ISSN 2347 - 3983
Volume 11. No.1, January 2023
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter021112023.pdf
https://doi.org/10.30534/ijeter/2023/021112023

NegML: A Privacy-Preserving Machine Learning Approach Based on Negative Database

Mayank Gupta[1], Surendra Gupta [2]
[1] M.E. Research Student, S.G.S.I.T.S., Indore, India, mayankitm1620@gmail.com
[2]Associate Professor, S.G.S.I.T.S., Indore, India, sgupta@sgsits.ac.in

Received Date: December 5, 2022 Accepted Date: December 28, 2022 Published Date : January 07, 2023

ABSTRACT

Machine learning has become an increasingly prominent subject in the age of big data. It has made significant advances in image identification, object detection, and natural language processing, among other areas. The initial aim of machine learning is to extract meaningful information from enormous amounts of data, which unavoidably raises privacy concerns. Numerous privacy-preserving machine-learning approaches have been presented so far. However, most of them suffer from significant improvements in efficiency or accuracy. A negative database (NDB) is a data representation that may safeguard data privacy by storing and exploiting the complementary form of original data. In this research, we provide NegML, a privacy-preserving machine learning approach based on NDB. Private data are first transformed to NDB before being fed into machine learning algorithms such as a Multilayer perceptron (MLP), Logistic regression (LR), Gaussian naive Bayes (GNB), Decision tree (DT), as well as Random forest (RF). NegML has the same computational complexity as the original machine learning algorithms without privacy protection. Experiment findings on heart illnesses, milk datasets, Car evaluation benchmark datasets and Blood fusion dataset show that the accuracy of NegML is equivalent to the original machine learning model in most circumstances, as well as the technique based on differential privacy.

Key words: Machine learning, Deep learning, Multilayer perceptron, Logistic regression, Gaussian naïve Bayes, Decision tree and Random forest, Negative database, original dataset, modified dataset.

1. INTRODUCTION

Machine learning (ML) has enormous potential to boost productivity. However, the data used to train ML models must be of high quality to provide decent results. Any ML algorithm performs admirably only when given massive amounts of ideal data for training. Many organizations collaborate to get such high-quality data. It is critical to protect data security, privacy, and profit-sharing while collecting data from many companies. Massive data collection is a vital enabler for Artificial Intelligence (AI) techniques, and Machine Learning (ML)[1][2], which is at the core of AI, leverages such data to construct predictive models. Nevertheless, collecting and using data to discover patterns are two wholly different things. Furthermore, it comes with several challenges that must be addressed by a person or an organization, including privacy issues such as data breaches, financial loss, and reputational harm. “Machine learning drives most of the most sensitive data processing, including search algorithms, recommender systems, as well as adtech networks” [3][4]. Privacy-preserving machine learning aims to bridge the gap between privacy and machine learning advantages. It is a crucial enabler for the privatization of collected data and the observance of data privacy rules in the manner prescribed. This research provides an introduction to the fundamental concepts behind privacy-preserving machine learning.
"Privacy-Preserving Machine Learning" is a method that prevents data leakage in machine learning algorithms using a step-by-step methodology. PPML allows a wide variety of privacy-enhancing measures to be used, enabling many input sources to train machine learning models collaboratively without disclosing their data in its raw form, as shown in Figure 1.

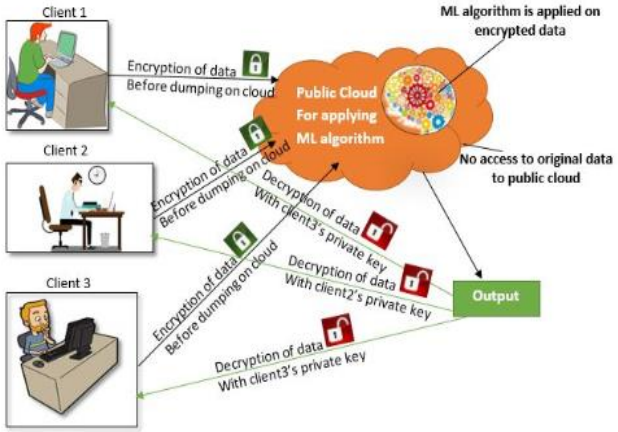

Figure 1: The concept of PPML

Machine learning has become an extremely prevalent subject in the age of big data. It has made significant advances in image identification, object detection, and natural language processing, among other areas. The initial aim of machine learning is to extract useful information from enormous amounts of data, which will undoubtedly raise privacy concerns that must be addressed. Present scenario [5][6], Several privacy-preserving Machine learning approaches have been presented. However, most of them suffered from a significant loss in either performance or accuracy. A negative database (NDB) is a data representation that may safeguard data privacy by storing and exploiting the complementary form of original data [7]. To ensure data privacy, NDB keeps it in a separate set of databases (DB) that can do standard database operations like insert, delete, update, and select. Recovering original data from a negative database is an NP-hard job. In addition, it allows for an approximate distance estimate. Due to these features, it may be used in various contexts to safeguard personal information. Nevertheless, the literature has not yet addressed the use of NDB for privacy-preserving Machine Learning[8].

