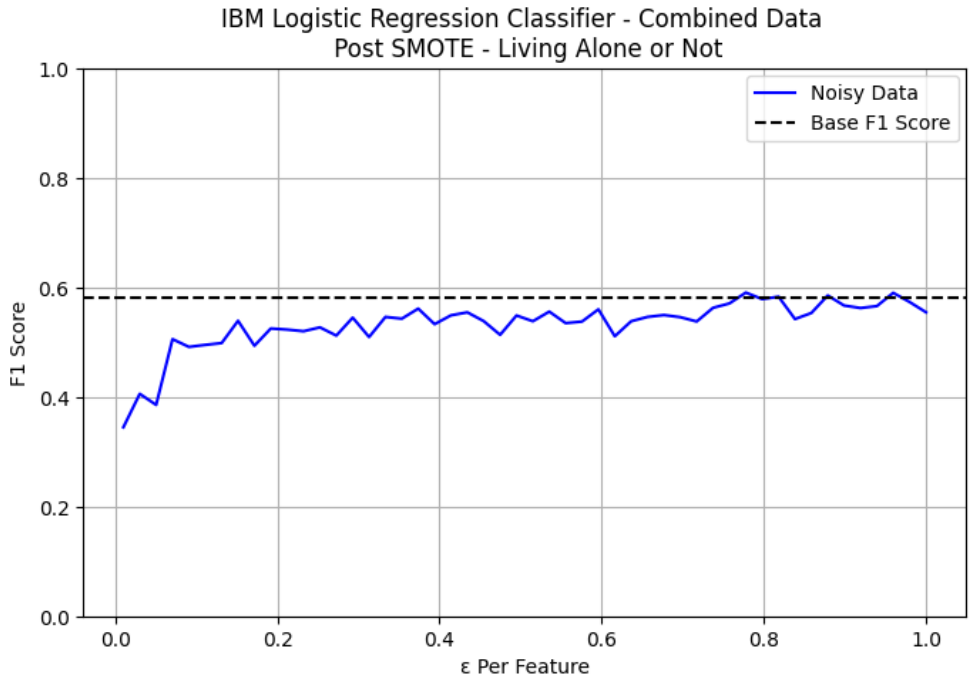

Figure 4.4 Logisitic Regression: Living Alone or Not Predictions (F1 Score vs Epsilon Per Feature)

Table 4.1 Overview of Results Using Hybrid Differential Privacy Machine Learning Models when Predicting Whether an Individual is Living Alone or Not

Model	Optimal Paramter	Performance Metric
Gaussian NB	$\varepsilon = 0.36$	Accuracy
Gaussian NB	$\varepsilon = 0.72$	F1 Score
Logistic Regression	$\varepsilon = 0.42$	Accuracy
Logistic Regression	$\varepsilon = 0.78$	F1 Score

4.1.2 Number of People Predictions

The accuracy of Gaussian NB, when tested against different values of ε per feature, reaches its default value when ε per feature is approximately 0.6, which is the optimal ε per feature value. This is demonstrated in Figure 4.5.

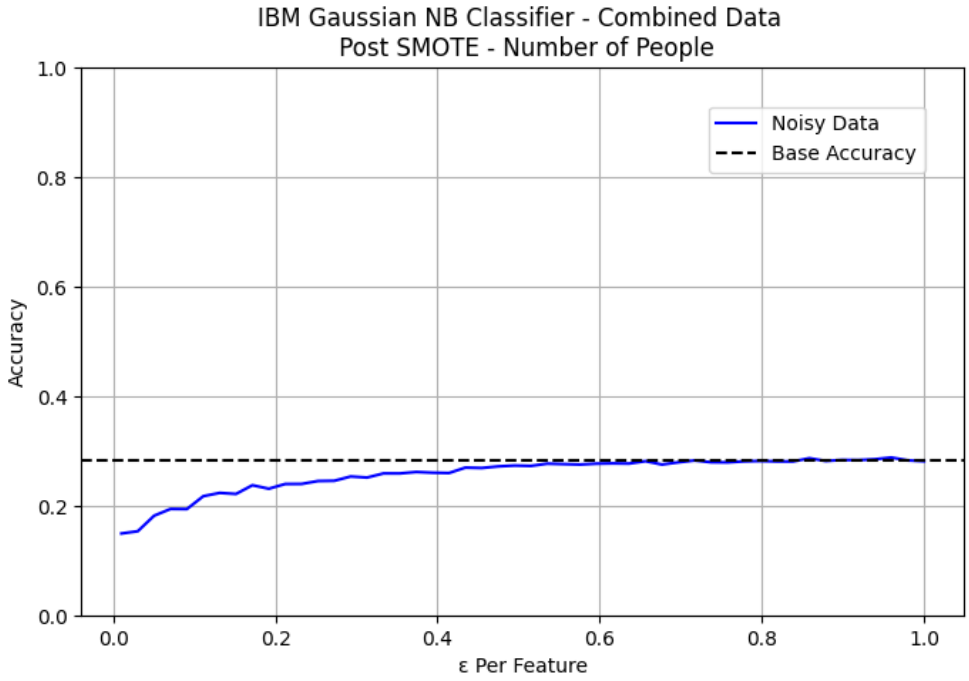

Figure 4.5 Gaussian NB: Number of People Predictions (Accuracy vs Epsilon Per Feature)

When tested against different values of ε per feature, the F1 score of Gaussian NB reaches its default value when ε per feature is approximately 0.62, which is the optimal ε per feature value as shown in Figure 4.6.

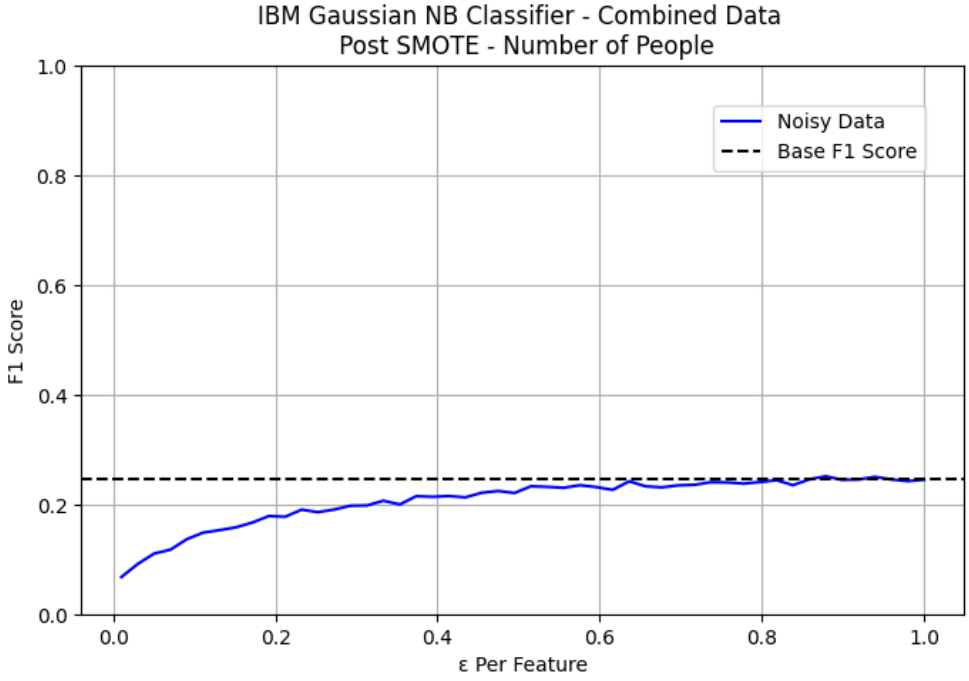

Figure 4.6 Gaussian NB: Number of People Predictions (F1 Score vs. Epsilon Per Feature)

The accuracy of Logistic Regression never reaches its default value when tested against different values of ε per feature. However, the optimal value of ε per feature can be taken as 0.74. At this ε per feature value, the accuracy is very close to its default value. This is shown in Figure 4.7.

