International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 05, May-2023

Security in Machine Learning - Impact of Label Poisoning Attack

Beauty Kumari
Department of Computer Science and Engg
Birla Institute of Technology,
MESRA Ranchi, India

Dr. Vandana Bhattacharjee
Department of Computer Science and Engg
Birla Institute of Technology,
MESRA Ranchi, India

Abstract— **Machine learning is a triumph technique in computer science and is extensively used in many important fields to support intelligent decision-making, image processing, pattern recognition, natural language processing, etc. The effectiveness of machine learning in security-sensitive applications hinges on a careful examination of their resistance to hostile input. Adversarial machine learning is the study of the attacks on learning algorithms, and the defenses against such attacks. One realistic, well-intentioned attack scenario is an adversary meticulously changing attack samples to attempt to elude a deployed system during testing. This paper provides an overview of this developing field, a discussion of adversary modelling approaches, as well as a look into poisoning attempts on the data and their analysis. We explore the threat models for Machine Learning systems and, vulnerabilities in machine learning algorithms and will also see the attack on various machine learning paradigms. Despite the fact that machine learning algorithms have been successfully applied in many settings, the algorithms and associated training data are vulnerable to a number of security concerns, leading to a considerable performance hit. So, it is more important to put our attention on security risks and respective defensive techniques of machine algorithms for our future and ever-changing technology. The report ends by referencing some of the major current research themes that open doors for future research.**

Keywords—Adversarial machine learning, Security threat, Poisoning attack, Label poisoning, Performance evaluation

I.INTRODUCTION

In the present scenario machine learning (ML) is one of the most desired research fields because of its application and its effectiveness which has been validated in various fields like pattern recognition, image processing, clustering, network intrusion detection, autonomous driving, etc. These applications are different from traditional machine learning settings in which the underlying data distribution is believed to be stationary since they are inherently hostile. The advancement of big data has worked a like catalyst along with complex mathematical calculations and its implementation in machine learning has enabled corresponding algorithms to disclose more precise patterns and make more accurate predictions and decisions than ever before. As we know it's a well-known fact that everything comes with its own diddle and ever-increasing dependence on machine learning by humans has made it prone to attacks in various ways. AML helps in many important ways and one we can mention is that we can plan for the course of actions to model and counter adversarial behaviour.

This paper is about the developing area of adversarial machine learning (AML) where we would see the design and development of machine learning, discusses various ML attacking technique, detailed discussion on poison attack, and going to see how poisoning data affects the accuracy of a machine learning model with label flipping technique.

II. RELATED WORK

Authors in [6] propose the idea that a particular idea of separability in the RKHS caused by the infinite-width network is important. It is also demonstrated that training (finite-width) networks with stochastic gradient descent are robust against data poisoning attempts. In another study [7] the authors suggest a method for creating linear classifiers that are indisputably resistant to a powerful label-flipping version in which each test example is individually targeted. In other words, the classifier makes a prediction for each test point and certifies that it would have made the same forecast even if certain training labels had been modified arbitrarily. In order to guarantee a high probability test-time robustness to adversarial manipulation of the input to a classifier, our method makes use of randomized smoothing. In another study[1] authors offer a weighted SVM that protects against poisoning attacks (label-flip) using K-LIDxas a differentiating characteristic that completely disregards the impact of suspect data samples on the SVM decision boundary. According to how likely it is that a sample's K-LID value will come from the protected K-LID distribution as opposed to the attacked K-LID distribution, each sample is given a weight. Using standard performance data sets as test subjects, experiments reveal that the suggested defence significantly lowers classification error rates (to an average of 10%). In another study [2] authors develop a better defensive plan that places a strong emphasis on using KPCA and K-mean clustering. The outcomes of our defence technique against data-poisoning attacks in a federated-learning system are demonstrated in this work to be improved when coupled with enhanced dimensionality-reduction algorithms. In another study[3], The authors demonstrate the bagging system's intrinsic validated resilience against data poisoning threats. Also, demonstrate that bagging with an xarbitrary basexlearning algorithmxprovably predicts the same label for a testing case when the quantity of altered, subtracted, or added training examples is constrained byxthresholdxand illustrates the tightness of our calculated threshold in the absence of any basic learning algorithm assumptions.

III. OVERVIEW OF SECURITY THREATS IN ML

A) DEFENSELESSNESS IN MACHINE LEARNING ALGORITHM:

Machine learning models are generated from data that contain vulnerabilities. The incorrect presumptions used in the ML model's construction and training represent a significant source of vulnerability. For machine learning (ML) models to be accurate and trustworthy, privacy protection is an implicit presumption made by data scientists. However, this presumption is incorrect and leads to significant privacy violations. As a result of the adversary's ability to produce adversarial cases and further impair the model's performance, this assumption raises the overall misclassification Data collection can occur occasionally in hostile settings and without oversight, such as when information is gathered from servers serving as honeypots. Since the adversary has direct access to the training data, it is possible for attackers to meticulously create hostile samples to be collected as data, which may worsen the model. An adversary could attempt to obtain the correlation between various data points and features in order to get the information from various data distributions, which would impair the performance of the model. The fact that ML models perform well on training and test data, which are frequently derived from the same underlying distribution, is one of their main weaknesses. The model will respond differently if the input data come from a different distribution.