This paper has been divided into five parts: The previous research is reviewed in Section 2. In Section 3, we talk about the research technique and strategy used in the recommended study. This was done so that we may better understand the work. The results of our efforts are shown in Section 4, along with a detailed breakdown of each result. Section 5 is where you can find the conclusion and suggestions for more study.

2. RELATED WORK

In the current era of big data, the analysis of large-scale datasets is often performed using classification algorithms [18] based on machine learning. The Extreme Learning Machine (ELM) classification algorithm is an emerging approach that uses a generalized single-layer feedforward network topology. Traditional ELM learning method inherently expects total access to the entire data collection. In most situations, this is a serious invasion of personal privacy. Due to safety considerations, sensitive information (such as medical records) cannot be shared. We present a privacy-preserving, efficient learning approach for ELM classification over data that has been vertically partitioned among multiple parties. The novel learning approach protects users' anonymity while working with numerical characteristics, creating a categorization model without releasing individual users' information.

This article[9] suggests a clinical diagnostic scheme that is both efficient & privacy-preserving, even when carried out by an (untrusted and malicious) third-party cloud service, so that medical professionals may get help with diagnoses without worrying about potentially damaging disclosures of patient or service provider information. Researchers provide a security model for multiclass support vector machines (SVM) in public health clouds and develop a clinical diagnosis scheme that protects patient privacy throughout the decision and diagnostic phases. To encrypt negative numbers, researchers suggest a new encoding method and design several security building blocks, including privacy-preserving decision function computing, privacy-preserving classification, as well as the search for the maximum decision function on encrypted fields, to enable the development of a privacy-preserving multiclass SVM for use in diagnostic techniques. Here, they detail the operational plan and the Dermatology test sheet. Analysis of security risks and experimental findings show that the suggested technique is effective and feasible for use in clinical diagnostic systems that protect patients' confidentiality.

This paper[10] presents a medical diagnostics method that protects patients' privacy and is based on multi-class support vector machines (SVMs). Both the distributed two trapdoors public key cryptosystem (DT-PKC) and the Boneh-Goh-Nissim (BGN) cryptosystem are the foundation for this approach. Researchers devise a safe computing protocol to carry out the computations required by the primary stage of the SVM classification method. This method can deal with data that can be separated linearly and data that cannot be separated linearly, and it's able to do so while maintaining the confidentiality of user data and support vectors. The findings demonstrate that our method is safe, dependable, highly accurate, and scalable.

This paper[11] specializes in functional encryption-based privacy-preserving deep neural networks. All functional encryption work is complicated as well as insecure. This study presents a way to calculate neural network activation functions using function-hiding inner product encryption (FHIPE). This is the first study on function-hiding inner product functional cryptography for machine learning privacy. Researchers also speed up functional encryption by calculating inner products. Relative to previous work in this sector, these experiments reveal 95x faster inner-product functional encryption-based safe activation and 10x faster FHIPE-based private activation.

This paper[12] offers PFMLP, a partly homomorphic encrypted federated learning framework for multi-party privacy-preserving machine learning. The essential principle is that all parties involved in the learning process must communicate the encrypted gradients via homomorphic encryption. Tests revealed that the PFMLP-trained model achieves almost the same accuracy, with the difference being less than 1%. As a solution to the computational burden of homomorphic encryption, researchers use an advanced Paillier method that may increase training speed by 25-28%. In addition, the article includes extensive discussions of comparisons, including encryption key length, the architecture of forms of collaboration, the number of learning clients, and so on.

In this study[13], researchers present a HE-friendly algorithm for the SVM training phase, avoiding wasteful operations and numerical instability in a secure environment. For real-time forecasting, the inference step is also conducted on the encryption process using fully homomorphic encrypting. Their studies on both synthetic and real-world datasets demonstrated that their HE-friendly approach beat the

state-of-the-art logistic regression classification using completely homomorphic encryption. This research is the first designers are aware of to provide a feasible approach for training an SVM model using completely homomorphic encryption. Consequently, research finding lends credence to further research into the privacy-preserving SVM model's potential for use in real-world settings.

Before the training phase of a huge volume dataset, this study discusses a k-Nearest Neighbor approach to construct models [15], applies an autonomous hyperparameter tuning method to select the ideal parameters based on the attributes, and then presents the results. Quality prediction as well as modeling performance evaluation based on high scores. To enhance the accuracy, practicality, correctness, and dependability of the scheme, we undertake several experiments using data from the actual datasets and the UCI machine learning repository.

This paper[16] presents a privacy-preserving machine learning algorithm for distributed hierarchies. Researchers increase collaborative learning. The suggested technique decreases learning overhead and protects each tier of a hierarchical distributed system. Focusing on the collaborative convergency in distinct learning groups, we also suggest an asynchronous technique to increase hierarchical decentralized network learning performance. Extensive studies on real-world data assess our suggested systems' privacy, effectiveness, and efficiency.

