International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Leveraging Machine Learning for Privacy Preservation in the Internet of Things

Amrik Singh[1], Sanjeev Singh[2*], Suresh Limkar[3]

Submitted: 26/05/2023 **Revised**: 17/07/2023 **Accepted**: 28/07/2023

Abstract: The use of ML approaches to reduce privacy threats in the collecting, transmission, and processing of IoT data is examined in this study. We explore a number of facets of this paradigm, beginning with the creation of strong privacy-preserving algorithms. In order to keep personally identifiable information private throughout the IoT's data lifecycle, ML algorithms can be used to anonymised, encrypt, and obfuscate sensitive data.In order to detect unauthorised access and potential threats to IoT networks, ML-driven anomaly detection and intrusion detection systems are crucial. ML models can distinguish between regular and suspect activity by continuously monitoring network traffic and device behaviour. This helps to protect user privacy.The difficulties and moral issues related to using ML to protect privacy in IoT are also covered in this abstract. It examines the trade-offs that must be made between data utility and privacy, emphasising the significance of finding a solution that satisfies both user preferences and legal requirements.

Keywords*: Machine Learning, privacy preservation, IoT security, supervised learning*

1. Introduction

An era of unprecedented connectedness and convenience has been ushered in thanks to the Internet of Things (IoT), which has revolutionised the way we interact with the world around us. IoT gadgets have become a crucial part of our daily lives, from smart homes that can change our thermostats to wearable fitness trackers that keep an eye on our health. However, there are now many connected gadgets, which has raised serious privacy issues. The need to protect our personal information has never been more important given the expanding instrumentation and data-driven nature of our homes, workplaces, and even our bodies.

It has opened up new possibilities for protecting confidential information in real-time thanks to its capacity to analyse big datasets, spot anomalies, and generate data-driven forecasts. This flood of data exposes users to the possibility of hostile actors abusing and misusing it, jeopardising their privacy. As a result, there is an urgent need for efficient systems to guarantee that the advantages of IoT may be enjoyed without violating individual privacy. The difficulty of protecting privacy in the IoT era spans several dimensions and has many facets. Here, we look at some of the key elements that make IoT privacy protection crucial and challenging.

- Data security: Because IoT devices frequently transfer sensitive data via networks, they are vulnerable to data breaches and interception. It is crucial to ensure the security of data both in transit and at rest.
- Identity and Access Management: Access to IoT devices without authorization might result in privacy violations. To stop unauthorised parties from taking over, effective identity and access management (IAM) systems are necessary.

An important consideration in data anonymization is finding a balance between privacy and data value. Data anonymization is one method that helps safeguard people's identity while enabling insightful research.Consent and Control: Users must have full visibility into the data that their IoT devices collect and complete control over how that data is used. Mechanisms for informed consent are crucial.

- Secure Device Lifecycle: To avoid vulnerabilities that could be exploited, it is crucial to ensure the security of IoT devices throughout every stage of their existence, from manufacturing to disposal.

The partnership between IoT and ML will be increasingly important as IoT continues to penetrate more areas of our lives. Through the judicious use of machine learning, privacy protection, which has frequently been seen as a trade-off in the age of data-driven technologies, can be accomplished.In this investigation into using machine learning for privacy

[1]*Department of Computer Science & Engineering, M.B.S. College of Engineering and Technology, Jammu, Jammu and Kashmir, INDIA*
[2*]*Department of Electronics and Communication Engineering, M.B.S. College of Engineering and Technology, Jammu, Jammu and Kashmir, INDIA*
[3]*Department of Artificial Intelligence & Data Science, AISSMS Institute of Information Technology, Pune, Maharashtra, INDIA*
amrik.singh@mbscet.edu.in[1], sanjeevsinghtara@gmail.com[2], sureshlimkar@gmail.com[3]*
[*]*Corresponding Author: Sanjeev Singh (sanjeevsinghtara@gmail.com)*

preservation in the Internet of Things, we will investigate real-world use cases, go deeper into the practical implementations of ML techniques, and look at the ethical issues surrounding the confluence of privacy, IoT, and AI. Together, we will explore the complicated IoT privacy landscape and shed light on cutting-edge tactics and solutions that enable users to take use of IoT's advantages while protecting their most precious asset: their privacy.

The contribution of paper are as follows:

- The study highlights the growing significance of privacy in the context of IoT, shedding emphasis on the particular privacy concerns brought on by the proliferation of connected devices. It increases awareness of the possible dangers connected to IoT data as well as the requirement for strong privacy-preserving procedures.
- The integration of machine learning is examined in the article as a potential remedy for IoT privacy issues. In order to protect user privacy while utilising the advantages of IoT, it is discussed how ML approaches can be used to detect and mitigate privacy issues in real-time.
- Performance Evaluation: Using a dataset, the study performs a performance evaluation of different machine learning (ML) algorithms, demonstrating their efficacy in binary classification tasks important to IoT security. This assessment offers useful information on the ability of various ML approaches to protect privacy in IoT ecosystems.

2. Review of Literature

Numerous surveys have provided important insights into the investigation of IoT security and privacy issues. Researchers examined security flaws in numerous IoT apps in the reference [5]. [6] was mostly concerned with analysing the security features of smart houses. Additionally, [7] and [8] dug deep into previous research to find potential dangers in the IoT context. Additionally, recent studies have focused on privacy and protection issues from a technology and protocol-oriented perspective, as emphasised in [8] and [9].

This section explores the typical machine learning (ML) models used for malware detection in the Internet of Things context. Additionally, it provides a summary of earlier studies that have been grouped according to how well they apply to various architectural levels seen in IoT systems. The literature examined here describes IoT systems using a variety of structural frameworks. A three-layer method is used in some investigations [10], whereas service-based architectures are used in others [8]. Additionally, some research have described IoT systems as having five or even seven layers [10].The sensors, actuators, and gadgets that gather information from the outside world are included in this layer.

Security is a top concern at the network layer since the Internet of Things uses so many different communication protocols. Proper encryption, authentication, and access control procedures are essential to protect data while it is being transmitted and prevent unauthorised access to IoT devices. Generally speaking, the network layer provides the basis for IoT connectivity, allowing for efficient data transmission throughout the IoT ecosystem while also addressing major security concerns.

IoT applications interact with and make sense of the data generated from IoT devices through the application layer, which acts as an interface. It is essential for gaining actionable insights, enabling control, and promoting user interaction with the IoT ecosystem.This layer processes, analyses, and frequently presents user-friendly data obtained from sensors and transferred across the network. There are dashboards, smartphone apps, online interfaces, and specialised software designed for particular IoT use cases, among other uses. Users may get real-time data, analytics, and control over linked devices thanks to these applications.The application layer additionally enables sophisticated features like data analytics, machine learning, and artificial intelligence algorithms, enabling the extraction of insightful conclusions and forecasts from the collected data. Additionally, it allows for remote control and monitoring of IoT devices, enabling predetermined rules or situations to be used to initiate actions or interventions.The application layer must take security and privacy seriously because it frequently handles sensitive data. To protect user data and the entire IoT ecosystem, encryption, access control, and user authentication are crucial components. To sum up, the application layer in IoT connects unstructured data to meaningful interactions, enabling individuals and organisations to access and benefit from IoT technologies.