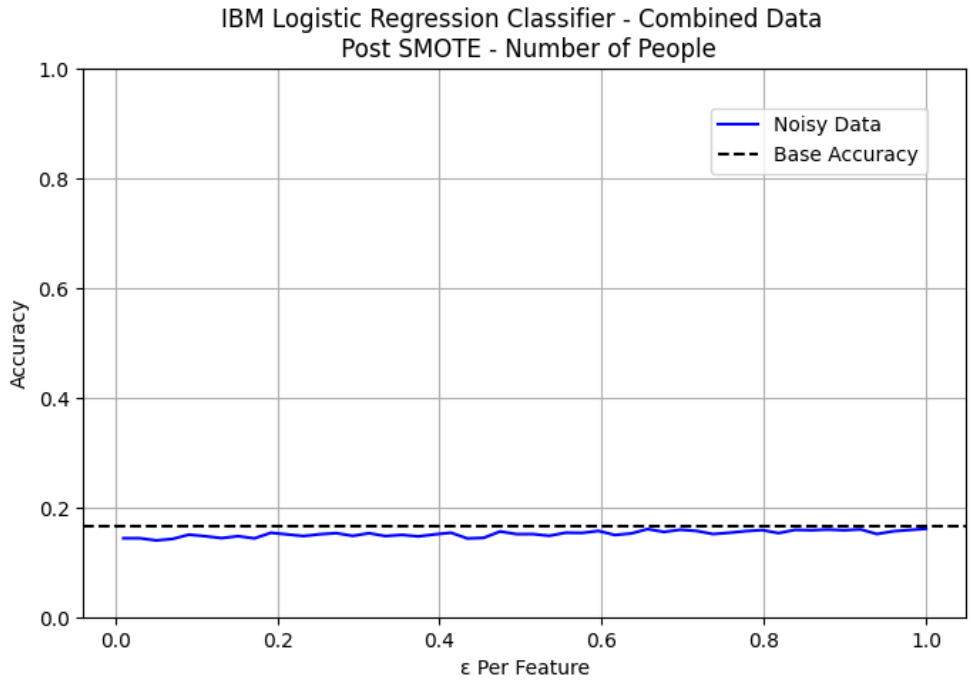

Figure 4.7 Logistic Regression: Number of People Predictions (Accuracy vs Epsilon Per Feature)

The F1 score of Logistic Regression reaches its default value when ε per feature is approximately 0.96. However, its optimal ε per feature can be taken as 0.76 since, at this ε per feature value, the accuracy is very close to its default value. Figure 4.8 displays this information. An overview of the results obtained using hybrid differential privacy machine learning models when classifying the number of people in a household is provided in Table 4.2

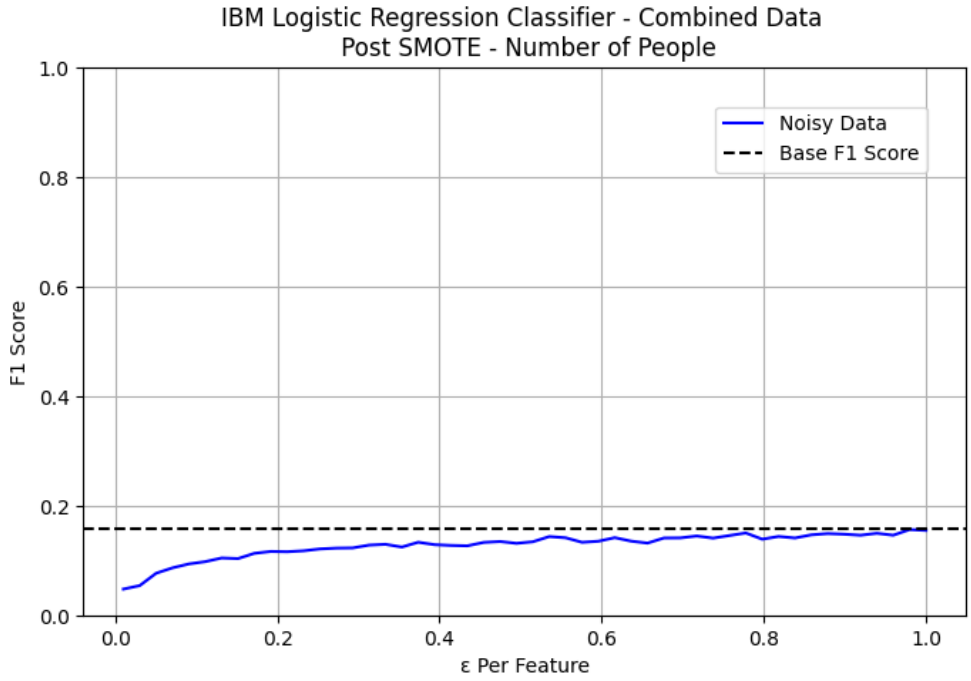

Figure 4.8 Logistic Regression: Number of People Predictions (F1 Score vs Epsilon Per Feature)

Table 4.2 Overview of Results Using Hybrid Differential Privacy Machine Learning Models when classifying the Number of People

Model	**Optimal Paramter**	**Performance Metric**
Gaussian NB	$\varepsilon = 0.6$	Accuracy
Gaussian NB	$\varepsilon = 0.62$	F1 Score
Logistic Regression	$\varepsilon = 0.74$	Accuracy
Logistic Regression	$\varepsilon = 0.76$	F1 Score

4.2 Noise-Addition Mechanisms

In this section, the Gaussian NB model was used for classification. When predicting whether a person is living alone in a household, the Gaussian Naive Bayes model achieved an accuracy of approximately 74% and an F1 score of approximately 75%. However, the model's accuracy dropped significantly to around 28% with an F1 score of around 25% when predicting the number of people residing in a household.

Three different noise-addition mechanisms were utilized: Gaussian, Geometric, and Laplace. In total, 50 different values of ε were tested for Gaussian, Geometric, and

Laplace mechanisms. For each of these 50 different values, the average values were computed across 50 different runs.

As mentioned earlier, the Gaussian mechanism requires an additional privacy parameter called delta (δ). To compare the Gaussian mechanism with other mechanisms, we determine the optimal value of delta and this topic is further discussed in Section 4.2.1. Section 4.2.2 displays the results when models are predicting whether a person is living alone or not inside a household, while Section 4.2.3 displays the results when models are predicting the number of people inside a household.

4.2.1 Gaussian Mechanism Optimal Delta

Delta (δ) indicates the probability of information accidentally being leaked. The closer the δ is to zero, the more noise is applied to the data. To find the optimal value of δ, we tested four different values: 0.25, 0.5, 0.75, and 1. For each of these delta values, we tested 50 different values of ε. For each of these 50 values, we computed the average values across 50 different runs. Four tests were conducted in total, with two used to determine if an individual lives alone and the other two used to determine the number of people residing in a household.

Figure 4.9 shows the Gaussian NB when classifying whether a person is living alone or not, where the accuracy of the classifier is being tested against ε per feature. It can be seen from the figure that when using $\delta = 1$, the classifier reaches its default accuracy with the smallest optimal ε per feature.

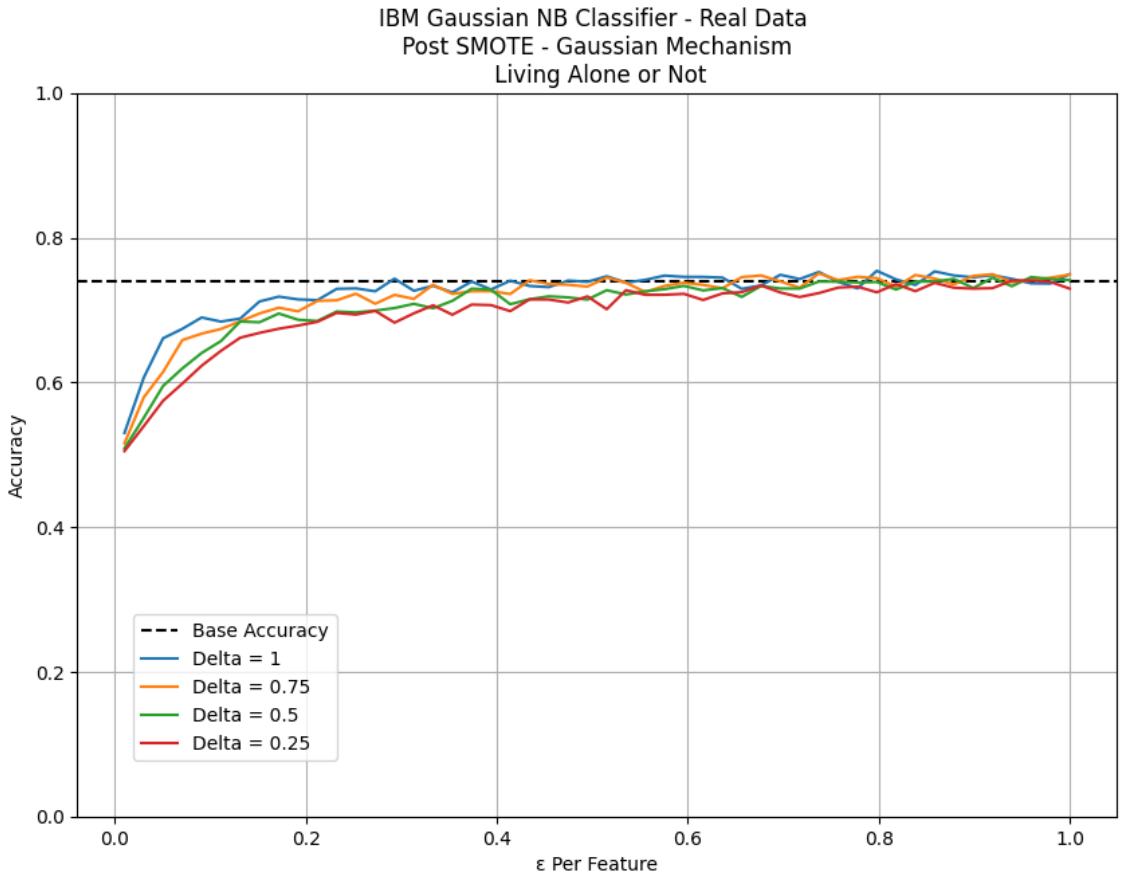

Figure 4.9 Gaussian NB: Living Alone or Not Predictions Using Different Delta Values (Accuracy vs Epsilon Per Feature)

In the given figure, denoted as Figure 4.10, the Gaussian NB model is used to classify whether a person is living alone or not. The F1 score of the classifier is being evaluated against ε per feature. From the figure, it can be observed that by using $\delta = 1$, the classifier achieves its default F1 score with the smallest optimal ε per feature.