This paper[17] offers a technique for extracting HOG (histograms-of-oriented-gradients) features from encryption-then-compression (Etc) photos for privacy-preserving machine learning. Etc images are encrypted using a block-based approach described for Etc systems with JPEG compression. As of late, numerous industries have started adopting cloud computing as well as machine learning. However, users' privacy is at risk in the cloud owing to the unreliability of service providers and the possibility of accidents. Because of this, we offer a new block-based extraction approach of HOG features, which, under certain circumstances, allows us to run any machine learning algorithm without interference. To show the efficacy of the suggested technique, it is used for a face image identification issue in an experiment using two different classifiers (linear SVM and Gaussian SVM).

3. RESEARCH METHODOLOGY

This section provides the research methodology of this implementation work for privacy-preserving using different machine learning techniques with different datasets.

In this project, we have used two types of original and modified datasets. After that, we use Q-k hidden algorithm to convert it into a negative dataset and, using different machine learning models, find the accuracy based on accuracy. The project's prime objective is to effectively apply machine learning models to check the performance of the machine learning model on the original and a negative dataset. In the modified dataset, we have removed some columns in our original dataset. The removed columns are decimal values and binary values.

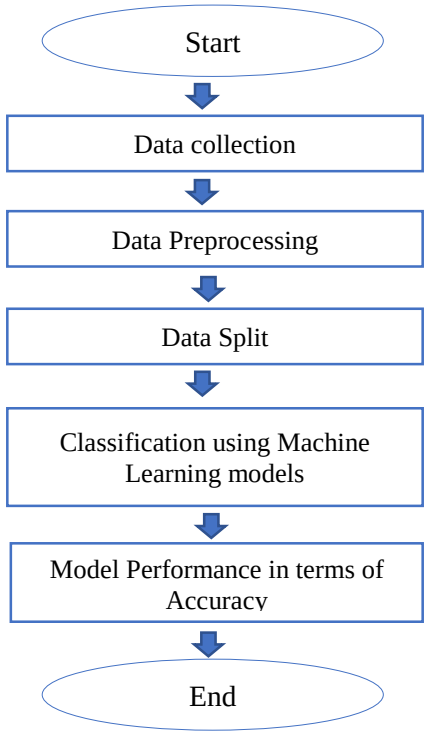

Figure 2: Proposed flowchart

The above figure 2 proposed flowchart shows the overall process of research work, as given figure steps are described below briefly:

3.1 Data Collection

For this work, we have used five types of a dataset collected from the Kaggle and UCI repository: heart disease, milk, car evaluation, and blood fusion datasets. These datasets are described below:

- **Heart Diseases Dataset:** There are 1120 rows and 12 columns in this dataset. They will predict, based on symptoms the person has heart disease or not
- **Milk dataset [1]:** There are 1059 rows and eight columns in this dataset. This dataset is used from Kaggle. In this dataset, they will predict the purity of milk
- **Car evaluation dataset[2]:** There are 1728 rows and six columns in this dataset. The Car Evaluation Database contains examples with the structural information removed, i.e., directly relates CAR to the six input attributes: buying, doors, persons, safety, etc. this database may be particularly useful for testing constructive induction and structure discovery methods.

- **Heart Diseases Dataset[3]:** There are 303 rows and 13 columns in this dataset. They will predict, based on symptoms the person has heart disease or not
- **Blood transfusion dataset[4]:** In this dataset, there are 748 rows and five columns all the values are integer value

3.2 Data Preprocessing

Data preprocessing is transforming raw data into a format that can be understood. It is a crucial phase in machine learning as well. Data preprocessing is a phase in the data mining and analysis process that turns raw data into a format that computers or machine learning can understand and evaluate. Furthermore, the gathered dataset eliminates null values, eliminates outliers, converts float values column to integer values using np floor techniques, and converts integer value to binary format. This procedure is required to properly apply Machine Learning Techniques to the dataset to get accurate results and predictions.

Step of original dataset:

1) First, upload the dataset
2) Pre-processing our dataset
    - Remove the null value
    - Remove Outliers
    - Describe a dataset
3) Hidden float value column into integer value by using the np floor method.
4) Convert integer value into binary form.
5) After converting the binary form, we can convert it into a negative dataset using the Qk-hidden algorithm.
6) Then apply the cross-validation (k-fold) machine learning model and find accuracy on the original and negative datasets.
7) Draw the comparison graph.

Step on the modified dataset:

1) First, upload the dataset
2) Pre-processing our dataset
    - Remove the null value
    - Remove Outliers
    - Describe a dataset
3) In the modified dataset, we have removed binary and floating value columns.
4) Convert integer value into binary form.
5) After converting the binary form, we can convert it into a negative dataset by using the Qk-hidden algorithm
6) After conversion, we added all the floating and binary columns.
7) Then apply the cross-validation (k-fold) machine learning model and find accuracy on the original and negative datasets.
8) Draw the comparison graph.

Negative Database Generation Algorithm: A negative database is a concise summary of the original database. This is the definition of the term. The original database is referred to as a positive database, whereas the compressed database is referred to as a negative database (NDB). In this study, the negative database was generated with the help of the QK-hidden method. Utilizing the QK-hidden method serves as an effective means of controlling the usage of the resulting NDB on a granular level.

3.3 Data Splitting

Here we split the dataset into two forms first training and second training. The training set contains 80% of the data and 20% of the testing set.