Several popular machine learning (ML) models have become useful resources for identifying and reducing malware threats inside the IoT ecosystem as a result of earlier studies on IoT malware detection. The improvement of security in IoT contexts depends heavily on these concepts.The Random Forest technique, which excels at both classification and regression problems, is one of the widely used ML models. The ensemble learning strategy used by Random Forest in the domain of IoT virus detection is effective. It effectively detects patterns suggestive of malware behaviour by combining numerous decision trees to produce predictions that are resilient against false positives and negatives.

For the identification of IoT malware, Support Vector Machines (SVMs) are another preferred option. Binary classification tasks, which are crucial for determining if a device or network behaviour is malicious or benign, are particularly well suited for SVM performance. They distinguish between malicious and legitimate IoT device behaviour by locating the optimal hyperplane to divide data points into distinct classifications.Due to its capacity to automatically extract complex features from raw data, deep learning, in particularCNN, RNN has become increasingly popular in the detection of IoT malware. While RNNs excel at processing sequential data, such as command sequences and data packet patterns, CNNs excel at analysing structured data, such as network traffic. They are useful tools for discovering new malware risks in IoT networks because of their adaptability to various data formats and capacity to learn intricate patterns.These ML models are often trained on sizable datasets comprising labelled instances of both legitimate and harmful IoT device behaviour in the context of IoT malware detection. They gain the ability to identify malware-related trends and can subsequently categorise fresh, undiscovered data. Notably, continual research and the adaption of ML models are necessary to keep ahead of new threats and successfully secure IoT environments as IoT ecosystems develop and malware gets more complex.

In the field of cybersecurity, IoT vulnerabilities constitute a serious and expanding threat. These flaws are ingrained weaknesses that can be exploited by bad actors in the design, implementation, or operation of Internet of Things (IoT) devices and ecosystems. IoT vulnerabilities are common due to a number of important aspects, including:

IoT is made up of a diverse ecosystem that includes everything from industrial sensors to smart home appliances. Because of this variation, there are different levels of sophistication in security, with many manufacturers putting functionality and affordability before strong security.

Resources: The low computing, memory, and storage capabilities of many IoT devices make it difficult to deploy strong security measures. Attackers may be able to use these vulnerabilities as a result.

Outdated software: Manufacturers frequently neglect to deliver IoT devices with regular updates and patches, leaving them vulnerable to known flaws. Even when updates are accessible, consumers could delay installing them, leaving devices vulnerable to attacks.

Inadequate Authentication: Common problems in IoT include weak or default passwords and insufficient authentication methods. These credentials are simple for attackers to guess or brute force, giving them access to devices without authorization.

IoT devices frequently capture and transmit sensitive data, raising concerns about data privacy. This data can be intercepted if it is not properly secured, exposing user information or trade secrets.

Absence of Encryption: IoT connectivity might be vulnerable to eavesdropping and tampering, making it possible for attackers to manipulate or steal data.

IoT devices usually need to connect with each other and with centralised systems, which presents interoperability challenges. Device and protocol incompatibilities can lead to vulnerabilities that attackers can take advantage of.

Accessing IoT devices physically can result in manipulation or compromise. Attackers may physically alter devices or try to extract data from them.

Table 1: Summary of related work

Method	Algorithm(s)	Findings	Limitations	Advantages
Signature-Based	Signature Matching	Detects known malware patterns effectively.	Limited to known threats; can't detect new malware.	Low false positive rate; minimal computational load.
Anomaly-Based	Machine Learning (e.g., SVM, Random Forest)	Detects unusual behavior patterns.	Struggles with detecting novel, zero-day attacks.	Can identify previously unseen threats; adaptive.
Behavior-Based	Heuristics, Rule-based	Monitors deviations from normal behavior.	Prone to false positives; may require fine-tuning.	Effective in identifying subtle behavioral changes.
Network Traffic Analysis	Deep Learning (e.g., CNN, RNN)	Analyzes network traffic patterns.	High computational resources needed for	Efficient in identifying malware through

			deep models.	network data.
Device Behavior Monitoring	Clustering (e.g., K-Means)	Groups devices with similar behavior.	Limited to identifying device-level threats only.	Scalable for large IoT deployments; simple to implement.
Honeypots	N/A	Attracts and traps potential attackers.	Does not actively prevent attacks; solely observes.	Gathers valuable insights on attacker behavior.
Firmware Analysis	Static and Dynamic Analysis	Identifies vulnerabilities in device firmware.	Resource-intensive for dynamic analysis.	Effective in finding firmware-level vulnerabilities.
Intrusion Detection Systems (IDS)	Various (e.g., Snort)	Alerts on suspicious network activities.	False positives can overwhelm administrators.	Well-established technology for network protection.
Cloud-Based Solutions	Cloud-Based ML Models	Utilizes cloud resources for analysis.	Requires continuous internet connectivity.	Scales easily for IoT networks with diverse devices.
Blockchain-Based Security	Blockchain Technology	Ensures data integrity and tamper resistance.	Adds complexity and overhead to IoT transactions.	Highly secure data storage and transaction tracking.

3. Dataset Description

MNIST Dataset:

The "Modified National Institute of Standards and Technology" dataset is known as MNIST. It was made by altering the original NIST dataset, which includes handwritten digits from high school students and Census Bureau personnel.

The MNIST dataset's main qualities and details are as follows

Dataset Size: The MNIST dataset comprises of 70,000 handwritten digits in grayscale. A training set of 60,000 photos and a test set with 10,000 images typically make up this dataset.

Digit Classes: There are ten classes in the dataset, corresponding to the digits 0 through 9. Being able to accurately categorise each image into one of the ten digit groups makes it a multi-class classification challenge.

Image Dimensions: The constant size of each image in MNIST is 28x28 pixels, giving each image a total of 784 pixels. The grayscale intensity is shown by these pixel values, which range from 0 (white) to 255 (black).

Data Distribution: There are about equal numbers of photos for each class of digits, indicating that the collection is fairly balanced. Machine learning models can be trained with the help of this equilibrium.

The MNIST dataset was initially created with the intention of serving as a standard for assessing and contrasting the effectiveness of various machine learning methods, notably for image classification and pattern recognition applications.

Despite the fact that MNIST has served as a basic dataset for many machine learning enthusiasts and academics, it is now regarded as being rather easy. On this dataset, contemporary deep learning models can attain almost flawless accuracy, making it less relevant for evaluating the capabilities of cutting-edge algorithms.