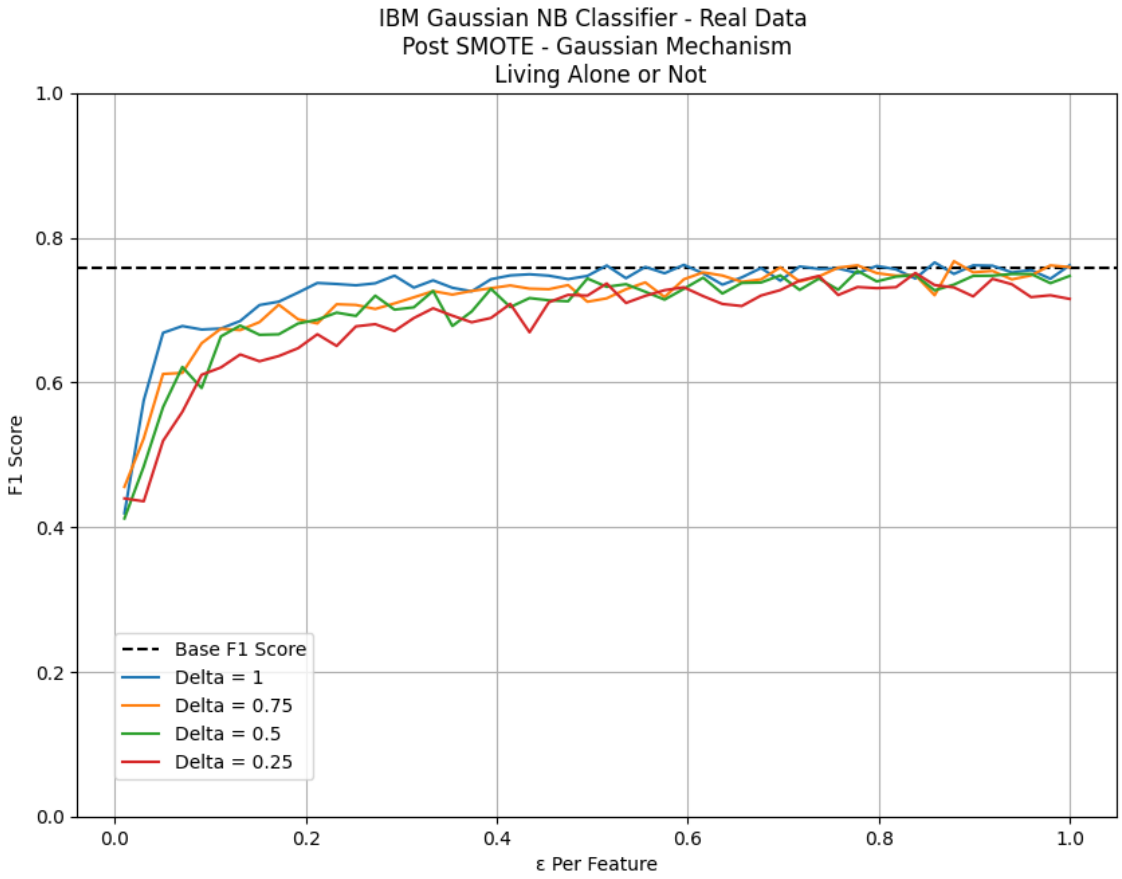

Figure 4.10 Gaussian NB: Living Alone or Not Predictions Using Different Delta Values (F1 Score vs Epsilon Per Feature)

The figure labeled as Figure 4.11 presents the use of the Gaussian NB model to classify the number of individuals residing in a household. The accuracy of the classifier is being assessed against ε per feature. The figure shows that when δ equals 1, the classifier reaches its optimal accuracy with the smallest ε per feature value.

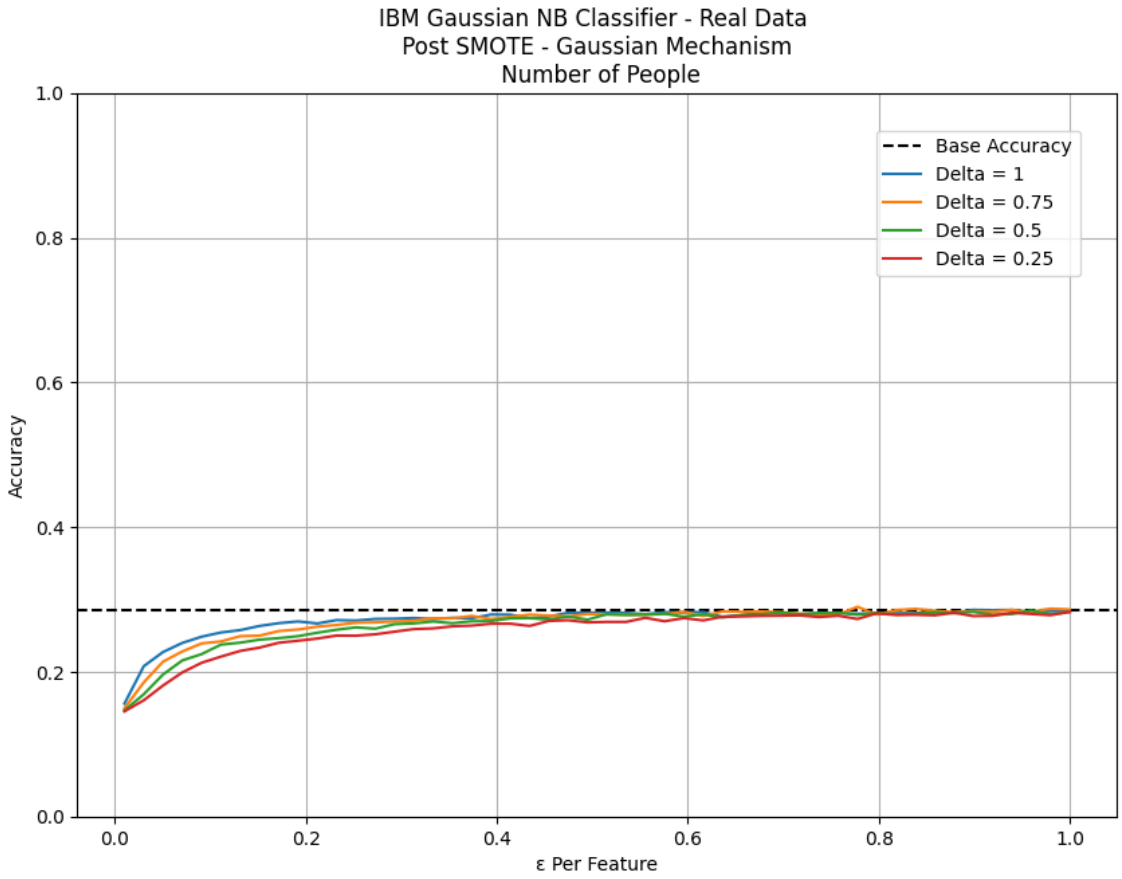

Figure 4.11 Gaussian NB: Number of People Predictions Using Different Delta Values. (Accuracy vs Epsilon Per Feature)

The classification of the number of people living in a household was carried out using the Gaussian Naive Bayes model. The F1 score of the classifier is being evaluated against the value of ε per feature, which is shown in Figure 4.12. The figure illustrates that the classifier achieves its default F1 score when δ equals 1, with the smallest possible value of ε per feature.