3.4 Classification Machine Learning Models

The Machine Learning Technique is implemented after the data has been prepared. We employ various categorization methods to protect our customers' privacy. The primary goal is to use Machine Learning techniques to evaluate these approaches' effectiveness and determine the degree to which they are accurate. In addition, we want to be able to identify the features that are mostly responsible for the accuracy of our predictions. The procedures are described below:

3.4.1 Multilayer Perceptron Model

A single neuron may be enough for a simple classifier or regression challenges. In a Multilayer Perceptron[19], MLP, the neurons, or perceptron's, are the fundamental units that are organized in layers, and here is where the actual potential of the ANN is unlocked. From the time an input is received by one layer, it is passed on to the next, and so on, until the output is formed. In figure 3, we see an MLP that is completely connected, meaning that each perceptron in the first layer is connected to those in the two layers so that their computed output may be shared. In a feedforward network, no output is transmitted from one perceptron to another in the same or higher layer. The black dots represent the five inputs into the example MLP. The third perceptron layer calculates its output and sends it to the final layer, which consists of two perceptrons that do identical calculations, yielding a 2 1 vector as the final output.

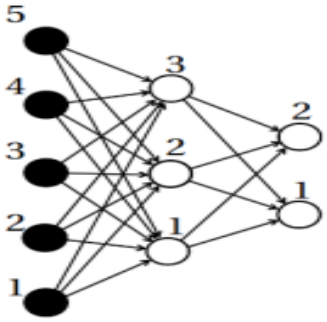

Figure 3: A Multilayer Perceptron taking fives inputs and generating a two × one output vector

3.4.2 Logistic Regression

Logistic regression[20] is a classification method that uses supervised learning. Estimating the likelihood of a binary response from one or more predictors is its primary usage. They may either be continuous or discrete. Data classification and differentiation tasks lend themselves well to logistic regression. Data is categorized as either "positive" (for diabetes) or "negative" (0 and 1), respectively. Logistic regression aims to best fit a model describing the connection between the outcome and predictor variables. The linear regression model is the foundation of logistic regression. An example of a sigmoid function used in a logistic regression model to forecast the probability of positive and negative class outcomes is shown below. P is a sigmoid function, defined as 1+1+e-(a+bx). Where P is the probability, a and b are Model parameters.

3.4.3 Decision Tree Classifier

The decision tree is a fundamental technique of classification. It is a way of learning via supervision. When the answer variable is categorical, the decision tree is utilized. The classification process may be described using the decision tree's tree-like structure-based model, depending on how input features are used. Input variables may be of any sort, including graphs, texts, discrete values, or continuous values [21]. The Decision Tree Algorithm's Steps and Procedures:

- Build the tree using nodes as the primary input feature.
- Determine which feature can best predict the output based on the input feature. This feature should have the largest information gain.
- A calculation is made to determine which character in each tree node provides the most information gain.
- Repeat step 2 to create a subtree by taking advantage of the feature that was avoided in the previous node's construction.

3.4.4 Gaussian Naïve Bayes

The expansion of the naive Bayes method is known as the Gaussian Naive Bayes method. One will have to determine the mean and standard deviation for the training data if you will utilize a Gaussian or normal distribution, the two functions that are the easiest to implement, even though other functions have been used to estimate data distribution. If the majority of the characteristics in a data collection are continuous, then the Gaussian Naive Bayes model is used. Within the context of this approach, it is assumed that the predictor values are chosen from a Gaussian distribution. As a result, the formula for conditional probability looks like this: -

$$P(x_i|y) = \frac{1}{\sqrt{2 \sqcap \sigma^2}} exp\left(-\frac{(x_i - \mu_y)}{2\sigma_y^2}\right) \dots\dots (1)$$

The mean μ_y and σ_y standard deviation of the predictor distribution is denoted by and respectively in this equation. [22].

3.4.5 Random Forest

This technique is a kind of ensemble learning and may also be used for classification and regression work. Compared to other models, the level of precision that it provides is far higher. This approach is quite adept at dealing with huge datasets. Leo Bremen is the creator of the Random Forest algorithm. It is a well-known method for learning in ensembles. By lowering the overall variance, Random Forest can enhance the Decision Tree performance. It functions by creating a large number of decision trees at the time of training, as well as it produces the class that corresponds to the mode of the classes or classification, as well as the mean prediction (regression) of the different trees[23][24].

Algorithm:

- The first stage is choosing the "R" features out of the "m" total features, where R has fewer features than M.
- The node that uses the optimal split point is part of the "R" characteristics.
- Determine the optimal way to divide the node into several subnodes.
- Repeat steps a to c until the "l" number of nodes has been achieved, then go on to step d.
- Created forest by doing steps a to d "a" time, which resulted in "n" number of individual trees being placed.

4. RESULTS AND DISCUSSION

The findings of the simulations and the experiments performed on the recommended model are presented in this report. To implement its implementation, this inquiry uses the Python Simulation Tool. Python is a framework for programming and numerical computing used by millions of researchers and technicians globally to analyze data, construct methodologies, or produce models. Python, a general-purpose programming language, and the Jupyter notebook environment are included in every experiment. The efficacy of the ML models will be discussed in the next section. In the course of our investigation, we have used three distinct datasets, namely the original, modified, and negative datasets. These datasets are being used to protect users' privacy.