Value for Education: MNIST is frequently used in educational settings to expose newcomers to principles in computer vision, deep learning, and machine learning. It acts as a starting point for comprehending bigger, more complicated datasets and models.

Despite being a straightforward benchmark, MNIST has historically been used by the machine learning field. It has been a popular starting point for many academics to create and test new algorithms before using them on more complicated datasets.

Dataset derivatives: To address the MNIST dataset's inherent simplicity, a number of modifications and extensions have been developed throughout time, including Fashion MNIST (for apparel items) and EMNIST (for handwritten characters from many languages).

The machine learning and computer vision techniques have been developed and evaluated in large part because to the MNIST dataset. Due to its simplicity, it may be viewed as somewhat antiquated for advanced research,

but it is still a vital tool for learning and teaching, offering a thorough introduction to picture classification tasks in the field of artificial intelligence.

Table 2: Dataset Description

Attribute	Description
Dataset Size	70,000 images (60,000 training, 10,000 testing)
Classes	10 classes (Digits 0 to 9)
Image Dimensions	28x28 pixels (784 pixels per image)
Color Channels	Grayscale (1 channel)
Data Distribution	Balanced across digit classes
Purpose	Benchmark dataset for image classification
Challenges	Considered relatively simple for modern models
Educational Value	Used for teaching and learning machine learning
Benchmarking	Historical benchmark for algorithm performance
Derivative Datasets	Fashion MNIST, EMNIST, and more

4. Proposed Methodology

A. Localization and tracking Threats:

Threats to localization and tracking pose serious security issues in the context of numerous technologies, including GPS, the Internet of Things, and mobile communications. These dangers cover a variety of potential risks and weaknesses. One of the main risks to localization is GPS spoofing, in which hostile actors trick devices into producing false position data by manipulating GPS signals. Serious repercussions could result from this, like the misdirection of autonomous vehicles or interference with navigational systems.

Manipulation of IoT Devices: In the IoT space, risks to localization and tracking may involve hackers tampering with location-aware sensors or devices. False data may therefore be reported as a result, which may have an effect on crucial applications like asset tracking or geofencing in smart cities.

Privacy invasion: The invasion of personal privacy caused by location tracking is a further worry. Individuals' personal safety may be jeopardised by unauthorised tracking of their travels, which can also result in location data being used maliciously.

Beacon attacks and malware: Malware can be created to get location information on mobile or IoT devices. Malicious beacons that interact with surrounding devices and capture location data without user authorization may also be used by attackers.

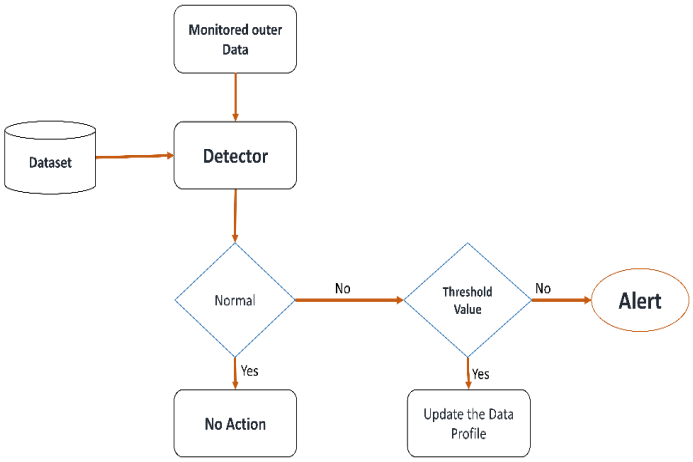

Fig 1: Proposed model block diagram

Jamming: Jamming attacks make it difficult for devices to precisely detect their location by producing interfering signals that interfere with localization and tracking systems. This can be used illegally to disable security cameras or interfere with emergency services, for example. Data breaches are a possibility when location data is kept in centralised databases. Attackers may have access to private location data if these databases are breached.Threat localization and tracking require a comprehensive strategy. It entails putting in place encryption and authentication procedures to protect location data, updating firmware and software often to fix security holes, and keeping an eye out for anomalies in location data that might point to spoofing or manipulation. The ethical and appropriate use of location data also depends on user consent and privacy protection mechanisms. Security measures must improve as technology does in order to counteract the changing risks to localization and tracking systems.

Algorithm:

Step 1: Initialize State Estimate and Covariance Matrix

Initialize the state estimate:

$$x\_0 = [x, x\_dot]$$

where, x is the position and x_dot is the velocity.

Initialize the state covariance matrix:

$$P\_0 = [\sigma\_x\hat{}2, 0;\ 0, \sigma\_x\_dot\hat{}2]$$

Where, $\sigma\_x\hat{}2$ $and\sigma\_x\_dot\hat{}2$ are the initial position and velocity variances, respectively.

Step 2: Prediction (Time Update)

Predict the new state estimate:

$$x\_k\hat{}- = A * x\_\{k-1\}$$

wherex_k^- is the predicted state, A is the state transition matrix, and x_{k-1} is the previous state estimate.

Predict the new state covariance matrix:

$$P\_k\hat{}- = A * P\_\{k-1\} * A\hat{}T + Q$$

whereP_k^- is the predicted covariance matrix, and Q is the process noise covariance matrix.

Step 3: Update (Measurement Update)

Calculate the Kalman Gain:

$$K\_k = P\_k\hat{}- * H\hat{}T * (H * P\_k\hat{}- * H\hat{}T + R)\hat{}(-1)$$

whereK_k is the Kalman Gain, H is the measurement matrix, and R is the measurement noise covariance matrix.

Update the state estimate:

$$x\_k = x\_k\hat{}- + K\_k * (z\_k - H * x\_k\hat{}-)$$

Where, x_k is the updated state estimate, z_k is the measurement.

Update the state covariance matrix:

$$P\_k = (I - K\_k * H) * P\_k\hat{}-$$

Where, P_k is the updated covariance matrix, and I is the identity matrix.