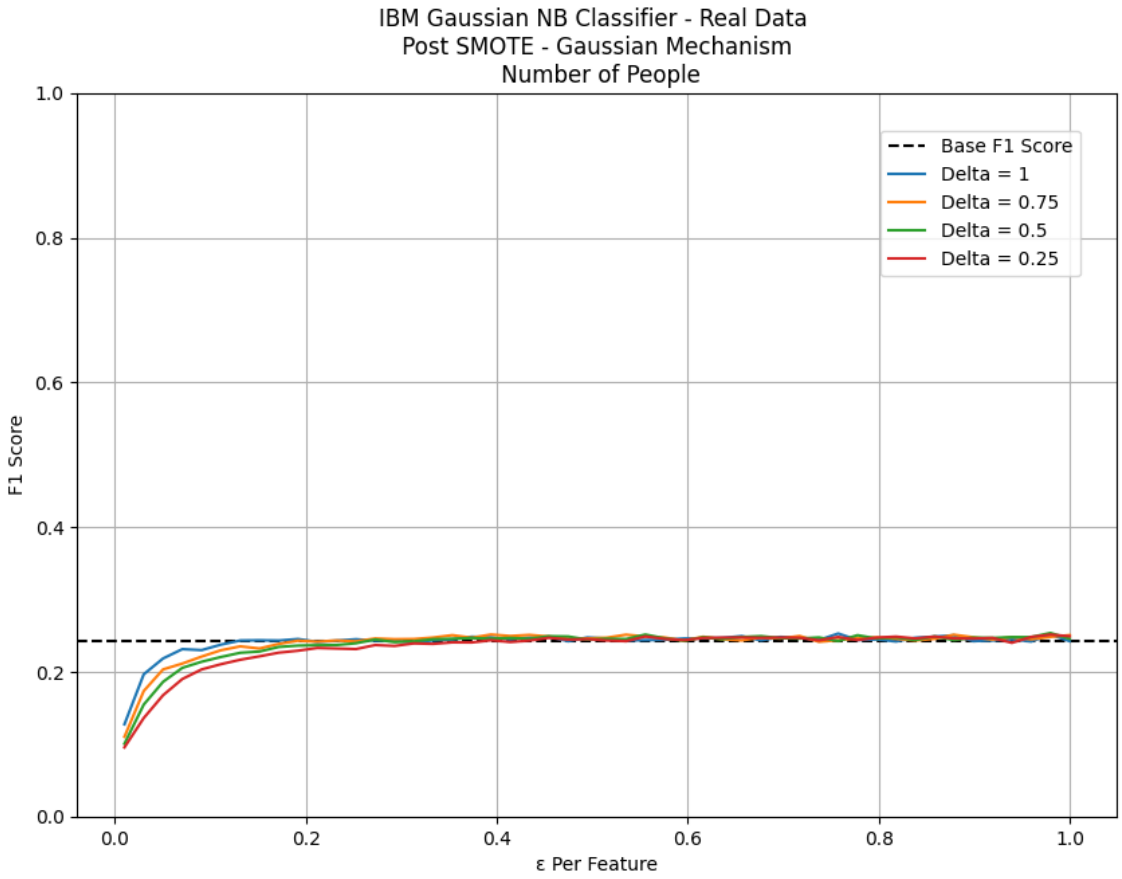

Figure 4.12 Gaussian NB: Number of People Predictions Using Different Delta Values (F1 Score vs Epsilon Per Feature)

It has been determined that the best value for delta (δ) is 1. This is because, in all four tests, the classifier achieves its default performance metric value with the smallest optimal ε per feature when using this δ value. Therefore, when conducting the tests to compare the Gaussian mechanism with other noise-addition mechanisms, $\delta = 1$ was used.

4.2.2 Living Alone or Not Predictions

Figure 4.13 displays the accuracy of Gaussian NB using Gaussian Mechanism with varying values of ε per feature. The optimal value of ε per feature is approximately 0.32, at which point the accuracy converges to its default value.

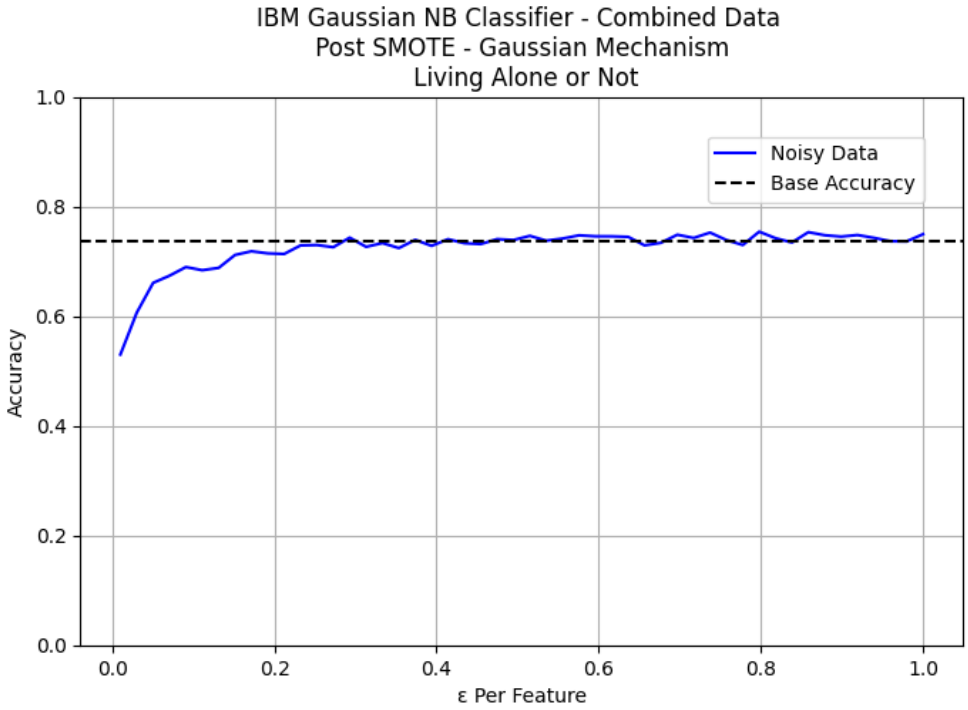

Figure 4.13 Gaussian NB: Living Alone or Not Predictions using Gaussian Mechanism (Accuracy vs Epsilon Per Feature)

In the given figure, labeled as Figure 4.14, we can observe the accuracy of Gaussian NB with Geometric Mechanism when subjected to different values of ε per feature. As per the Figure, the accuracy of the model converges to its default value when ε per feature is approximately 0.64.

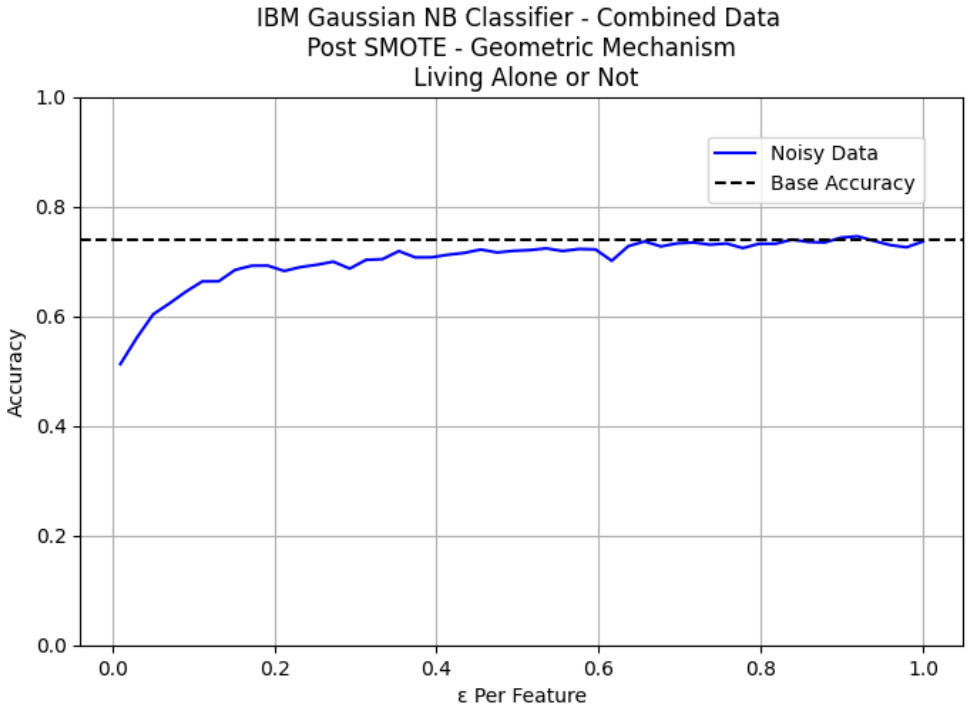

Figure 4.14 Gaussian NB: Living Alone or Not Predictions using Geometric Mechanism (Accuracy vs Epsilon Per Feature)

We tested the accuracy of Gaussian NB utilizing the Laplace Mechanism by varying the values of ε per feature, as shown in Figure 4.15. The outcomes suggest that the optimal value of ε per feature is approximately 0.6, which is the point where the accuracy converges to its default value.