Classification Accuracy: The most often used parameter for assessing classification models is classification accuracy. It is simple to compute and understand, and it is a single number that may encapsulate the model's capabilities, all of which contribute to the widespread usage of this metric. The ratio of the number of accurate predictions to the total number of samples that were inputted into the classification system is the classification accuracy. The proportion of a classifier's total accurate predictions expressed as a percentage when that number is calculated by dividing the number of occurrences represents the classifier's accuracy. Mathematically, suppose the accuracy of the classifier is deemed to be satisfactory. In

that case, the classifier may classify future data tuples in which the class label is known.

When analyzing classification models, accuracy is one parameter that may be used. Informally, accuracy refers to the percentage of correct predictions our model made. The preceding is the formally accepted definition of accuracy:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \ldots\ldots (2)$$

"Where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives".

4.1 Simulation results of Heart Disease Dataset – 1190 rows 12 columns

This section provides the simulation results of proposed machine learning classifiers using a heart disease dataset containing 1190 rows and 12 columns.

Table 1: Accuracy Performance of Heart disease dataset-1

Models	Original Dataset	Modified Negative Dataset	Negative Dataset
Multilayer Perceptron	81	71	75
Logistic Regression	78	73	76
Gaussian Naïve Bayes	76	72	74
Decision Tree	84	73	74
Random Forest	91	83	86

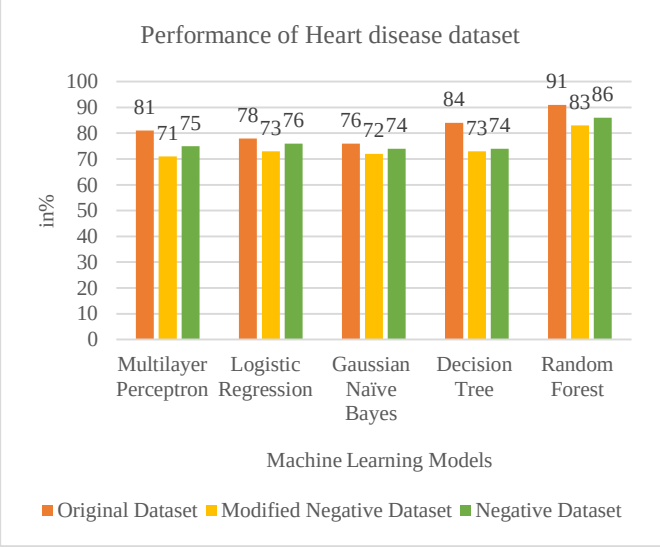

Figure 4: Bar graph of accuracy measure of heart disease data-1 using machine learning models

The following figure 4 and table 1 show the performance of the proposed models using the heart disease data-1 dataset. The proposed MLP model gets 81% accuracy on the original dataset and 71% and 75% on modified negative and negative datasets. The second proposed logistic regression model gets 78% accuracy on the original dataset and 73% and 76% accuracy on a modified negative and negative dataset. Third Gaussian naïve Bayes obtained 76% accuracy on modified negative and negative datasets and 72% and 74% accuracy. The fourth decision tree proposed model gets 84% accuracy on the original dataset, 73% and 74% accuracy on a modified negative and negative dataset, and the last proposed random model obtains 91% accuracy on the original dataset and 83% and 86% accuracy on the original modified negative and negative dataset. We can see that the proposed RF gets the highest accuracy, 91% on the original dataset, 83% highest accuracy of random forest modified negative, and 86% on the random forest on a negative dataset.

4.2 Simulation results of Milk New Dataset

This section provides the simulation results of proposed machine learning classifiers using the Milk New dataset in terms of accuracy performance measures.

Table 2: Accuracy Performance of Milk New dataset

Models	Original Dataset	Modified Negative Dataset	Negative Dataset
Multilayer Perceptron	75	72	74
Logistic Regression	82	75	72
Gaussian Naïve Bayes	98	89	91
Decision Tree	97	87	92
Random Forest	94	82	91

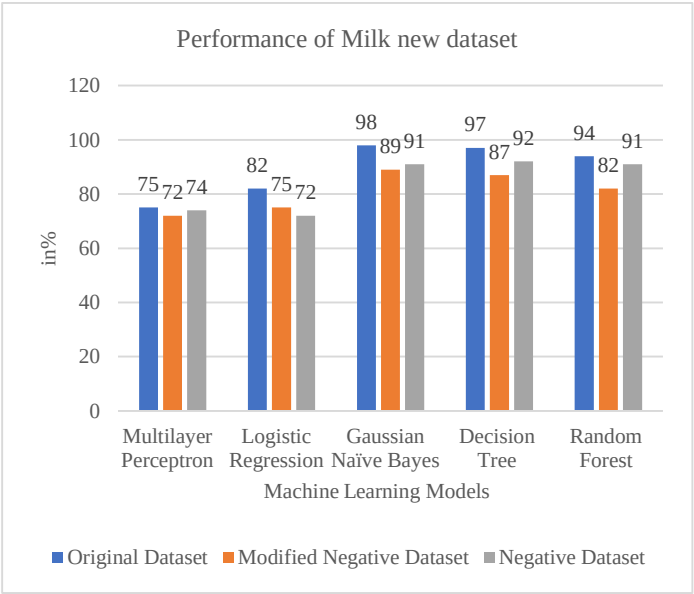

Figure 5: Bar graph of accuracy measure of Milk New dataset using machine learning models