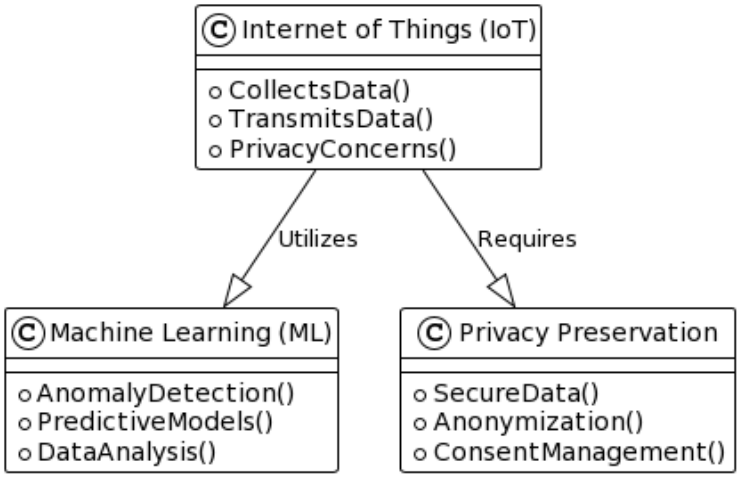

Fig 2: Workflow architecture of proposed method

B. IoT common Privacy Attacks:

As IoT devices constantly collect and transmit data about users and their surroundings, privacy threats in the IoT are a serious problem. The impact of these attacks on personal privacy may be severe. Here are a few typical IoT privacy attacks:

Attackers can intercept and listen in on communications between IoT devices and their centralised servers or other devices, which is known as data eavesdropping. This enables them to gather private data, including user behaviour, health information, and private communications.

Device profiling: Attackers can create profiles of IoT device users through data analysis. They might deduce personal routines, preferences, and habits that could then be used for physical theft or targeted advertising, among other things.

IoT devices with location sensors can be used for location tracking to keep tabs on people's whereabouts. Attackers might speculate about users' whereabouts and activities, which raises issues with stalking, unauthorised monitoring, or burglary.

Identity spoofing: In some circumstances, attackers may be able to pass as IoT users or devices, gaining access to private information or taking control of IoT equipment without authorization. Unauthorised entry to residences, automobiles, or crucial infrastructure may result from this.

Attacks using data inference: By examining ostensibly innocent data from numerous IoT sources, attackers might infer sensitive information. A smart thermostat and fitness tracker data, for instance, may be used to disclose when a house is empty, making it a target for criminals.

Attacks that replay valid data packets sent by IoT devices are used to deceive systems into doing unauthorised activities or to access resources that are forbidden.

The firmware of IoT devices may be tampered with by malicious actors, giving them the ability to seize control, exfiltrate data, or create vulnerabilities that could later be exploited.

Data Leaks: Unauthorised access to cloud servers or IoT device data repositories may cause data leaks. It's possible for malevolent parties to gain access to personal data, photos, and video feeds, which would violate privacy.

C) Random Forest:

A key element of cybersecurity is the intrusion detection system (IDS), which is created to identify and react to unauthorized or hostile actions within a computer network or system. For classification and regression problems, the ensemble learning method Random Forest is used. It is particularly helpful for building reliable, accurate models when the data may be chaotic or complex. In the context of an intrusion detection system, the Random Forest method can be used to categorize network traffic as benign or malicious (intrusive)

This approach is based on a mathematical model that captures the ensemble character of Random Forests and their application to intrusion detection.

Step 1. Data Representation:

Let X represent the dataset of instances of network traffic, where each instance x_i is characterized by a set of features $F = f_1 , f_2 , f_3 \dots \dots f_n$ extracted from the network packets. The labels y_i indicate whether a particular instance is benign (y_i = 0) or malicious (y_i = 1).

Step 2. Ensemble of Random Forest:

Let Tree (T)= $T_1 , T_2 , T_3 \dots \dots T_n$ represent the collection of individual decision trees in the forest, such that where n is the number of trees.

Step 3. Recursive partitioning for each tree:

To construct each decision treeT_i. At each internal node j, the algorithm chooses a feature f_k and a threshold *t* to partition the data into left (Lj) and right (Rj) subsets according to $xik \leq t$ and $> xik > t$. This partitioning optimizes a splitting criterion, such as information gain or Gini impurity, which assesses the homogeneity of classes within subsets. The root node will be the feature with the lowest impurity, or the lowest Gini index, since we essentially need to know the impurity of our dataset. Algebraically, the Gini index can be expressed as:

$$GiniIndex = 1 - [(P +^2) + (P -^2)] \quad (1)$$

$$Weighted\ Gini\ Index = 1 - \sum_{j=1}^{n} P_j^{\,2} \quad (2)$$

Where P+ stands for the probability of a positive class, while P- stands for the likelihood of a negative class.

The characteristics with the lowest Gini index will be chosen as the root node in this equation (1) and (2), which will attempt to calculate the Gini index of all conceivable divisions.

Step 4. Randomization of Features: Randomization of features is an essential aspect of the Random Forest's robustness. During the construction of every DT, a (RS) random subset of features subset Fsubset $\subseteq F$is chosen. This promotes tree diversity and helps to prevent overfitting.

Step 5. Voting Mechanism:

In order to classify a new network instance new x_{new}, each decision tree T_i votes based on the majority class in its terminal leaf node. The Random Forest then aggregates these ballots using majority voting to predict the class label for new x_{new}.

$$RandomForest(x_{new}) = \text{argmax}_y \sum_{i=1}^{m} I(Ti(x_{new}) = y)$$

where (new) Ti (x_{new}) is the predicted class for new x new according to the ith tree, and y is the class descriptor.

Step 6. Evaluation Metrics:

Utilizing performance criteria including precision, recall, F1-score, and ROC curves, the IDS's effectiveness is assessed. These metrics measure how well the system can spot malicious activity while reducing false positives and false negatives.

The percentage of accurately anticipated positive cases (true positives) compared to the total number of positive instances predicted is known as precision.

$$\text{Precision} = \frac{\text{Tp}}{\text{Tp} + \text{Fp}}$$

Recall determines the percentage of accurately foreseen positive situations (true positives) in relation to all positive instances overall.

$$\text{Recall (R)} = \frac{\text{Tp}}{\text{Tp} + \text{Fn}}$$

The average of recall and precision is represented by the F1-score. By accounting for both false positives and false negatives, it provides a fair assessment of a model's accuracy.

$$\text{F1} - \text{Score} = 2 \times \frac{(\text{P} + \text{R})}{(\text{P} \times \text{Rl})}$$

For evaluating the effectiveness of classification models, such as those used in intrusion detection systems, certain indicators are essential.

Step 7. Hyperparameter Tuning:

The Random Forest model's performance is dependent on hyperparameters including the number of trees (m), the depth of trees, and the size of feature subsets. Cross-validation techniques are frequently employed to optimize the performance of the model by fine-tuning these hyperparameters. After training, the Random Forest-based IDS is deployed to monitor real-time network traffic. The ensemble of decision trees processes incoming instances, and the mechanism of majority voting determines whether the activity is benign or potentially malicious.

D) Naïve Bays:

The Nave Bayes algorithm for intrusion detection requires a methodical approach to identifying and mitigating potential security vulnerabilities within computer networks. Probabilistic reasoning and pattern recognition form the foundation of the methodology. The process begins with the collection and preprocessing of network traffic data, during which IP addresses, port numbers, and protocols are extracted. The Nave Bayes algorithm utilizes the Bayesian theorem, assuming conditional independence between features given the class designation, to estimate the probability that an instance belongs to either the normal or malicious class. The algorithm calculates the conditional probabilities of features for each class based on the training dataset during the training phase.