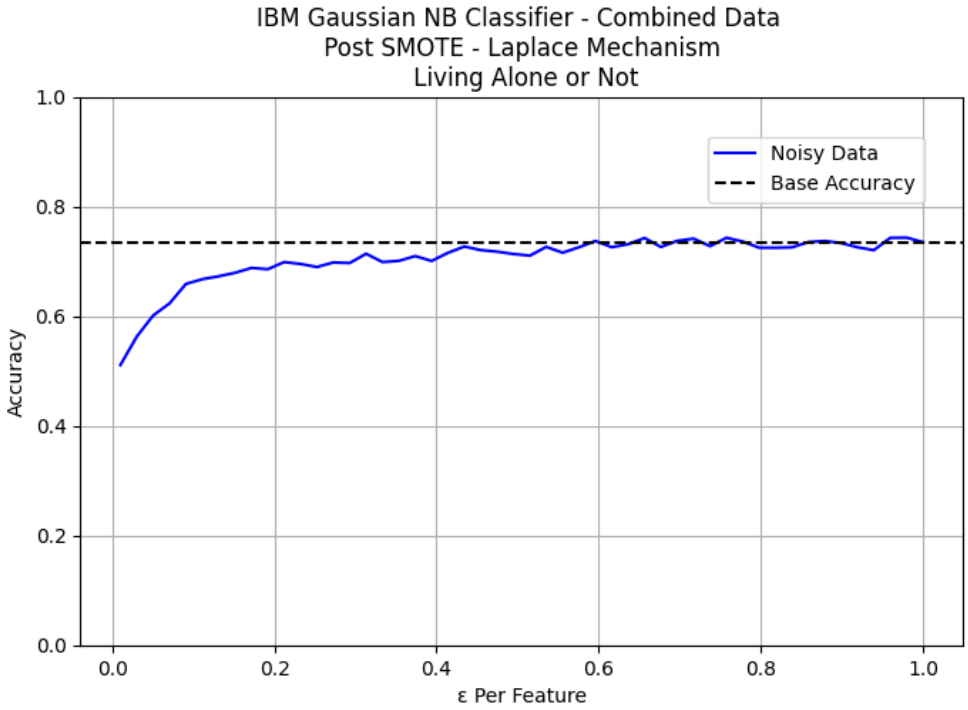

Figure 4.15 Gaussian NB: Living Alone or Not Predictions using Laplace Mechanism (Accuracy vs Epsilon Per Feature)

The F1 score of Gaussian NB using Gaussian Mechanism with varying values of ε per feature is displayed in Figure 4.16. The observed results indicate that the optimal value of ε per feature is approximately 0.28, at which point the F1 score converges to its default value.

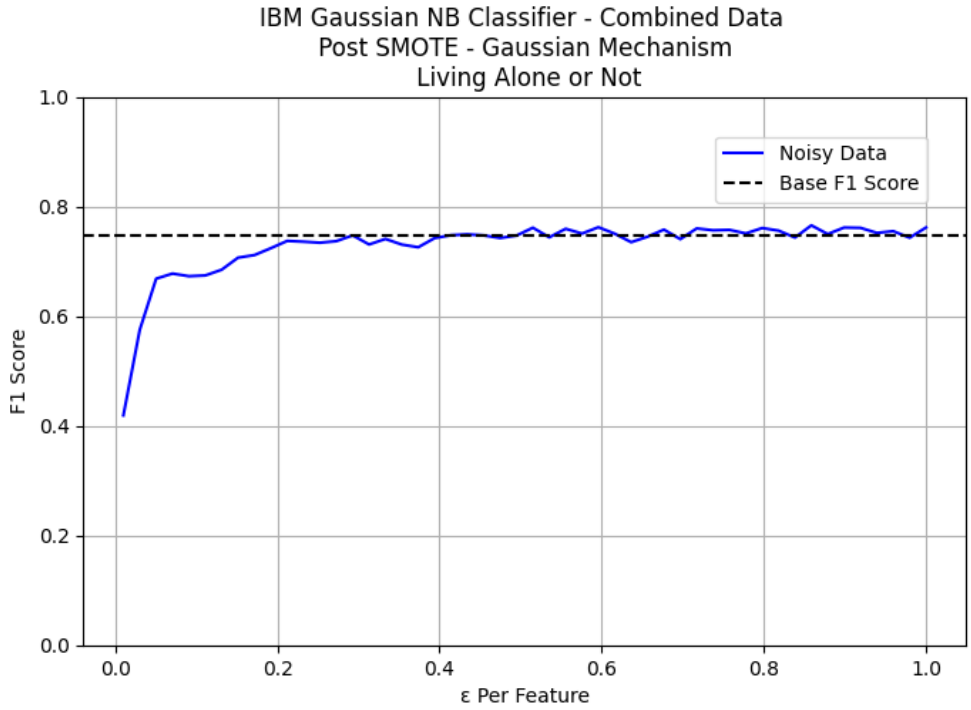

Figure 4.16 Gaussian NB: Living Alone or Not Predictions using Gaussian Mechanism (F1 Score vs Epsilon Per Feature)

The F1 score of Gaussian NB using Geometric Mechanism was tested with varying values of ε per feature. The optimal value of ε per feature is approximately 0.64, at which point the F1 score converges to its default value, as displayed in Figure 4.17.

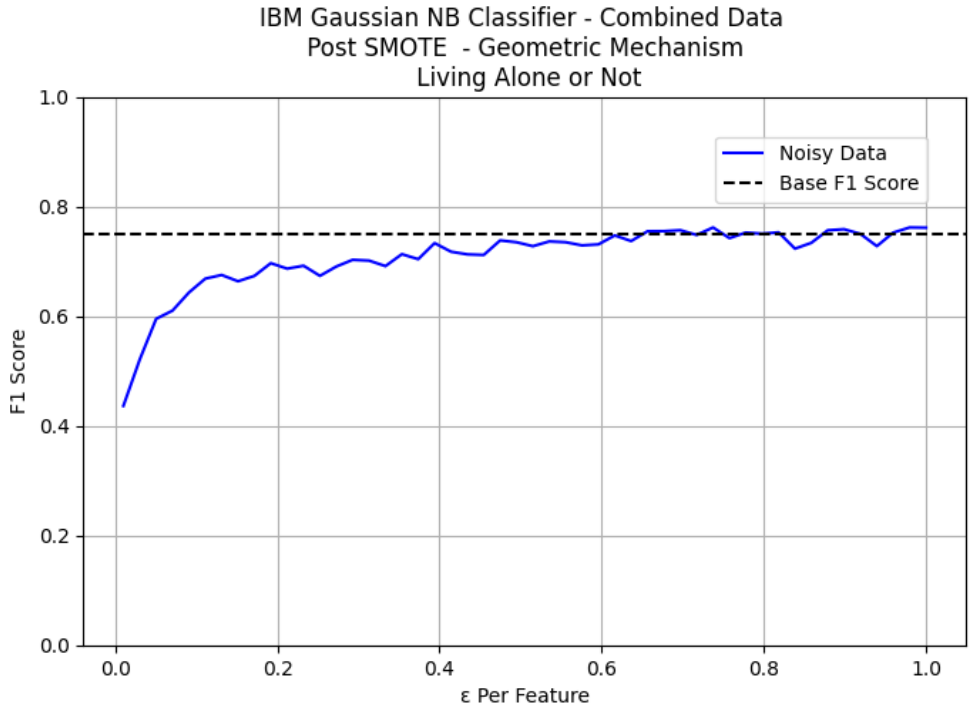

Figure 4.17 Gaussian NB: Living Alone or Not Predictions using Geometric Mechanism (F1 Score vs Epsilon Per Feature)

Figure 4.18 shows the F1 score of Gaussian NB using Laplace Mechanism, with varying values of ε per feature. The F1 score does not converge to its default value, but an optimal value of ε per feature can be selected at approximately 0.66. At this point, the F1 score is very close to its default value. Table 4.3 provides an overview of some of the results obtained using noise-addition mechanisms when predicting whether an individual is living alone or not.

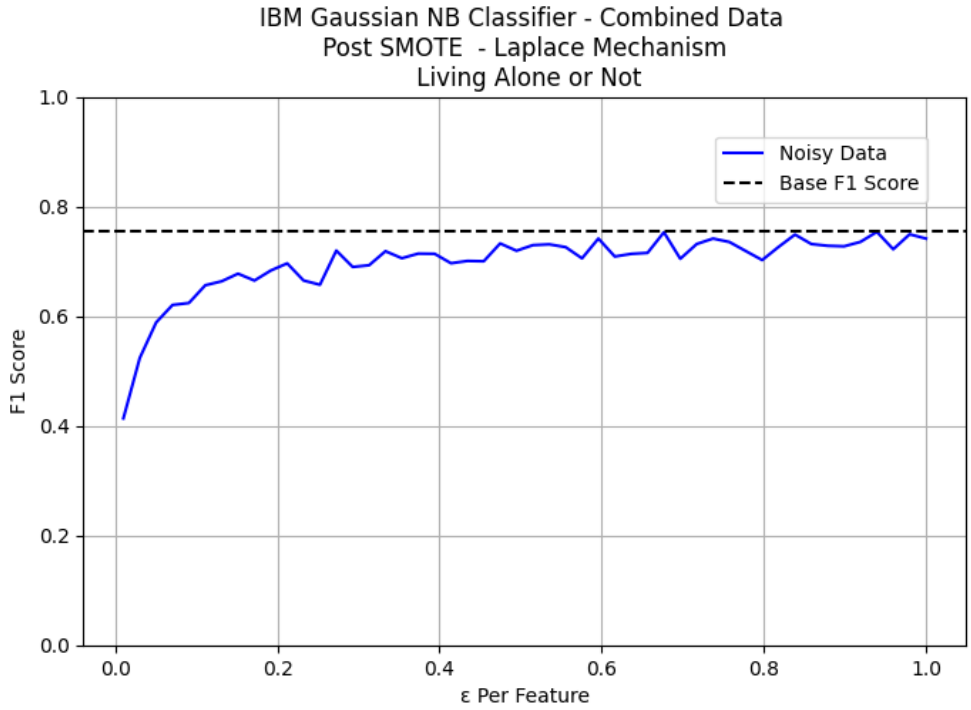

Figure 4.18 Gaussian NB: Living Alone or Not Predictions using Laplace Mechanism (F1 Score vs Epsilon Per Feature)