The following figure 5 and table 2 show the performance of proposed models using the Milk New dataset. The proposed MLP model gets 75% accuracy on the original dataset, and 72% and 74% on modified negative and negative datasets. The second proposed logistic regression model gets 82% accuracy on the original dataset and 75% and 72% on the modified negative and negative datasets. Third Gaussian naïve Bayes

obtained 98% accuracy on modified negative and negative datasets and 89% and 91% accuracy. The fourth decision tree proposed model gets 97% accuracy on the original dataset, 87% and 92% accuracy on a modified negative and negative dataset, and the last proposed random model obtains 94% accuracy on the original dataset and 82% and 91% accuracy on the original modified negative and negative dataset. We can see that the proposed GNB gets the highest accuracy at 98%, 89%, and 91% on the original modified negative and negative datasets.

4.3 Simulation results of the Car Evaluation Dataset

This section provides the simulation results of proposed machine learning classifiers using the Car Evaluation dataset in terms of accuracy performance measures.

Table 3: Accuracy Performance of Car Evaluation dataset

Models	Original Dataset	Modified Negative Dataset	Negative Dataset
Multilayer Perceptron	69	67	67
Logistic Regression	62	61	61
Gaussian Naïve Bayes	71	63	63
Decision Tree	82	67	67
Random Forest	69	65	65

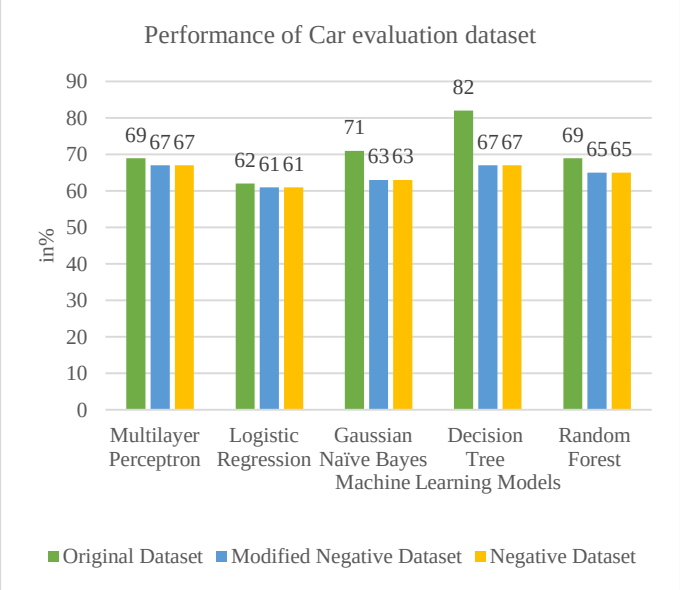

Figure 6: Bar graph of accuracy measure of Car Evaluation dataset using machine learning models

The following figure 6 and table 3 show the performance of proposed models using the Car Evaluation dataset. The proposed MLP model gets 69% accuracy on the original dataset and 67% on modified negative and negative datasets. The second proposed logistic regression model gets 62% accuracy on the original dataset and 61% on the modified negative and negative datasets. Third Gaussian naïve Bayes obtained 71% accuracy on modified negative and negative datasets and 63% accuracy. The fourth decision tree proposed model gets 82% accuracy on the original dataset, 67% accuracy on a modified negative and negative dataset, and the last proposed random model obtains 69% accuracy on the original dataset and 65% on the modified negative and negative dataset. We can see that the proposed decision tree gets the highest accuracy, 82%, on the original dataset and the highest 67% accuracy obtained by the decision tree and MLP model on the modified negative dataset.

4.4 Simulation results of Heart Disease Prediction -303 rows 14 column

This section provides the simulation results of proposed machine learning classifiers using a heart disease dataset containing 1190 rows and 12 columns.

Table 4: Accuracy Performance of Heart Disease Prediction dataset-II

Models	Original Dataset	Modified Negative Dataset	Negative Dataset
Multilayer Perceptron	76	68	71
Logistic Regression	81	67	73
Gaussian Naïve Bayes	77	72	73
Decision Tree	72	68	69
Random Forest	89	71	73