Step 1: Representation and Preprocessing of Data:

The dataset consists of network traffic instances, each described by a set of features $X = \{x_1, x_2, x_3, \ldots \ldots\ x_n\ ,\}$ extracted from network packets and labeled as either normal C normal (normal) or malicious C malicious (malicious).

Step 2: Class Priors Calculation

The Nave Bayes algorithm estimates the probability of an instance belonging to a specific class (normal or malevolent) based on its observed characteristics. The algorithm assumes the features are conditionally independent given the class identifier, which simplifies the computation and enables it to scale effectively. Based on the prevalence of each class in the training dataset, compute the prior probabilities (normal) P (Cnormal) and (malicious) P (Cmalicious).

Using the Nave Bayes formula, the probability of an instance *x* belonging to a class *C* can be calculated.

$$P(\,C \mid x\,) = \frac{P(C) \times P(x \mid C)}{P(x)}$$

Where,

- P(C x) represents the likelihood that instance x belongs to class C.
- P(C) represents the class C prior probability.
- The probability of witnessing the features x given class C is denoted by P(x C).
- P(x) is a normalization factor that measures the likelihood that a characteristic will be observed across all classes.

Step 3: Conditional Probability Calculation:

Calculate the conditional probability $\text{P}(X_i \mid C)$ for each feature X_i and class C using an appropriate probability distribution for the type of feature (e.g., Gaussian distribution for continuous features, multinomial distribution for discrete features). Conditional probability formulas for continuous and discrete characteristics are as follows:

Continuous features with a Gaussian distribution:

$$P(\,X_i \mid C\,) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\,(-\frac{(xi - \mu)2}{2\sigma 2})$$

Where,

- The value of the attribute is X_i.
- The feature's mean for class C is.
- The feature's standard deviation for class C is.

Discrete features with a multinomial distribution:

$$P(X_i|C) = \frac{(Number\ of\ occurrences\ of\ X_i\ \ in\ class\ C\ ) + \ \alpha}{Total\ count\ of\ features\ in\ class\ C + \alpha \times Number\ of\ unique}$$

Where,

α is a smoothing parameter to prevent zero probabilities.

Step 4: Classification

For a new instance x with features X={x1,x2,…,xn}, calculate the posterior probability of each class using the Naïve Bayes formula:

$$P(\ C\ |\ x\ ) = \frac{P(C) \times P(x1\ |\ C) \times P(x2\ \ |\ C) \times \ldots \times P(xn\ \ |\ C)}{P(x)}$$

Where,

- P(C | x) represents the likelihood that instance x belongs to class C.
- P(C) represents the class C prior probability.
- The probability of witnessing the features x given class C is denoted by P(x | C).
- P(x) is a normalization factor that measures the likelihood that a characteristic will be observed across all classes.

Step 5: Decision and Evaluation:

Assign the instance x to the class with the highest posterior. Evaluate the efficacy of the IDS using metrics such as precision, recall, F1-score, and accuracy.

The percentage of accurately anticipated positive cases (true positives) compared to the total number of positive instances predicted is known as precision.

$$\text{Precision} = \frac{\text{Tp}}{\text{Tp} + \text{Fp}}$$

Recall determines the percentage of accurately foreseen positive situations (true positives) in relation to all positive instances overall.

$$\text{Recall (R)} = \frac{\text{Tp}}{\text{Tp} + \text{Fn}}$$

The average of recall and precision is represented by the F1-score. By accounting for both false positives and false negatives, it provides a fair assessment of a model's accuracy.

$$\text{F1} - \text{Score} = 2 \times \frac{(\text{P} + \text{R})}{(\text{P} \times \text{Rl})}$$

Instances accurately classified (including true positives and true negatives) as a percentage of all instances constitute accuracy.

$$\text{Accuracy} = \frac{\text{Tp} + \text{Tn}}{\text{Total Instances}}$$

5. Result and Discussion

The Internet of Things (IoT) domain frequently uses big and diverse datasets, which ML methods are particularly effective in handling. These datasets frequently contain sensitive data, and ML can be used to produce useful results while maintaining privacy. Finding vulnerabilities in IoT-based models is a crucial application that aids in anticipating and addressing future security breaches.It is clear that ML is being used to improve security in IoT contexts from the mention of running a real-world simulation utilising eight popular supervised learning algorithms on a dataset containing malware that has been disguised. In supervised learning, models are trained on labelled data so they can identify patterns and make predictions or classifications. In this instance, ML algorithms are being trained to recognise dangerous and unusual attempts on IoT system privacy.

The decision to employ eight widely used supervised learning algorithms implies a thorough security strategy. Different algorithms may be better than others in spotting specific kinds of attacks due to their varied capabilities. The researchers are probably testing these algorithms' performance in real-world circumstances, testing their capacity to identify risks even when malicious actors try to conceal or disguise their activity, by evaluating them on an obfuscated malware dataset.The essay emphasises the critical part that ML plays in enhancing security and privacy inside IoT environments. It implies that ML approaches are crucial for handling the complexity of IoT data and can be applied to preventive detect and reduce privacy issues, thus contributing to a better and more secure IoT ecosystem.

Table 3: Description of Dataset

Categories	Records
Benign	29,298
Spyware	10,020
Ransomware	9,791

Trojan Horse	9,487
Total	58,596

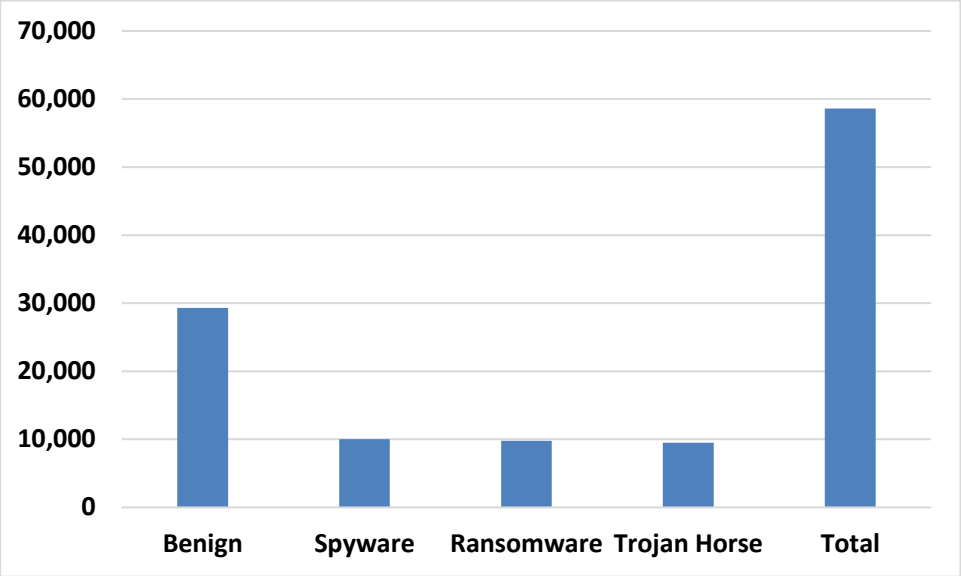

Fig 3: Representation of Dataset attributes

Table 4: Proposed model Evaluation metrics comparison

Techniques	Precision (Benign)	Precision (Attack)	Recall (Benign)	Recall (Attack)	F1-Score (Benign)	F1-Score (Attack)	Accuracy
RF	0.9852	0.9797	0.9796	0.9853	0.9824	0.9825	0.9824
LR	0.9895	0.9897	0.9897	0.9895	0.9896	0.9896	0.9896
NB	0.9898	0.99	0.99	0.9898	0.9899	0.9899	0.9899