Real-life examples of malfunctioning of the ML model: -

Self-driving cars: What could possibly go wrong?

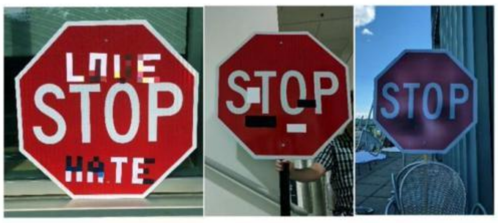


Fig 1: Adversarial example

A team of eight researchers has discovered that by altering street signs, an adversary could confuse self-driving cars and cause their machine learning systems to misclassify signs and take wrong decisions, potentially putting the lives of passengers in danger. Attackers use this fundamental flaw to create adversarial examples that incorrectly train the model and worsen its performance.

B) ADVERSARIAL MODEL

The four parameters of objective, knowledge, capacity and attacking technique should be included in a well-defined adversarial model.

1) ATTACKER'S OBJECTIVE:

The three views listed below can be used to classify an attacker's hostile objectives:

- Security Violation:

  With the traditional CIA model (confidentiality, integrity, availability), an attacker may seek to compromise the functionality of an ML system (integrity and availability), or to extract confidential or private information about the ML system.

- Attack Specificity:
  Attacks can be launched randomly against any ML system or targeted against a particular ML architecture or methodology.

- Error Specificity:
  An attacker may attempt to trick the system into incorrectly classifying an input sample into a particular class (error-specific attacks) or into any class other than the correct class in the context of ML classifier systems (error-generic attacks).

2) ATTACKER'S KNOWLEDGE

- Perfect Knowledge:
  Attacks with perfect knowledge, also known as white-box attacks, occur when a target ML system is attacked. This includes the training dataset, ML architecture, learning algorithms, trained model parameters, etc. The worst-case scenario for an attack can be seen in this environment.

- Limited Knowledge:
  Attacks with limited knowledge , also known as Gray-box attacks,, are ones in which the attacker only possesses a subset of the information about the targeted ML system. In most cases, the attacker is expected tobe aware of the feature set, the model architecture, and the learning methods but not the training data and trained model parameters.

- zero knowledge:
  Attacks with zero knowledge , also known as black-box attacks, presuppose that the attacker is unaware of all "precise" details pertaining to the targeted ML system.

3) ATTACKER'S CAPACITY

It refers to the degree to which an attacker can access and alter training data or input samples, or view the matching output of a trained model. The degree of the attacker's access and data manipulation comprises reading, injecting, changing, or logically corrupting training data or input samples, in the order of the attacker's capabilities from poor to severe.

4) ATTACKING TECHNIQUES

- FGSM (The Fast Gradient Sign Method) algorithm:
  An effective adversarial sample generation technique called FGSM produces samples by adding noise to the source image in the gradient directions.

- C&W algorithm:
  Developed by Carlini and Wagner, the C&W attack is a potent adversarial sample generation approach that performs better in terms of computation speed. On both distilled and undistilled DNN models, it has reportedly produced outstanding results.

- DeepFool algorithm:
  Based on an iterative linearization of the classifier, the DeepFool algorithm calculates the distance between the adversarial samples' original input and the decision boundary. The DeepFool algorithm offers a quick and precise method for determining how robust a classifier is and for properly fine-tuning it to perform better.

- JSMA (Jacobian-based Saliency Map) algorithm:
  To effectively produce adversarial samples based on computing forward derivatives, the author [9]developed the Jacobian-based Saliency Map (JSMA) technique. To determine the input features of a sample x that significantly altered the outcome classification, JSMA computes the Jacobian matrix of x.

C) ATTACKS ON DIFFERENT MACHINE LEARNING CONCEPTS

Here, many assaults on the supervised and unsupervised learning paradigms of machine learning are examined.

1) SUPERVISED MACHINE LEARNING:

Barreno et al. [5] proposed a taxonomy that distinguishes the features of vulnerabilities and attacks along three different dimensions to illustrate a general model of security for supervised machine learning. Their debate is framed in the context of a supervised machine learning system that is intended to identify and protect against possible attackers. These are the three dimensions:

• Influence: Explorative vs. Causative
This shows whether the categorization of new data itself is compromised (explorative) in real time, or whether the classification of training data itself is compromised (causative), resulting in the production of a flawed prediction or classification model.

• Security Violation: Integrity vs Availability
This determines whether the exploitation focuses on compromise through the production of false negatives (integrity) or through an abundance of false positives (availability).

• Specificity: Targeted vs. Indiscriminate
Whether a specific instance is the focus (targeted) or a larger class is relevant in this aspect (indiscriminate).