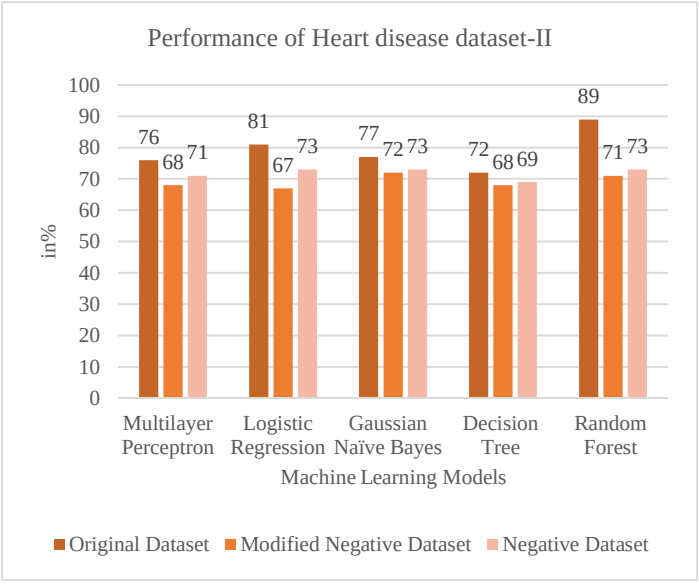

Figure 7: Bar graph of accuracy measure of Heart Disease Prediction dataset-II using machine learning models

The following figure 7 and table 4 show the performance of the proposed models using the Heart Disease Prediction dataset-II dataset. The proposed MLP model gets 76% accuracy on the original dataset, and 68 % and 71% on modified negative and negative datasets. The second proposed logistic regression model gets 81% accuracy on the original

dataset and 67% and 73% on modified negative and negative datasets. Third Gaussian naïve Bayes obtained 77% accuracy on modified negative and negative datasets and 72% and 73% accuracy. The fourth decision tree proposed model gets 72% accuracy on the original dataset, 68% and 69% accuracy on a modified negative and negative dataset, and the last proposed random model obtains 89% accuracy on the original dataset and 71% and 73% accuracy on a modified negative and negative dataset. We can see that the proposed random forest gets the highest accuracy, 89%, on the original dataset.

4.5 Simulation results of Blood fusion dataset

This section provides the simulation results of proposed machine learning classifiers using the Blood fusion dataset in terms of accuracy performance measure.

Table 5: Accuracy Performance of Blood fusion dataset

Models	Original Dataset	Modified Negative Dataset	Negative Dataset
Multilayer Perceptron	70	69	69
Logistic Regression	77	71	71
Gaussian Naïve Bayes	75	71	71
Decision Tree	72	66	66
Random Forest	73	71	71

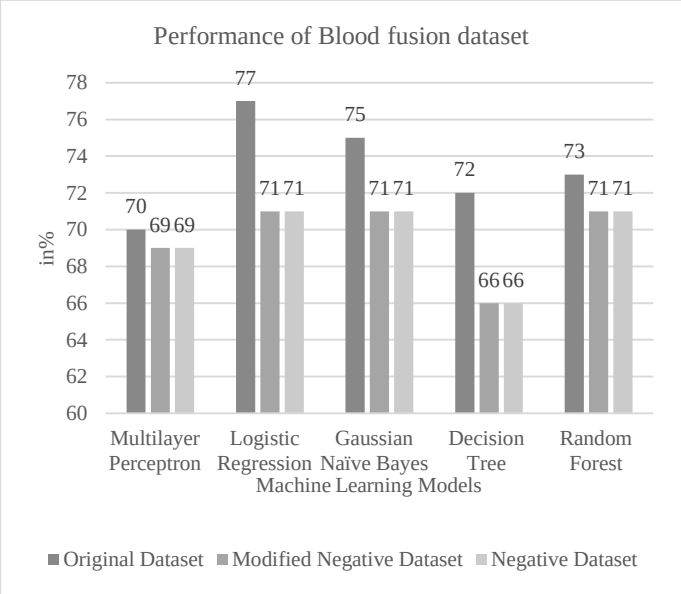

Figure 8: Bar graph of accuracy measure of Blood fusion dataset using machine learning models

The following figure 8 and table 5 show the performance of the proposed models using the blood fusion dataset. The proposed MLP model gets 60% accuracy on the original dataset and 69% on modified negative and negative datasets. The second proposed logistic regression model gets 77% accuracy on the original dataset and 71% on the modified negative and negative datasets. Third Gaussian naïve Bayes obtained 71% accuracy on modified negative and negative datasets and 75% accuracy. The fourth decision tree proposed model gets 72% accuracy on the original dataset, 66% accuracy on a modified negative and negative dataset, and the last proposed random model obtains 73% accuracy on the original dataset and 71% accuracy on a modified negative and negative dataset. We can see that the proposed logistic regression gets the highest accuracy of 73% on the original dataset.

5. CONCLUSION

In this research, we propose a novel model called NegML and introduce the negative database concept to machine learning to protect users' privacy. The results of the experiments performed on various datasets show that by modifying the settings, NegML can provide varying degrees of protection for the user's privacy. Compared to the original deep learning model, which did not include any kind of privacy protection, NegML can maintain the majority of the accuracy while also offering protection.

We will integrate the suggested technique with the stochastic gradient descent methodology in our future work. Additionally, we will seek to safeguard the weights used in distributed deep learning models and the confidential data that was initially collected. We will also try to utilize the suggested strategy for applications in the real world that include large amounts of data and classified data.