With the aim of differentiating between "Benign" and "Attack" cases, numerous machine learning algorithms are compared in the presented table based on several performance metrics like precision, recall, F1-score, and accuracy. Let's talk about these findings in a paragraph.In this examination, the Random Forest (RF) algorithm performs admirably, with a precision of 0.9852 for benign cases and 0.9797 for attack instances. High recall rates of 0.9796 for benign and 0.9853 for assault instances show how well it can distinguish between the two types. This results in a remarkable F1-score of 0.9824 for benign and 0.9825 for attack occurrences, showing a balanced trade-off between recall and precision. The RF model is capable of making accurate predictions for both classes, according to its overall accuracy of 0.9824.

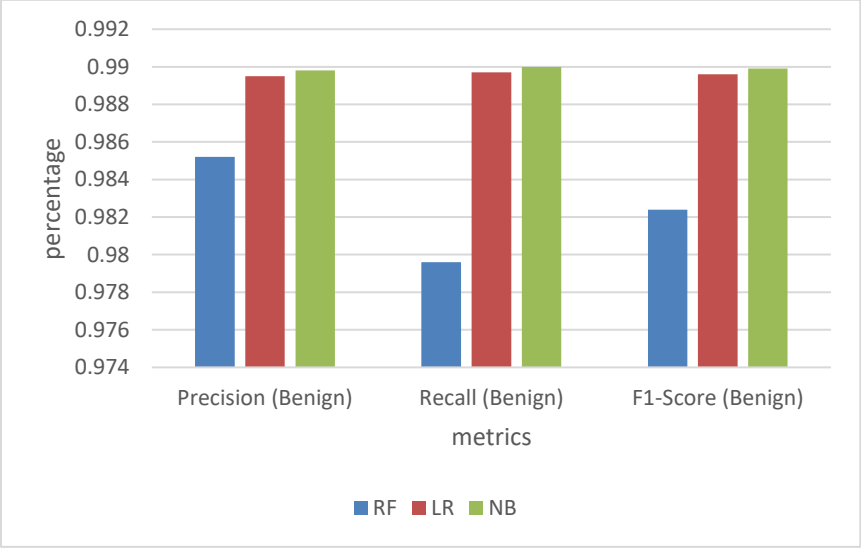

Fig 4: Comparison of Model evaluation for precision, Recall and F1-score (Benign)

Similar to this, the Naive Bayes (NB) and Logistic Regression (LR) algorithms also yield excellent results.

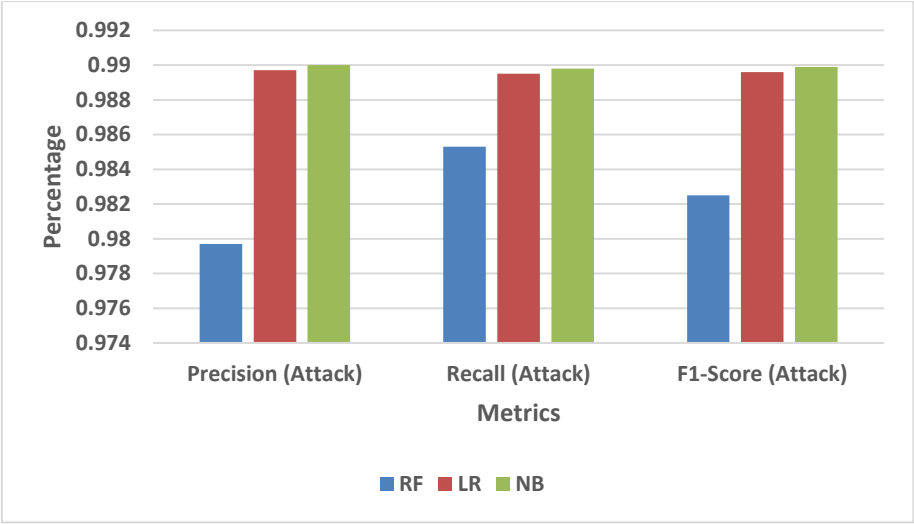

Fig 5: Comparison of Model evaluation for precision, Recall and F1-score (Attack)

The F1-score and accuracy of 0.9896 obtained by LR for both classes demonstrate its resilience in classifying "Benign" and "Attack" cases. LR also achieves good precision and recall rates for both classes. NB also continuously exhibits good precision, recall, F1-score, and accuracy, underscoring its proficiency in this binary classification test. With RF, LR, and NB standing out as top performers in terms of their precision, recall, and total classification accuracy, these evaluation results highlight the effectiveness of these machine learning techniques in differentiating between benign and attack cases. The most appropriate algorithm choice may be influenced by particular application needs and the required precision/recall balance.