2) UNSUPERVISED MACHINE LEARNING:

Data in unsupervised learning only contains the input features and has no labels attached to them. These are utilized to cluster or group data based on similar input features or to discover a new data representation. Generative models, autoencoders, and clustering algorithms are three types of attacks against unsupervised ML models.

D) DIFFERENT TYPES OF ATTACKS ON THE MACHINE LEARNING MODEL

1) EVASION ATTACKS: -

Evasion attacks evade the ML model by passing an adversarial example so that the model misclassifies. It is a test-time attack that does not require accessing and manipulating the training data.

It refers to creating an input that, although appearing natural to a human, is misclassified by ML models. A typical example is altering a few pixels in a photo before uploading, causing the image recognition system to misclassify the outcome. In fact, this adversarial scenario can deceive people.

- See the example below.

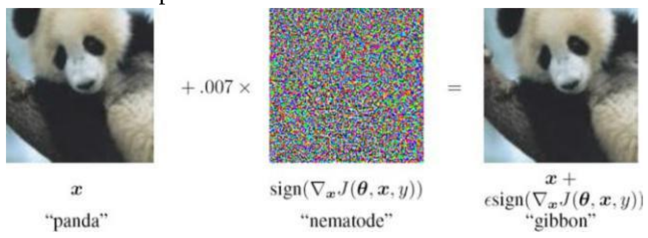

+ .007 ×
=
x
"panda"
$\mathrm{sign}(\nabla_x J(\theta, x, y))$
"nematode"
$x + \epsilon\mathrm{sign}(\nabla_x J(\theta, x, y))$
"gibbon"

Fig2: Adversarial example generated by adding noise

2) EQUATION SOLVING ATTACK:

The equation solving attack is applicable to cloud providers who provide ML as a service via APIs and for models such as multi layer perceptron, binary logistic regression, and multi-class logistic regression where they are represented as equations in known and unknown variables. The goal is to use the data to find the unknown variables, which are usually the parameters used to train the models. These attacks are expected to reveal information about the model and its architecture to the attacker.

3) PATH FINDING ATTACK: -

Path-finding attacks are used to traverse binary trees, multi-n-ary trees, and regression trees. In these attacks, the value of each input feature is varied till the conditions at each node are satisfied, while the input traverses the tree. The tree is traversed until a leaf is reached or an internal node with a split over a missing feature is found. The value of the leaf node is the output which reveals the path followed.

4)BLACK BOX ATTACKSXUSING TRANSFERABIL ITY PRO PERTY: -

In black box attacks, the adversary has no access to the data and the model. The attacker can only access the oracle that returns output for the input chosen by the attacker. The lack of knowledge of the model can be alleviated using the property of

transferability which states that samples crafted to mislead model A are likely to mislead model.

5) MEMBER INFERENCE ATTACK: -

In member inference attacks, the attacker finds if a query passed to the prediction API is part of the training set and if so, leaks the training data information.

6) POISONING ATTACKS: -

The model training phase of machine learning is prone to attacks.

In order to change the statistical properties of the training/test dataset, a type of attack known as a "poisoning attack” tries to introduce a small percentage of "poisoned" samples. This compromises the ML model and causes it to suffer from an increased rate of misclassified samples at the prediction stage.

The integrity and availability of an ML system are both targeted by a poisoning attack, which is seen as a causative attack.

There are various poisoning attack algorithms, some of which are discussed below.

- Label-flipping Attack:

By inverting the labels, this form of attack adds label noise to the training set. Label-flipping Attack is a type of causative attack and, can compromise integrity or availability of an ML system. The generation of adversarial samples is accomplished using a variety of flipping algorithms, such as adversarial label flipping (ALF)., nearest-prior label flipping (NPLF), farthest-prior label flipping (FPLF), and random label flipping (RLF).

- Clean label Attack:

Targeted clean-label data poisoning is a sort of adversarial tactic on machine learning systems in which an adversary inserts a few correctly labeled, little altered samples into the training data, causing a model to incorrectly classify a particular test sample during inference. Neural nets are the target of this kind of attack. They intend to wrongly label one test instance. For instance, they can trick a face recognition system into believing one person is someone else, or they can trick a spam filter into allowing or blocking a certain email.

- Gradient Descent Attack:

A causative, availability-compromised assault known as a "gradient descent-based poisoning attack" introduces hostile samples into the training dataset in order to have the greatest possible effect on the performance of a machine learning system. In order to enhance the learner's objective function, the label of a harmless training set is first flipped, and then it is moved using the gradient descent function. This is a prevalent label-flipping attack technique.

IV. METHODOLOGY

In this thesis we are using model diabetes prediction and with the help of poisoning attack (Label flipping) we would see how prone our models can be to this attack. So, with the change in amount of poisoning we would see the gradual increase of error in the prediction model. We shall see the fall in the accuracy with the gradual change in the database by poisoning the diabetes data (Pima Indians Diabetes Database) using different classifier.