REFRENCES

1. H. Li, L. Xiong, L. Ohno-Machado, and X. Jiang, **Privacy-preserving RBF kernel support vector machine**, *Biomed Res. Int.*, 2014,doi: 10.1155/2014/827371.
2. N. Phan, X. Wu, H. Hu, and D. Dou, **Adaptive Laplace mechanism: Differential privacy preservation in deep learning**, 2017.doi: 10.1109/ICDM.2017.48.
3. D. Bhatt, **Privacy-Preserving in Machine Learning** (PPML), *analyticsvidhya*,2022. https://www.analyticsvidhya.com/blog/2022/02/privacy-preserving-in-machine-learning-ppml/
4. S. Khan, V. Saravanan, G. C. N, T. J. Lakshmi, N. Deb, and N. A. Othman, **Privacy Protection of Healthcare Data over Social Networks Using Machine Learning Algorithms**,*Comput. Intell. Neurosci.*, vol. 2022, p. 9985933, 2022, doi: 10.1155/2022/9985933.
5. A. Ullah *et al.*, **Fusion of Machine Learning and Privacy-Preserving for Secure Facial Expression Recognition**, *Secure. Commun. Networks*, 2021, doi: 10.1155/2021/6673992.

6. L. Wang *et al.*, **A User-Centered Medical Data Sharing Scheme for Privacy-Preserving Machine Learning**, *Secure. Commun. Networks*, vol. 2022, p. 3670107, 2022, doi: 10.1155/2022/3670107.
7. F. Esponda, E. S. Ackley, P. Helman, H. Jia, and S. Forrest, **Protecting data privacy through hard-to-reverse negative databases**, 2007. doi: 10.1007/s10207-007-0030-1.
8. D. Zhao, P. Zhang, J. Xiang, and J. Tian, **NegDL: Privacy-Preserving Deep Learning Based on Negative Database,** pp. 1–14, 2021, [Online]. Available: http://arxiv.org/abs/2103.05854
9. M. Zhang, W. Song, and J. Zhang, **A Secure Clinical Diagnosis With Privacy-Preserving Multiclass Support Vector Machine in Clouds**, *IEEE Syst. J.*, vol. 16, no. 1, pp. 67–78, 2022, doi: 10.1109/JSYST.2020.3027758.
10. Y. Chen, Q. Mao, B. Wang, P. Duan, B. Zhang, and Z. Hong, **Privacy-Preserving Multi-Class Support Vector Machine Model on Medical Diagnosis**, *IEEE J. Biomed. Heal. Informatics*, 2022, doi: 10.1109/JBHI.2022.3157592.
11. P. Panzade and D. Takabi, **Towards Faster Functional Encryption for Privacy-preserving Machine Learning**, in *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 2021, pp. 21–30. doi: 10.1109/TPSISA52974.2021.00003.
12. H. Fang and Q. Qian, **Privacy-preserving machine learning with homomorphic encryption and federated learning**, *Futur. Internet*, 2021, doi: 10.3390/fi13040094.
13. S. Park, J. Byun, J. Lee, J. H. Cheon, and J. Lee, **HE-friendly algorithm for privacy-preserving SVM training**, *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.2981818.
14. R. Talbi, **Towards practical privacy-preserving collaborative machine learning at a scale,** 2020. doi: 10.1109/DSN-S50200.2020.00037.
15. W. Romsaiyud, H. Schnoor, and W. Hasselbring, **Improving k-Nearest Neighbor Pattern Recognition Models for Privacy-Preserving Data Analysis,** 2019. doi: 10.1109/BigData47090.2019.9006281.
16. Q. Jia, L. Guo, Y. Fang, and G. Wang, **Efficient Privacy-Preserving Machine Learning in Hierarchical Distributed System**, *IEEE Trans. Netw. Sci. Eng.*, 2019, doi: 10.1109/TNSE.2018.2859420.
17. M. Kitayama and H. Kiya, **HOG feature extraction from encrypted images for privacy-preserving machine learning**, 2019. doi: 10.1109/ICCE-Asia46551.2019.8942217.
18. F. O. Catak, A. F. Mustacoglu, and A. E. Topcu,**Privacy preserving extreme learning machine classification model for distributed systems**,2016. doi: 10.1109/siu.2016.7495740.
19. F. Murtagh, **Multilayer perceptrons for classification and regression,** *Neurocomputing*, 1991, doi: 10.1016/0925-2312(91)90023-5.
20. A. Ng, **Logistic Regression Classification,** *14 Oct.*, 2013.
21. M. Jena and S. Dehuri, **Decision tree for classification and regression: A state-of-the art review,** *Informatica (Slovenia)*. 2020. doi: 10.31449/INF.V44I4.3023.
22. S. S A, **Comparative Study of Naive Bayes, Gaussian Naive Bayes Classifier and Decision Tree Algorithms for Prediction of Heart Diseases**, *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 9, no. 3, pp. 475–486, 2021, doi: 10.22214/ijraset.2021.33228.
23. G. Stamatescu and C. Chitu, **Privacy-Preserving Sensing and Two-Stage Building Occupancy Prediction Using Random Forest Learning**, *J. Sensors*, 2021, doi: 10.1155/2021/8000595.
24. A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu, **Improved Random Forest for Classification**, *IEEE Trans. Image Process.*, 2018, doi: 10.1109/TIP.2018.2834830.