Table 4.3 Overview of Results Using Noise-Addition Mechanisms when Predicting Whether an Individual is Living Alone or Not

Mechanism	**Optimal Parameter**	**Performance Metric**
Gaussian	$\varepsilon = 0.32$	Accuracy
Geometric	$\varepsilon = 0.64$	Accuracy
Laplace	$\varepsilon = 0.6$	Accuracy
Gaussian	$\varepsilon = 0.28$	F1 Score
Geometric	$\varepsilon = 0.64$	F1 Score
Laplace	$\varepsilon = 0.66$	F1 Score

4.2.3 Number of People Predictions

Figure 4.19 shows the accuracy of Gaussian NB using Gaussian Mechanism with varying values of ε per feature. The accuracy converges to its default value at an optimal ε per feature of approximately 0.42.

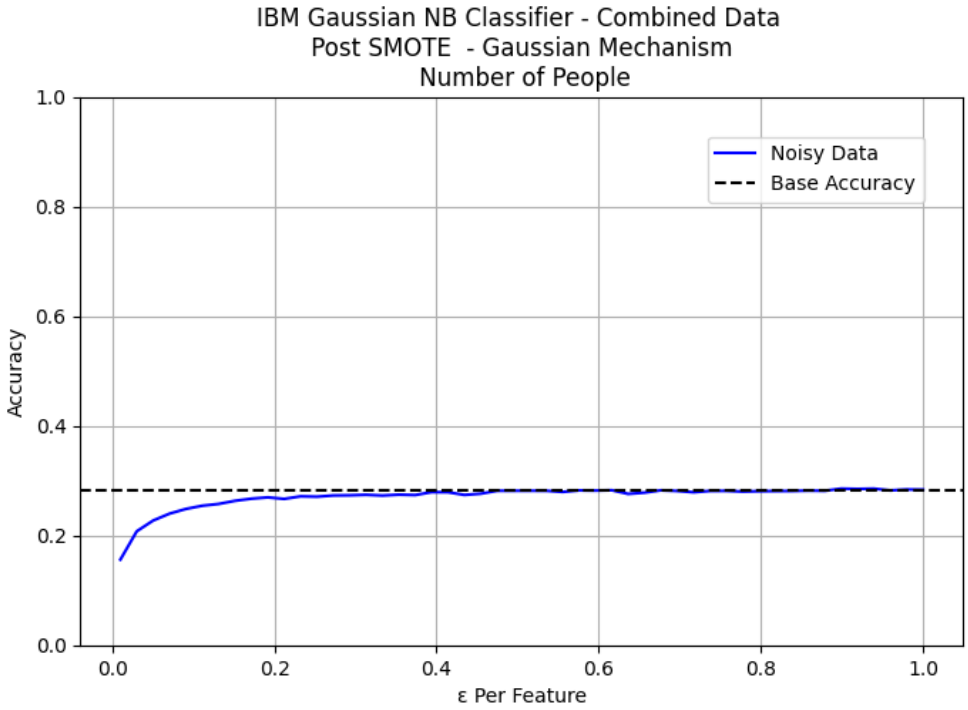

Figure 4.19 Gaussian NB: Number of People Predictions using Gaussian Mechanism (Accuracy vs Epsilon Per Feature)

The accuracy of Gaussian NB using Geometric Mechanism with varying values of ε per feature never converges to its default value, as shown in Figure 4.20. The optimal value of ε per feature can be selected at approximately 0.76. At this point, the accuracy is very close to its default value.

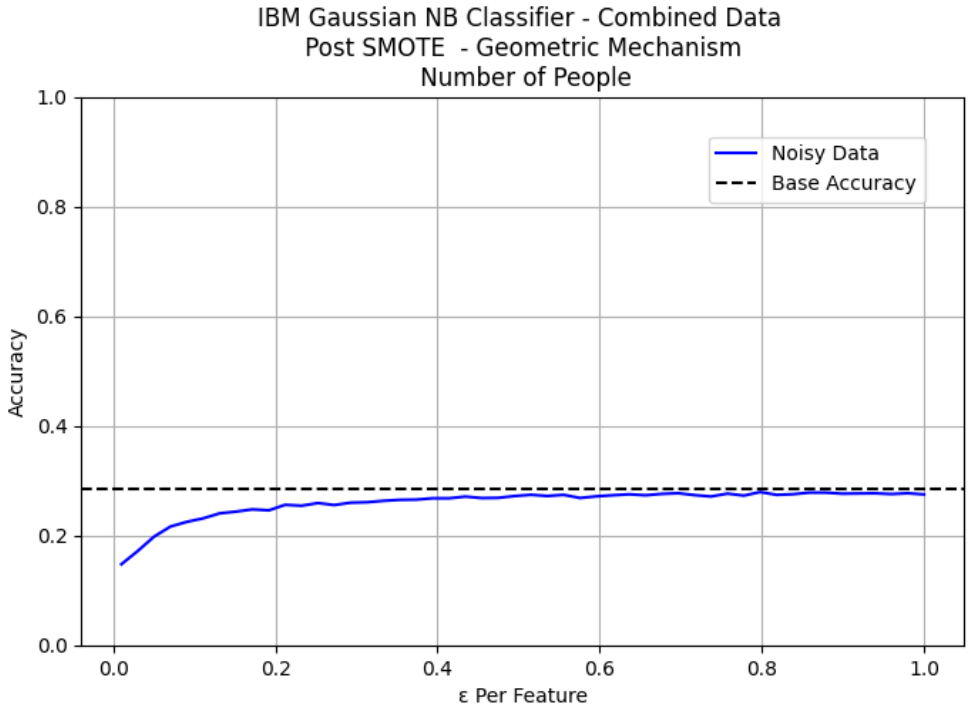

Figure 4.20 Gaussian NB: Number of People Predictions using Geometric Mechanism (Accuracy vs Epsilon Per Feature)

The accuracy of Gaussian NB when employing the Laplace Mechanism for noise addition, with varying values of ε per feature is presented in Figure 4.21. The Figure illustrates that the optimal ε per feature value is approximately 0.72, at which the accuracy reaches its default value

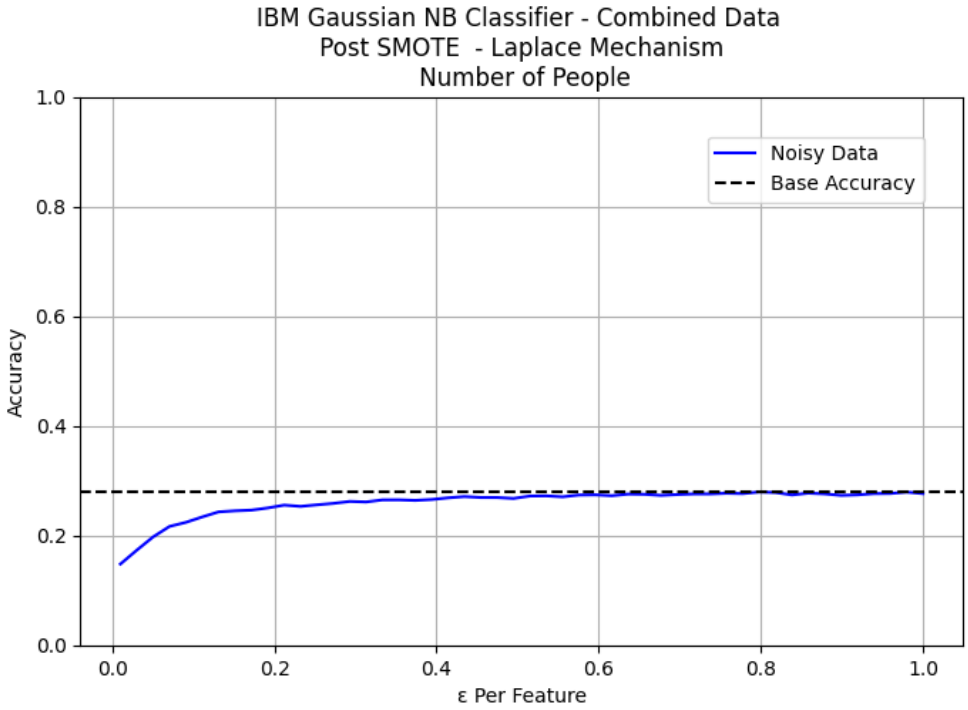

Figure 4.21 Gaussian NB: Number of People Predictions using Laplace Mechanism (Accuracy vs Epsilon Per Feature)