A) DATASET USED:

Pima Indians Diabetes Database

The National Institute of Diabetes and Digestive and Kidney Diseases is the source of this dataset. Based on specific diagnostic metrics in the dataset, the goal of the dataset is to diagnostically predict whether a patient has diabetes.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 05, May-2023

The datasets consist of one target variable, the Outcome, and several medical predictor variables. The patient's BMI, insulin level, age, number of previous pregnancies, and other factors are predictor variables.
Number of Observation Units: 768
Variable Number: 9

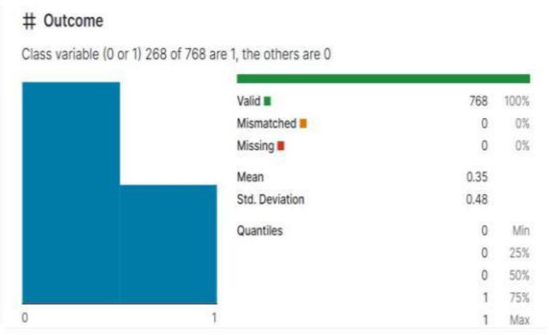
This dataset consists of multiple independent variables and one dependent variable (Outcome). Independent variables include:
1.Pregnancies: Number of times pregnant.

2.Glucose: Plasma Glucose Concentration a 2-hour in an Oral Glucose Tolerance Test(mg/dl).

3.Blood pressure: Diastolic Blood Pressure (mm/Hg).
4.Skin Thickness: Triceps Skin Fold Thickness (mm)
5.Insulin: 2-Hour Serum Insulin (mu U/ml)
6.BMI: Body Mass Index (weight in kg/ (height in m) ^2)

7.Diabetes Pedigree Function: It provides information about diabetes history in relatives and the genetic relationship of those relatives with patients. Higher Pedigree Function means a patient is more likely to have diabetes.
8.Age: Age of an individual (years)

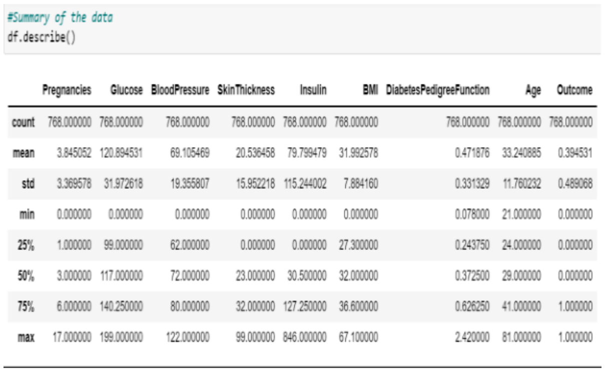
9.Outcome: Target Variable (0 or 1) where ‘0’ denotes the patient is not diabetic and ‘1’ denotes the patient is diabetic.


Outcome
Class variable (0 or 1) 268 of 768 are 1, the others are 0

Valid 🟩	768	100%
Mismatched 🟧	0	0%
Missing 🟥	0	0%
Mean	0.35	
Std. Deviation	0.48	
Quantiles	0	Min
	0	25%
	0	50%
	1	75%
	1	Max