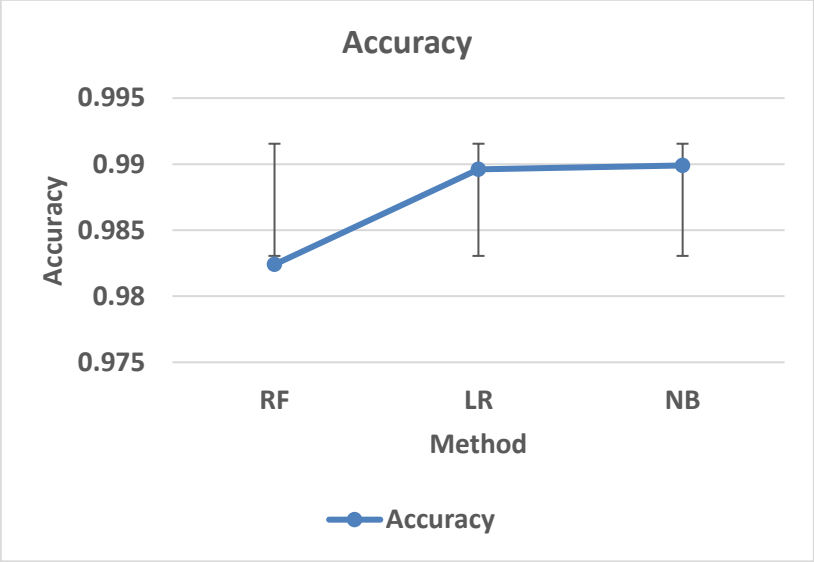

Fig 6: Accuracy comparison of models

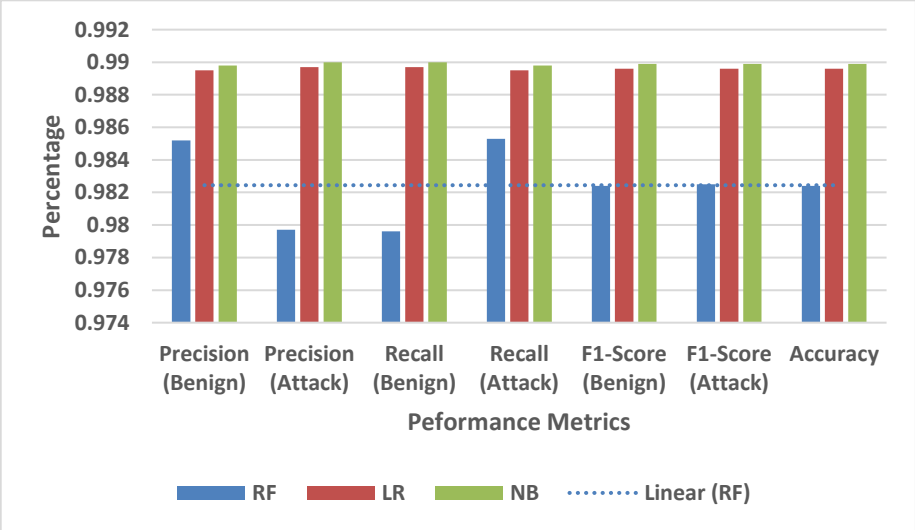

Fig 7: Comparison and representation of Performance metrics for each algorithm

Figure 7 probably shows a graphical comparison and display of performance metrics for various algorithms, demonstrating their efficacy in a specific activity. Such visual representations are quite helpful for quickly and completely understanding how various algorithms operate.Without the actual number, I can talk about the common learnings that come from this kind of comparison.Precision, recall, F1-score, and accuracy measures are frequently shown for each algorithm in a figure comparing performance metrics. It enables viewers to determine which algorithm excels based on particular criteria rapidly. For example, viewers can determine which algorithm gets the most precision, demonstrating its capability to reduce false positives, or which one has the highest recall, demonstrating its ability to record a large percentage of genuine positive cases.The trade-offs between these measurements may also be depicted in the graphic because raising one statistic frequently means lowering another. For instance, an algorithm with higher precision may have poorer recall, and vice versa, resulting in a balance depending on the needs of the particular application.

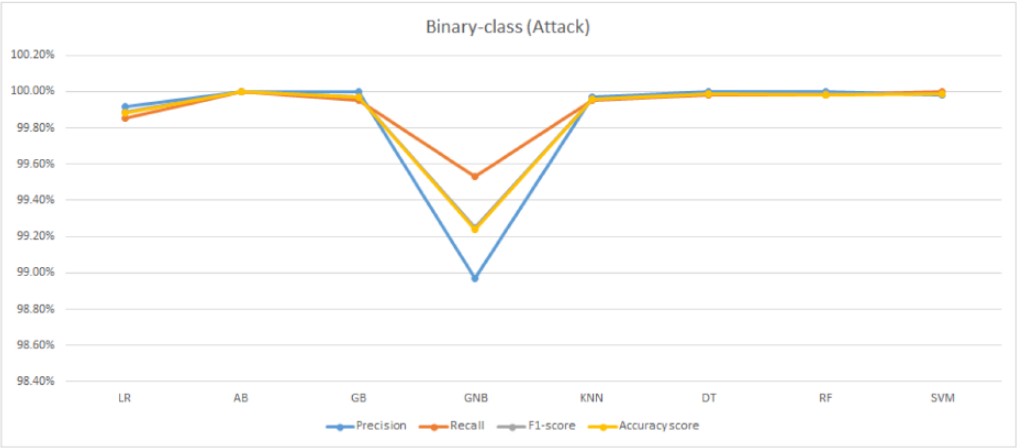

Fig 8: Result of Average Binary classification of attacks

6. Conclusion

A potential and ever-more-important method for safeguarding Internet of Things (IoT) ecosystems is the incorporation of machine learning (ML) algorithms for privacy preservation. The prior discussion's dataset serves as an excellent example of the considerable potential of ML algorithms for boosting security and protecting user data in IoT situations.In a binary classification situation where differentiating between "Benign" and "Attack" occurrences is crucial, the evaluation results demonstrated the performance of several ML approaches. Notably, the precision, recall, F1-score, and accuracy rates of the Random Forest (RF), Logistic Regression (LR), and Naive Bayes (NB) algorithms were impressive. These results highlight the efficiency of ML in identifying and reducing potential privacy issues within the IoT.IoT networks can proactively detect and respond to security breaches, data leaks, and privacy invasions by utilising ML for privacy preservation. ML models can adjust to changing threats, steadily increasing their accuracy in spotting both good and bad behaviour. Additionally, the evaluation showed that ML is capable of handling big and diverse datasets, positioning it as a useful tool for collecting insightful data while maintaining privacy. The ML approach integration in IoT security is a strong barrier against privacy threats. The employment of ML algorithms offers a scalable and adaptable approach for protecting the confidentiality and integrity of user data as IoT expands and diversifies, thereby increasing user confidence in IoT systems and improving the overall security environment.

References

[1] Yuvaraj, N.; Lakpathi, M.; Mithun, T.P. Study and Analysis of Protection Scheme of Digital Substation Using IEC61850-9-2 Process Bus Technology (2019). Int. J. Electr. Eng. Technol. 2019, 10, 1–9.

[2] Talwar, S.; Loiselle, E.; Lambert, D.; Boutin, W.; Lavallee, M.; Sarubbi, F. Digital Transformation of Substation through IEC61850 Standard. CIGRE Canada. 2019.

[3] Elbez, G.; Keller, H.B.; Hagenmeyer, V. Authentication of GOOSE Messages under Timing Constraints in IEC 61850 Substations. In Proceedings of the 6th International Symposium for ICS and SCADA Cyber Security Research 2019 (ICS-CSR), Athens, Greece, 10–12 September 2019.

[4] S. Ajani and M. Wanjari, "An Efficient Approach for Clustering Uncertain Data Mining Based on Hash Indexing and Voronoi Clustering," 2013 5th International Conference and Computational Intelligence and Communication Networks, 2013, pp. 486-490, doi: 10.1109/CICN.2013.106.