The Gaussian Mechanism was employed for noise addition to test the F1 score of Gaussian NB with varying values of ε per feature. Figure 4.22 shows the optimal value of ε per feature is approximately 0.12, at which the F1 score converges to its default value

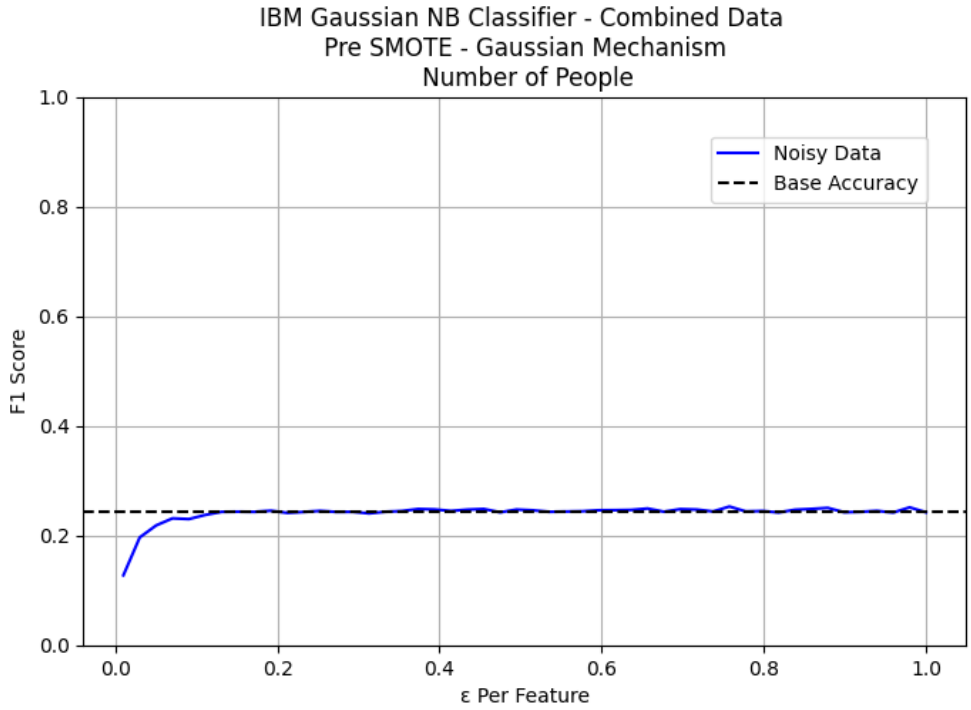

Figure 4.22 Gaussian NB: Number of People Predictions using Gaussian Mechanism (F1 Score vs Epsilon Per Feature)

The F1 score of Gaussian NB using the Geometric mechanism for noise addition was tested with varying values of ε per feature. The optimal value of ε per feature is approximately 0.24, at which point the F1 score converges to its default value, as demonstrated in Figure 4.23.

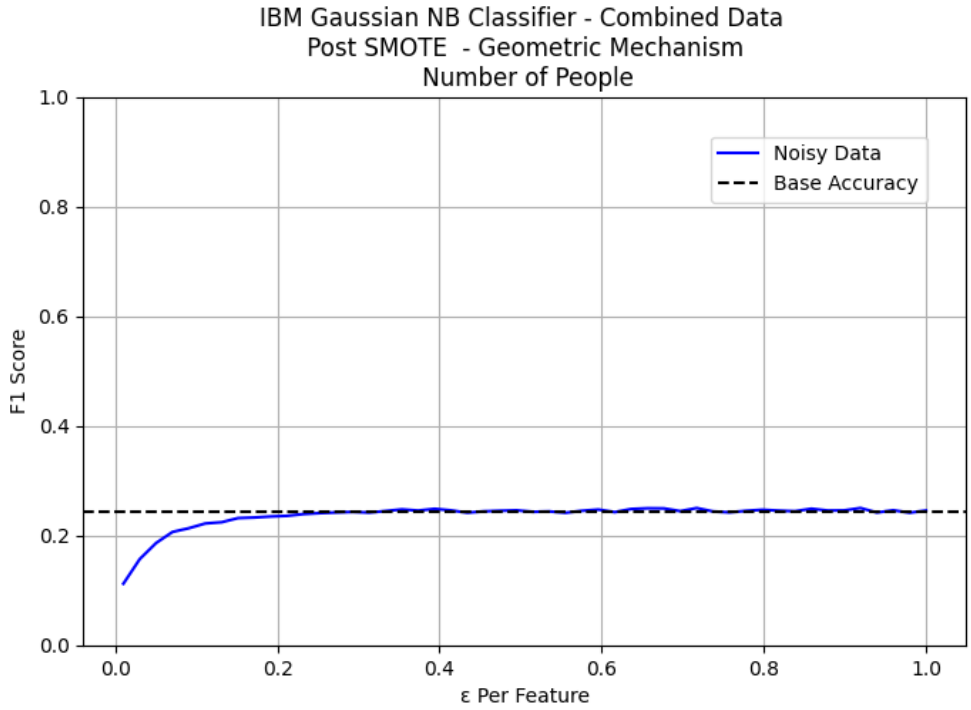

Figure 4.23 Gaussian NB: Number of People Predictions using Geometric Mechanism (F1 Score vs Epsilon Per Feature)

Gaussian NB's F1 score was tested with different values of ε per feature when the Laplace Mechanism was used for noise addition, as shown in Figure 4.24. The Figure indicates that the optimal value of ε per feature is around 0.26, at which point the F1 score converges to its default value. Results obtained using noise-addition mechanisms when predicting the number of people in a household are provided in table 4.4.

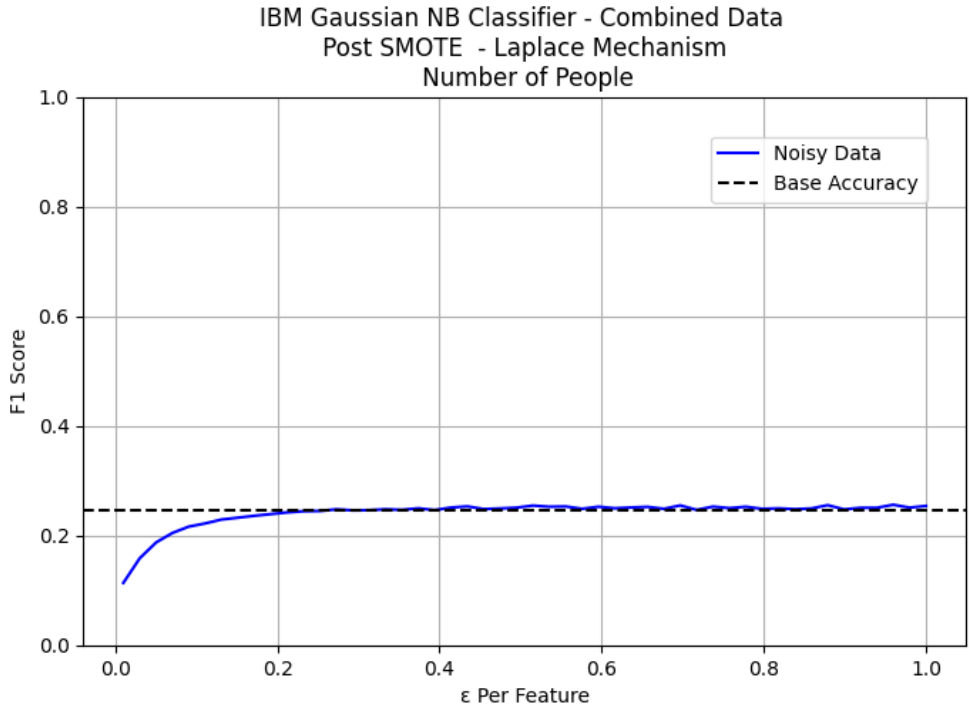

Figure 4.24 Gaussian NB: Number of People Predictions using Laplace Mechanism (F1 Score vs Epsilon Per Feature)

Table 4.4 Overview of Results Using Noise-Addition Mechanisms when Predicting the Number of People

Mechanism	**Optimal Parameter**	**Performance Metric**
Gaussian	$\varepsilon = 0.42$	Accuracy
Geometric	$\varepsilon = 0.76$	Accuracy
Laplace	$\varepsilon = 0.72$	Accuracy
Gaussian	$\varepsilon = 0.12$	F1 Score
Geometric	$\varepsilon = 0.24$	F1 Score
Laplace	$\varepsilon = 0.26$	F1 Score

4.3 Discussion of Results

The default values of accuracy and F1 score without any noise were used as a point of reference for comparison. For performance metrics being tested against privacy parameters, the expected behavior is that when the privacy parameter is small, the metric being measured should be somewhat lower than its default value and then gradually increase as the privacy parameter increases (i.e., the noise applied decreases).