Fig3: Target variable(outcome) description

- As we can see in the above graph total 768 row is used for observation.
- Here in totality 268 out of total is positive and others are negative.
- Other data description is well defined in the above graph.
- Data Description:


```
#Summary of the data
df.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.394531
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.489068
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Fig4: Snapshot of dataset description

B) CLASSIFIER USED:

1. Logistic regression
2. Decision tree
3. Support vector machine
4. Random forest

1) LOGISTIC REGRESSION

Regression Machine learning is a classification method that falls under the Supervised category (a sort of machine learning in which machines are trained using labeled data, and based on that learned data, the output is predicted) of machine learning algorithms.

The primary function of logistic regression in machine learning is to forecast the results of a categorical dependent variable from a set of independent variables. In plain English, a categorical dependent variable is one whose data is coded in the form of either 1 (stands for success/yes) or 0 (stands for failure/no) and is duality or binary in nature.

Mathematically, the same idea can be described as a logistic regression model that forecasts P(Y=1) as a function of X. The result is a discrete or direct value that is categorical in nature. These could be 0/1, True/False, or Yes/No. However, as I previously mentioned, the Logistic Regression Algorithm is based on Statistics, therefore rather than returning a result of 0 or 1, it returns a probabilistic result that falls between 0 and 1.

Although one of the most straightforward machine learning algorithms, logistic regression has a wide range of uses in classification issues, including spam identification, diabetes prediction, and even cancer detection.

Logistic Function (Sigmoid Function):

The projected values are converted to probabilities using a mathematical tool called the sigmoid function.

It transforms any real value between 0 and 1 into another value. The logistic regression's result must fall within the range of 0 and 1, and because it cannot go beyond this value, it has the shape of an "S" curve. The sigmoid function or logistic function is another name for the S-form curve.We apply the threshold value idea in logistic regression, which establishes the likelihood of either 0 or 1. Examples include values that incline to 1 over the threshold value and to 0 below it.It is also known as the Logistic Regression Machine Learning Activation function.

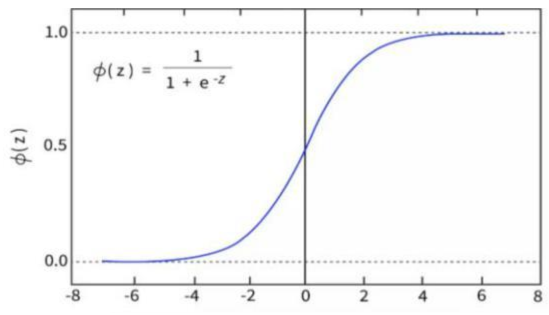


Fig5: Graph of logistic regression classifier

a)LOGISTIC REGRESSION ASSUMPTION:

- There should be minimal or no multicollinearity among the independent variables.
- The dependent variable should be categorical in nature that is have a **finite number of categories or distinct groups**. Example: Gender-Male and Female
- The sample size in our dataset should be large to give the best possible results, that is probabilities between 0 and 1 for our Logistic Regression Model.
- Categorical dependent variables must be meaningful.
- For a binary classifier, the target variables must be binar always.

2) DECISION TREE:

Decision Tree is a supervised learning method that can be applied to classification and regression issues; however, it is most frequently used to address classification issues. It is a tree structured classifier, where internal nodes stand in for the dataset's features, branches for the rules of classification, and each leaf node for the result.Two nodes—the Decision Node and the Leaf Node—make up a decision tree. A choice is made using a Decision node, which has several branches, whereas a Leaf node is the result of that decision and does not have any additional branches.The features of the given dataset are used to execute the tests or make the judgments.The decision tree's general structure is shown in the diagram below:

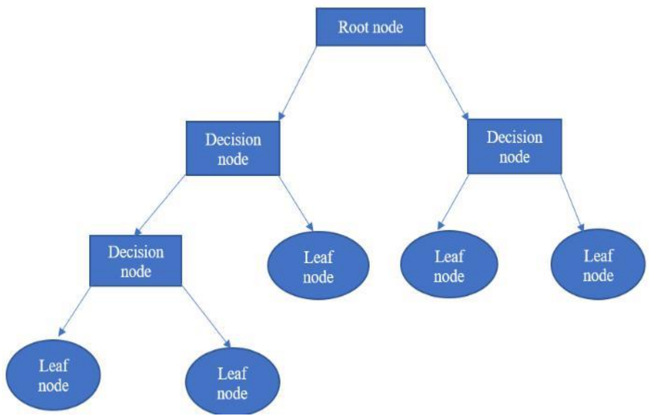


Fig6: Decision tree structure

It is a graphical depiction for obtaining all feasible answers to a choice or problem based on predetermined conditions.

It is known as a decision tree because, like a tree, it begins with the root node and grows on subsequent branches to form a structure resembling a tree.

The CART algorithm, which stands for Classification and Regression Tree algorithm, is used to construct a tree.

A decision tree only poses a question and divides the tree into subtrees according to the response (Yes/No).

TERMS USED IN DECISION TREES:

Root node: The decision tree's starting point is known as the root node. It is a representation of the whole dataset, which is then split into two or more homogeneous sets.