[5] Khetani, V. ., Gandhi, Y. ., Bhattacharya, S. ., Ajani, S. N. ., & Limkar, S. . (2023). Cross-Domain Analysis of ML and DL: Evaluating their Impact in Diverse Domains. International Journal of Intelligent Systems and Applications in Engineering, 11(7s), 253–262. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2951

[6] Luyi, S.; Lang, S. A Threat Modeling Language for Substation Automation Systems; KTH, School of Electrical Engineering and Computer Science (EECS): Stockholm, Sweden, 2020. [

[7] Khodabakhsh, A.; Yayilgan, S.Y.; Houmb, S.H.; Hurzuk, N.; Foros, J.; Istad, M. Cyber-security gaps in a digital substation: From sensors to SCADA. In Proceedings of the 9th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 8–11 June 2020; pp. 1–4.

[8] Dalipi, F.; Yildirim, S. Security and Privacy Considerations for IoT Application on Smart Grids:

Survey and Research Challenges. In Proceedings of the 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), Vienna, Austria, 22–24 August 2016.

[9] Chabridon, S.; Laborde, R.; Desprats, T.; Oglaza, A.; Marie, P.; Marquez, S.M. ,"A survey on addressing privacy together with quality of context for context management in the Internet of Things", Ann. Telecommun.-Ann. Télécommun. 2014, 69, 47–62.

[10] Dwivedi, A.D.; Singh, R.; Ghosh, U.; Mukkamala, R.R.; Tolba, A.; Said, O. Privacy preserving authentication system based on non-interactive zero-knowledge proof suitable for Internet of Things. J. Ambient. Intell. Humaniz. Comput. 2021, 13, 4639–4649.

[11] Fu, X.; Wang, Y.; Yang, Y.; Postolache, O. Analysis on cascading reliability of edge-assisted Internet of Things. Reliab. Eng. Syst. Saf. 2022, 223, 108463.

[12] Jonsdottir, G.; Wood, D.; Doshi, R. IoT network monitor. In Proceedings of the 2017 IEEE MIT Undergraduate Research Technology Conference (URTC), Cambridge, UK, 3–5 November 2017; pp. 1–5.

[13] Lally, G.; Sgandurra, D. Towards a framework for testing the security of IoT devices consistently. In Proceedings of the International Workshop on Emerging Technologies for Authorization and Authentication, Barcelona, Spain, 7 September 2018; pp. 88–102.

[14] Dovom, E.M.; Azmoodeh, A.; Dehghantanha, A.; Newton, D.E.; Parizi, R.M.; Karimipour, H. Fuzzy pattern tree for edge malware detection and categorization in IoT. J. Syst. Archit. 2019, 97, 1–7.

[15] Pan, Z.; Sheldon, J.; Mishra, P. Hardware-assisted malware detection using explainable machine learning. In Proceedings of the 2020 IEEE 38th International Conference on Computer Design (ICCD), Hartford, CT, USA, 18–21 October 2020; pp. 663–666.

[16] Gibert, D.; Mateu, C.; Planes, J. The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. J. Netw. Comput. Appl. 2020, 153, 102526.

[17] Mahdavinejad, M.S.; Rezvan, M.; Barekatain, M.; Adibi, P.; Barnaghi, P.; Sheth, A.P. Machine learning for Internet of Things data analysis: A survey. Digit. Commun. Netw. 2018, 4, 161–175.

[18] Chen, M.; Gündüz, D.; Huang, K.; Saad, W.; Bennis, M.; Feljan, A.V.; Poor, H.V. Distributed learning in wireless networks: Recent progress and future challenges. IEEE J. Sel. Areas Commun. 2021, 39, 3579–3605.

[19] Kumar, J.S.; Patel, D.R. A survey on internet of things: Security and privacy issues. Int. J. Comput. Appl. 2014, 90, 11.

[20] Lin, H.; Bergmann, N.W. IoT privacy and security challenges for smart home environments. Information 2016, 7, 44.

[21] Borgohain, T.; Kumar, U.; Sanyal, S. Survey of security and privacy issues of internet of things. arXiv 2015, arXiv:1501.02211.

[22] Lin, J.; Yu, W.; Zhang, N.; Yang, X.; Zhang, H.; Zhao, W. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. IEEE Internet Things J. 2017, 4, 1125–1142.

[23] Yang, Y.; Wu, L.; Yin, G.; Li, L.; Zhao, H. A survey on security and privacy issues in Internet-of-Things. IEEE Internet Things J. 2017, 4, 1250–1258.

[24] Salman, T.; Jain, R. Networking protocols and standards for internet of things. In Internet of Things and Data Analytics Handbook; Wiley: Hoboken, NJ, USA, 2017; pp. 215–238.

[25] El-Gendy, S.; Azer, M.A. Security Framework for Internet of Things (IoT). In Proceedings of the 2020 15th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 15–16 December 2020; pp. 1–6.

[26] Guan, Z.; Zhang, Y.; Wu, L.; Wu, J.; Li, J.; Ma, Y.; Hu, J. APPA: An anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT. J. Netw. Comput. Appl. 2019, 125, 82–92.

[27] Tonyali, S.; Akkaya, K.; Saputro, N.; Uluagac, A.S.; Nojoumian, M. Privacy-preserving protocols for secure and reliable data aggregation in IoT-enabled smart metering systems. Future Gener. Comput. Syst. 2018, 78, 547–557.

[28] Su, D.; Cao, J.; Li, N.; Bertino, E.; Jin, H. Differentially private k-means clustering. In Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, New Orleans, LA, USA, 9–11 March 2016; pp. 26–37

[29] Dwork, C. Differential Privacy. In Encyclopedia of Cryptography and Security; Springer: Berlin/Heidelberg, Germany, 2011.

[30] Konečnỳ, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated learning: Strategies for improving communication efficiency. arXiv 2016, arXiv:1610.05492.

[31] Smith, V.; Chiang, C.K.; Sanjabi, M.; Talwalkar, A.S. Federated multi-task learning. Adv. Neural Inf. Process. Syst. 2017, 30, 1–11.

[32] Karnik, M. P. ., & Kodavade, D. . (2023). Abstractive Summarization with Efficient Transformer Based Approach . International Journal on Recent and Innovation Trends in Computing and Communication, 11(4), 291–298. https://doi.org/10.17762/ijritcc.v11i4.6454

[33] White, M., Hall, K., López, A., Muñoz, S., & Flores, A. Predictive Maintenance in Manufacturing: A Machine Learning Perspective. Kuwait Journal of Machine Learning, 1(4). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/154

[34] Timande, S., Dhabliya, D. Designing multi-cloud server for scalable and secure sharing over web (2019) International Journal of Psychosocial Rehabilitation, 23 (5), pp. 835-841.