The results obtained using hybrid differential privacy machine learning models show the Gaussian Naive Bayes model provides a better level of differential privacy than the Logistic Regression model. When tested against ε per feature, both the accuracy and F1 score start at much lower values compared to their default values, with higher levels of noise (i.e., at very low ε per feature), but gradually converge to their default values as ε per feature increases (i.e., lower noise). This is in line with expected behavior.

On the other hand, the results for the Logistic Regression model show a different behavior. The accuracy starts at a lower value, which is quite close to its default value, and gradually improves as ε per feature increases. The expected behavior was seen for the F1 score, where it starts at a much lower value compared to its default value and gradually converges to its default value as ε per feature is increased.

For the F1 scores in both models, it was seen that their F1 scores converged to their default values at high values of ε per feature in comparison to their accuracies. It was observed that the Gaussian NB model outperformed the Logistic Regression model in terms of accuracy and F1 score, and it had smaller optimal epsilon values for both metrics.

All of the noise-addition mechanisms used ε as their privacy parameter, but the Gaussian noise-addition mechanism had an extra privacy parameter δ. Before comparing the Gaussian noise-addition mechanism with other noise-addition mechanisms, four tests were conducted to determine the optimal value of δ. It was found that the optimal value of δ is 1.

The results from using noise-addition mechanisms show that, when compared to the other noise-addition mechanisms used, the Gaussian noise-addition mechanism is the best method for achieving differential privacy. The results for all mechanisms follow the expected behavior for both performance metrics. For both accuracy and F1 score, all noise-addition mechanisms start around the same initial values and then converge to their default values as ε per feature increases.

Smaller ε per feature values were taken as the optimal values since, for some tests, the metric being tested reaches its default value at high epsilon values ($\varepsilon > 0.8$). The point with the highest recorded metric for $0 < \varepsilon \leq 0.8$ was taken. The ε per feature value at this point was taken as the smaller optimal ε per feature value. The metric values at the smaller optimal ε per feature were very close to their default values.

When hybrid differential privacy machine learning models were used, the logistic regression classifier's accuracy failed to converge to its default accuracy while pre-

dicting the number of people living in a household. Similarly, when noise-addition mechanisms were used for noise addition, the F1 score of Gaussian NB using the Laplace mechanism failed to converge when predicting whether a person was living alone or not. Moreover, while predicting the number of people living in a household, the accuracy of Gaussian NB using the Geometric Mechanism never converged.

All results for the accuracy and F1 score of the Gaussian NB classifier follow the expected behavior when noise-addition mechanisms are used. Accuracies and F1 scores increase and converge to their default values as ε approaches 1.

5. CONCLUSION

This thesis proposes whether differential privacy can effectively balance household privacy with efficient data utilization and information extraction.SMOTE was used to address the data set imbalance, and synthetic data was generated using the GAN technique.

Differential privacy was achieved by adding noise to the data for noise addition, We analyzed the effect of noise addition on performance metrics like accuracy and the F1 score of the models. Two sets of tools were used to add noise. The first set included hybrid differential privacy machine learning models, while the second included noise-addition mechanisms.

Overall, we can conclude that data utilization is possible with small values of ε that provide better privacy with differential privacy. The use of these values ensures that the users' privacy is protected while still allowing for the effective use of the data.

The Gaussian NB model was found to provide a better level of differential privacy than the Logistic Regression model. It followed the expected behavior of performance metrics when tested against ε per feature. Moreover, the Gaussian NB model had smaller optimal ε per feature values for both accuracy and F1 score as compared to the Logistic Regression model.

It was also concluded that the Gaussian noise-addition mechanism is the best method for achieving differential privacy when compared to the other noise-addition mechanisms used. This is because, when using the Gaussian Naive Bayes model for classification, it was observed that the default metric values were achieved with smaller optimal ε per feature values than when using other noise addition mechanisms for both the accuracy and F1 score.

BIBLIOGRAPHY

Armel, K. C., Gupta, A., Shrimali, G., & Albert, A. (2013). Is disaggregation the holy grail of energy efficiency? the case of electricity. *Energy policy*, *52*, 213–234.

Beckel, C., Sadamori, L., & Santini, S. (2012). Towards automatic classification of private households using electricity consumption data. In *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, (pp. 169–176).

Beckel, C., Sadamori, L., Staake, T., & Santini, S. (2014). Revealing household characteristics from smart meter data. *Energy*, *78*, 397–410.

Chang, H.-H. (2012). Non-intrusive demand monitoring and load identification for energy management systems based on transient feature analyses. *Energies*, *5*(11), 4569–4589.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Cook, D., Schmitter-Edgecombe, M., Crandall, A., Sanders, C., & Thomas, B. (2009). Collecting and disseminating smart home sensor data in the casas project. In *Proceedings of the CHI workshop on developing shared home behavior datasets to advance HCI and ubiquitous computing research*, (pp. 1–7).

Cook, D. J. (2012). How smart is your home? *Science*, *335*(6076), 1579–1581.

Desfontaines, D., Mohammadi, E., Krahmer, E., & Basin, D. (2019). Differential privacy with partial knowledge. *arXiv preprint arXiv:1905.00650*.

Dwork, C. (2006). Differential privacy. In *International colloquium on automata, languages, and programming*, (pp. 1–12). Springer.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from imbalanced data sets*, volume 10. Springer.

Firth, S., Lomas, K., Wright, A., & Wall, R. (2008). Identifying trends in the use of domestic appliances from household electricity consumption measurements. *Energy and buildings*, *40*(5), 926–936.

Hamza, N., Hefny, H. A., et al. (2013). Attacks on anonymization-based privacy-preserving: a survey for data mining and data publishing.

Hofmann, M. & Siebenbrunner, T. (2023). A rich dataset of hourly residential electricity consumption data and survey answers from the iflex dynamic pricing experiment. *Data in Brief*, *50*, 109571.

Holohan, N., Braghin, S., Mac Aonghusa, P., & Levacher, K. (2019). Diffprivlib: the ibm differential privacy library. *arXiv preprint arXiv:1907.02444*.

Hua, N. (2016). E-commerce performance in hospitality and tourism. *International Journal of Contemporary Hospitality Management*, *28*(9), 2052–2079.

Irish Social Science Data Archive. Home, irish social science data archive.

Jain, S., Babu, S., Nair, A. R., & Sawle, Y. (2021). Smart metering: Transforming from one-way to two-way communication. *Active Electrical Distribution Network: A Smart Approach*, 573–595.

Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kumar, P. V. & Reddy, V. R. (2014). A survey on recommender systems (rss) and its applications. *International Journal of Innovative Research in Computer and Communication Engineering*, *2*(8), 5254–5260.

Marques, J. F. & Bernardino, J. (2020). Analysis of data anonymization techniques. In *KEOD*, (pp. 235–241).

McLoughlin, F., Duffy, A., & Conlon, M. (2012). Characterising domestic electricity consumption patterns by dwelling and occupant socio-economic variables: An irish case study. *Energy and buildings*, *48*, 240–248.

Molina-Markham, A., Shenoy, P., Fu, K., Cecchet, E., & Irwin, D. (2010). Private memoirs of a smart meter. In *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*, (pp. 61–66).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, *12*, 2825–2830.

Pekey, M., Çelebi, Y. D., Anıl, C., & Levi, A. (2021). Private information inference of households from electricity consumption data. In *2021 International Balkan Conference on Communications and Networking (BalkanCom)*, (pp. 166–170). IEEE.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Revuelta Herrero, J., Lozano Murciego, Á., López Barriuso, A., Hernández de la Iglesia, D., Villarrubia González, G., Corchado Rodríguez, J. M., & Carreira, R. (2018). Non intrusive load monitoring (nilm): A state of the art. In *Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection-15th International Conference, PAAMS 2017 15*, (pp. 125–138). Springer.

Sadeghianpourhamami, N., Ruyssinck, J., Deschrijver, D., Dhaene, T., & Develder, C. (2017). Comprehensive feature selection for appliance classification in nilm. *Energy and Buildings*, *151*, 98–106.

Taylor, P. (2022). Amount of data created, consumed, and stored 2010-2020, with forecasts to 2025. *Statista. Available online: https://www. statista. com/statistics/871513/worldwide-data-created/(accessed on 24 October 2023)*.

Tina, G. & Amenta, V. (2014). Consumption awareness for energy savings: Nialm algorithm efficiency evaluation. In *2014 5th International Renewable Energy Congress (IREC)*, (pp. 1–6). IEEE.

Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X., & Wang, F.-Y. (2017). Generative adversarial networks: introduction and outlook. *IEEE/CAA Journal of Automatica Sinica*, *4*(4), 588–598.

Zeifman, M. & Roth, K. (2011). Nonintrusive appliance load monitoring: Review and outlook. *IEEE transactions on Consumer Electronics*, *57*(1), 76–84.

Zoha, A., Gluhak, A., Imran, M. A., & Rajasegarar, S. (2012). Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, *12*(12), 16838–16866.