- Leaf Node: Leaf nodes are the last output nodes, and the tree cannot be further divided after a leaf node is obtained.
- Splitting: The process of splitting entails dividing the decision node/root node into sub-nodes in accordance with the specified conditions.
- Branches or subtrees: created by splitting a tree.
- Pruning: Pruning is the removal of undesirable branches from a tree.
- Parent/Child node: The parent node in a tree is the main node, while the child nodes are the other nodes.

DECISION TREE ALGORITHM:

Step 1: According to S, start the tree at the root node, which contains the entire dataset.

Step 2: Utilize the Attribute Selection Measure to identify the best attribute in the dataset (ASM).

Step 3: Separate the S into subsets that include potential values for the best qualities.

Step 4: Create the decision tree node that holds the best attribute. Step 5: Recursively generate new decision trees using the subsets of the dataset produced in step 3. Continue doing this until you reach a point when you can no longer categorise the nodes and you refer to the last node as a leaf node.

a)ATTRIBUTE SELECTION MEASURES:

The biggest challenge while designing a decision tree is deciding which attribute is ideal for the root node and sub-nodes. An approach known as attribute selection measure, or ASM, has been developed to address such issues. We can choose the ideal attribute for the tree's nodes with ease using this measurement. There are two widely used ASM approaches, namely:

- Information Gain
- Gini Index

Information Gain:

After segmenting a dataset based on an attribute, information gain is the measurement of changes in entropy.

It determines how much knowledge a feature gives us about a class. Information Gain= Entropy(S)- [(Weighted Av) *Entropy (each feature)

Entropy is a statistic used to assess the impurity of a particular characteristic. It describes the randomness of data. Entropy

is determined by:
Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)
Here,

- S= Total number of samples
- P(yes)= probability of yes
- P(no)= probability of no

Gini Index:

The CART (Classification and Regression Tree) technique uses the Gini index as a measure of impurity or purity while building decision trees. An attribute with a low Gini index is preferable to one with a high Gini index. It solely generates binary splits, which are generated by the CART algorithm using the Gini index.

The formula below can be used to compute the Gini index:

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2$$

where,

'pi' = probability of an object being classified to a particular class.

3) SUPPORT VECTOR MACHINE:
SVM was developed at the AT & T Bell laboratories by Vapnik and his co workers in 1995. One of the most widely used methods for Supervised Learning, Support Vector Machine (SVM), is used to solve Classification and Regression issues. In machine learning, it is generally employed to solve classification issues. The SVM algorithm's objective is to establish the best line or decision boundary that can divide n-dimensional space into classes so that we may quickly classify fresh data points in the future. A hyperplane is the name of this optimal decision boundary. Consider the diagram below, where a decision boundary or hyperplane is used to categorise two distinct categories:

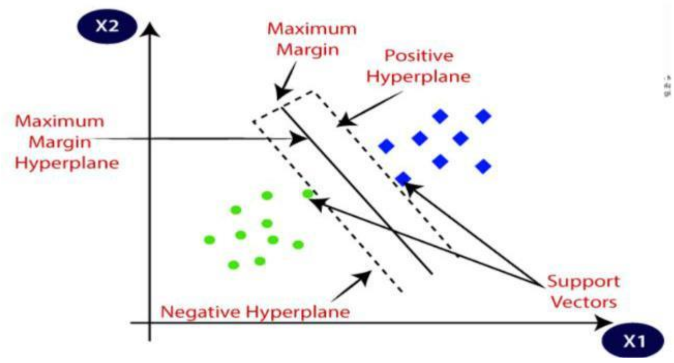


Fig7: SVM classifier

SVM comes in two varieties:

Linear SVM: Linear SVM is used for data that can be divided into two classes using a single straight line. This type of data is called linearly separable data, and the classifier employed is known as a Linear SVM classifier.

Non-linear SVM: Non-Linear SVM is used for non-linearly separated data. If a dataset cannot be classified using a

straight line, it is considered non-linear data, and the classifier employed is referred to as a Non linear SVM classifier.

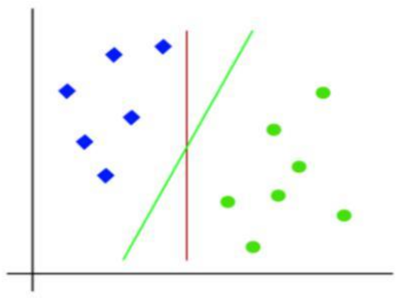


Fig8: linear SVM

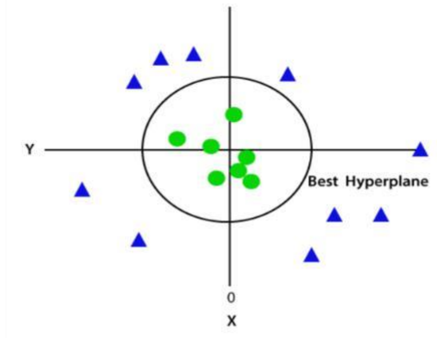


Fig9: Non-linear SVM

SVM Kernels:

The kernels are a set of mathematical operations used by SVM algorithms. A kernel's job is to take data as input and change it into the required form.

Different kernel functions are used by various SVM algorithms. These functions come in a variety of types, including sigmoid, polynomial, radial basis function (RBF), linear, and nonlinear.

RBF is the kernel function type that is most favoured. as a result of its localization and finite reaction along the entire x-axis.

The scalar product between two points in an incredibly appropriate feature space is returned by the kernel functions. Thus, even in the situation of very high-dimensional spaces, by

defining a notion of similarity, with little computational expense.

a)Popular SVM Kernel Functions:

1) *Linear Kernel:*

It is the most fundamental kind of kernel and is often of a one-dimensional kind. When there are many features, it turns out to be the best feature. For text classification issues, the linear kernel is typically favoured because most of these classification issues can be divided linearly.

The speed of linear kernel functions is superior to other functions.
Formula:

$$F(x, xj) = sum(x.xj)$$

Here, **x, xj** represents the data you're trying to classify.

activation function for neurons, is comparable to this kernel function.

$$F(x, xj) = tanh(\alpha xay + c)$$

2) *Polynomial Kernel:*

It is a more extensive representation of the linear kernel. Due to its
lower accuracy and efficiency compared to other kernel functions, it
is not as popular.

Formula:

3) *Gaussian Radial Basis Function (RBF):*

It is a popular and often used kernel function in svm. It is frequently selected for non-linear data. When there is no prior data knowledge, it aids in proper separation.

SVM Applications:

- Handwriting recognition,
- Intrusion detection,
- Face detection,
- Email classification,
- Gene classification.

4) RANDOM FOREST:

The supervised learning method includes the well-known machine learning algorithm Random Forest. It can be applied to ML Classification and Regression issues. Its foundation is the idea of ensemble learning, which is the process of mixing various classifiers to solve a challenging problem and enhance the performance of the model.

Random Forest is a classifier that, as the name implies, "contains a number of decision trees on various subsets of the provided dataset and takes the average to enhance the predictive accuracy of that dataset." Instead of depending on a single decision tree, the random forest uses forecasts from all the trees to anticipate the outcome based on which predictions received the most votes.

Higher accuracy is obtained and overfitting is avoided because to the larger number of trees in the forest.

F(x, xj) = (x.xj+1)^d

Here '.' shows the **dot product** of both the values, and **d** denotes the degree.

F(x, xj) representing the **decision boundary** to separate the given classes.

4) *Sigmoid Kernel:*

For neural networks, it is mostly favored. A two-layer

F(x, xj) = exp(-gamma * ||x - xj||^2)

The value of gamma varies from **0 to 1**. You have to manually provide the value of gamma in the code. The most preferred value for **gamma is 0.1**.

perceptron model of the neural network, which serves as an

The Random Forest algorithm's operation is shown in the diagram below:

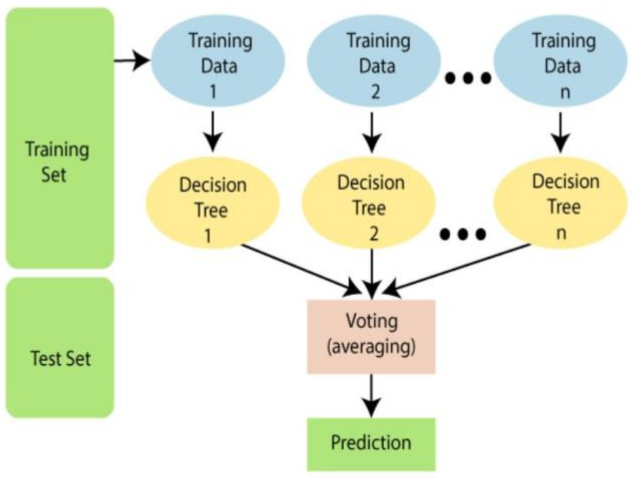


Fig10: Random Forest flow

a)Assumptions for Random Forest:

For the classifier to predict accurate results rather than an assumed outcome, there should be some real values in the feature variable of the dataset.

There must be extremely little correlation between the forecasts from each tree.

b)Random Forest algorithm:

Step 1: Pick K data points at random from the training set.

Step 2: Construct the decision trees linked to the chosen data points (Subsets).

Step 3: Select N for the size of the decision trees you wish to construct.

Repeat steps 1 and 2 in step 4.

Step 5: Assign new data points to the category that receives most votes by looking up each decision tree's predictions for the new data points.

c)LIBRARY USED:

We are implementing the whole experiment with the help of python programming language.

Given below are the pre-written libraries of python which came handy while doing this project.

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.

Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays.

Matplotlib is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in plotting area, decorates the plot with labels, etc.

Scikit Learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPyxand Matplotlib.

V. EXPERIMENTS

The experiments are performed to show the poisoning affect on the machine learning model. In the experiment we would see the gradual poisoning in the data from 2 percent to 20 percent and change in the accuracy percentage. All the necessary details are tabulated below:

A)DATA POISON PERCENTAGE:

The given below table shows data poisned percentage which is being used in the experiment:

Poison data(%)	**Label(outcome) flipped with row range and count.**
0	0
2	650-665(16)
4	500-530(31)
6	300-346(47)
8	100-161(62)
10	100-161,200-214(77)
12	100-161,200-214,50-65(93)
14	100-161,200-214,50-65,400-15(108)
16	100-161,200-214,50-65,400-30(123)
18	100-161,200-214,50-65,400-30,753-69(139)
20	100-161,200-214,50-65,400-30,753-69,700-15(155)

Table1: Details of data poison percentage

B) EXPERIMENTAL RESULTS:

Model performance (in %) using different classifiers with poisoning:

The data taken for experiment is being poisoned with different ratio and the accuracy is being seen gradually decreasing as observed in the given below table.

Poison data	**Logistic regression**	**Random forest**	**Decision tree**	SVM
0%	77	81	68	78
2%	76	80	68	78
4%	75	75	68	75
6%	72	74	73	72
8%	70	70	67	70
10%	65	66	59	66
12%	65	66	58	66
14%	64	66	52	64
16%	61	66	58	61
18%	64	64	57	61
20%	62	64	53	59

Table2: Details of model performance (in %) using different classifiers with poisoning

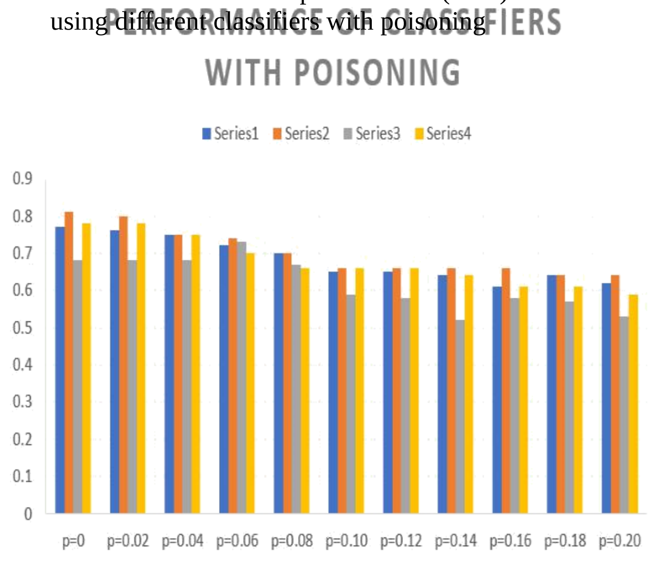


Fig15: Performance of different classifiers

Here,

Series1: Logistic regression

Series2:Random forest

Series3: Decision tree

Series4 : SVM

CONCLUSION

In this experiment we poisoned the data with poisoning ratio being minimal with the various classifiers and took the observation. In Logistic Regression accuracy is being decreased from the point we poisoned the data , but for some Instances the accuracy increased too. In Decision Tree,Its accuracy is also noted like Logistic Regression where we saw accuracy increase in certain case. Unlike Logistic Regressin and Decision Tree, SVM and Random Forest shows the strict desrease in the accuracy whenever the data poisoning percentage is increased . As we work on the small percentage of poisoning, the minor changes could also be seen with the accuracy as we noticed the increase in the accuracy foe certain case which could not be seen with large poisoning percentage. So we can conclude that data poisining attack affects the Machine Learning Model by label flipping the data and impacting the learning of the model and hence changing or degrading its accuracy.

FUTURE SCOPE

This experiment has the potential to go further for another poisoning attack i.e. clean label attack, gradient descent attack etc and it would be interesting to see the observation.

REFERENCES

[1] Defending Support Vector Machines Against Data Poisoning Attacks | IEEE Journals & Magazine | IEEE Xplore
[2] Detection and Mitigation of Label-Flipping Attacks in Federated Learning Systems with KPCA and K-Means | IEEE Conference Publication | IEEE Xplore
[3] Intrinsic Certified Robustness of Bagging against Data Poisoning Attacks | Proceedings of the AAAI Conference on Artificial Intelligence
[4] A Survey of Adversarial Machine Learning in Cyber Warfare | Defence Science Journal (drdo.gov.in)
[5] M. Barreno, B. Nelson, A. D. Joseph, J. D. Tygar, The security of machine learning, Machine Learning 81 (2) (2010) 121–148.
[6] Robust Learning for Data Poisoning Attacks Yunjuan Wang 1 Poorya Mianjy 1 Raman Arora 1
[7] Certified Robustness to Label-Flipping Attacks via Randomized Smoothing *Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, Zico Kolter Proceedings of the 37th International Conference on Machine Learning*, PMLR 119:8230-8241, 2020.
[8] Towards a Robust and Trustworthy Machine Learning System Development: An Engineering Perspective Pulei Xiong 1 , Scott Buffett 2 , Shahrear Iqbal 2 , Philippe Lamontagne 2 , Mohammad Mamun 2 , Heather Molyneaux2a aCybersecurity, National Research Council Canada, 1200 Montreal Rd, Ottawa, K1A 0R6, ON, CA
[9] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: 2016 IEEE European symposium on security and privacy (EuroS&P), IEEE, 2016, pp. 372–387.
[10] H. Xiao, H. Xiao, C. Eckert, Adversarial label flips attack on support vector machines, Frontiers in Artificial Intelligence and Applications 242 (2012) 870–875. doi:10.3233/978-1-61499-098-7-870

[11] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, T. Goldstein, Poison frogs! targeted clean-label poisoning attacks on neural networks, arXiv preprint arXiv:1804.00792 (2018).
[12] M. Fang, N. Z. Gong, J. Liu, Influence function based data poisoning attacks to top-n recommender systems, in: Proceedings of The Web Conference 2020, 2020, pp. 3019–3025.
[13] M. Fang, G. Yang, N. Z. Gong, J. Liu, Poisoning attacks to graph-based recommender systems, in: Proceedings of the 34th Annual Computer Security Applications Conference, 2018, pp. 381–392.
[14] Huang, J. Mu, N. Z. Gong, Q. Li, B. Liu, M. Xu, Data poisoning attacks to deep learning based recommender systems, arXiv preprint arXiv:2101.02644 (2021)
[15] https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database
[16] Fig2:(https://tse4.mm.bing.net/th?id=OIP.ZFgkPQ_15TW8ubOde8_dwQ HaC2&pid=Api&P=0)
[17] Fig5:(https://scaler.com/topics/images/logistic-function-in-machine-learning.webp)
[18] Fig 7: (https://static.javatpoint.com/tutorial/machine -learning/images/support-vector-machine-algorithm.png)
[19] Fig8:(https://static.javatpoint.com/tutorial/machine -learning/images/support-vector-machine-algorithm4.png)
[20] Fig9:(https://static.javatpoint.com/tutorial/machine -learning/images/support-vector-machine-algorithm9.png)
[21] Fig10:(https://static.javatpoint.com/tutorial/machine -learning/images/random-forest-algorithm.png)