TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

From smart cities and healthcare to industrial automation and connected homes, the Internet of Things' (IoT) explosive growth has transformed a number of industries. Unprecedented levels of ease and efficiency are made possible by IoT devices, which facilitate smooth connectivity, real-time data sharing, and automation. However, because IoT networks frequently comprise a large number of networked devices that might serve as entry sites for cyber threats, its widespread adoption also brings with it serious security vulnerabilities.

Strong authentication, encryption, network security, and proactive threat detection are all components of a multi-layered strategy needed to improve IoT security. Critical systems can only be accessed by authorized people and devices thanks to the implementation of role-based access control (RBAC) and multi-factor authentication (MFA). End-to-end encryption guards against illegal interception and modification of data while it's in transit and at rest. Patch management and secure firmware updates keep devices updated with the newest security features, which helps reduce vulnerabilities. Additionally, by separating IoT devices from vital infrastructure, network segmentation prevents hacks from spreading. Real-time threat identification and response are made possible by AI-driven anomaly detection and behavioral analytics, which lowers the possibility of security breaches [1].

Lastly, by guaranteeing constant device and user verification, the integration of blockchain technology for secure identity management and zero-trust architecture enhances IoT ecosystems. IoT systems can strengthen their defences against changing cyberthreats by incorporating these security techniques.

Digital infrastructure is changing as a result of the Internet of Things' (IoT) incorporation into next-generation networks, opening up new opportunities for a variety of businesses. The growing number of sophisticated cyberattacks, which range from distributed denial-of-service (DDoS) attacks to data breaches and illegal access, makes ensuring the security and privacy of IoT devices a crucial task. More sophisticated and flexible security frameworks are required since traditional security methods are unable to keep up with the changing threat landscape. A viable strategy for protecting IoT ecosystems is the Zero Trust security model, which places a strong emphasis on stringent access constraints, ongoing authentication, and real-time threat detection.

Another important factor is data security, which calls for safe storage options, access restrictions, and privacy-preserving methods like differential privacy and homomorphic encryption to guarantee that private IoT data is

kept private and impenetrable. In order to mitigate the hazards associated with centralized data repositories, blockchain technology is being utilized more and more for secure transactions, decentralized identity management, and immutable audit trails.

Strong security measures are more important than ever to preserve operational integrity and data privacy as IoT technologies proliferate. The Zero Trust security model is becoming more and more popular as a result of emphasizing the value of strong defences [2].

The Internet of Things is transforming many facets of contemporary life, from smart homes to industrial automation. Networks are more vulnerable to attacks from malicious or hacked nodes, nevertheless, as they get larger and more intricate. Strong methods for identifying and reducing security risks must be developed since these breaches have the potential to compromise the effectiveness and dependability of IoT systems.

This strategy seeks to strengthen threat detection capabilities and increase IoT networks' resistance to intrusions by utilizing ensemble learning techniques. Adopting strong and astute security measures is crucial as IoT develops further in order to safeguard user privacy, data integrity, and the general dependability of next-generation networks.

The suggested approach integrates two ensemble learning algorithms and makes use of the Zero Trust security architecture to allay these worries. The ability of these algorithms to manage the complex and multifaceted nature of data in next-generation networks has led to their selection, guaranteeing a more reliable and secure Internet of Things ecosystem.

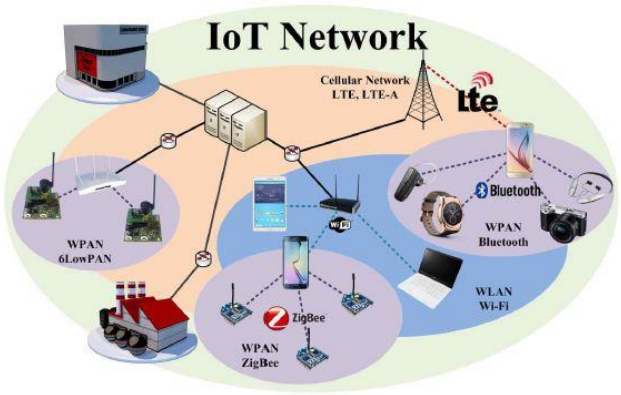

Fig 1. 1 IoT Device Network

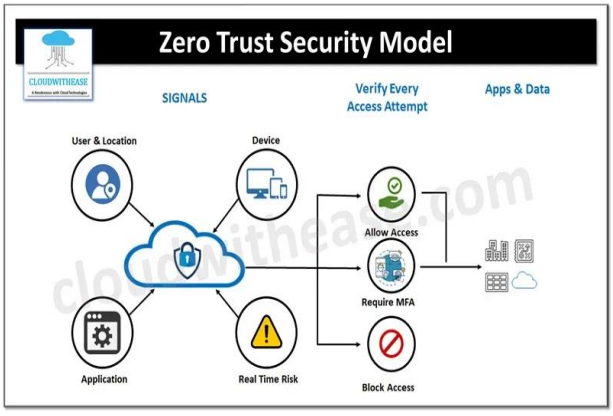

Fig 1. 2 Zero Trust Security Model

CHAPTER 2

LITERATURE SURVEY

- The study **"TrustNextGen: Security Aspects of Trustworthy Next-Generation Industrial Internet of Things"** by **G. Rathee**, **R. Iqbal**, **C. A. Kerrache**, and **H. Song** examines the important security issues and solutions related to next-generation Industrial Internet of Things (IIoT) systems. It was published in the IEEE Internet of Things Journal (Volume 11, Issue 15, August 2024, Pages 25568–25576).

- In Mobile Networks and Applications (Volume 28, Issue 1, February 2023, Pages 296–312), **I. H. Sarker**, **A. I. Khan**, **Y. B. Abushark**, and **F. Alsolami's** research article **"Internet of Things (IoT) Security Intelligence: A Comprehensive Overview, Machine Learning Solutions, and Research Directions"** offers a thorough examination of IoT security issues, investigates machine learning-based solutions for threat detection and mitigation, and identifies future research directions to improve IoT security intelligence.

- In the study **"Hybrid Privacy-Preserving Federated Learning Against Irregular Users in Next-Generation Internet of Things,"** published in the Journal of Systems Architecture (Volume 148, March 2024, Article No. 103088), **A. Yazdinejad**, **A. Dehghantanha**, **G. Srivastava**, **H. Karimipour**, and **R. M. Parizi** suggest a hybrid approach to improve privacy-preserving federated learning, addressing security threats posed by malicious or irregular users in next-generation IoT networks.

- **A. Rejeb, K. Rejeb, H. Treiblmaier, A. Appolloni, S. Alghamdi, Y. Alhasawi,** and **M. Iranmanesh**, in their research article **"The Internet of Things (IoT) in Healthcare: Taking Stock and Moving Forward,"** published in *Internet of Things* (Volume 22, July 2023, Article No. 100721), provide a comprehensive analysis of IoT applications in healthcare, examining current advancements, challenges, and future directions to enhance healthcare systems through IoT technologies.

- **Q.V. Pham, M. Zeng, O. A. Dobre, Z. Ding,** and **L. Song**, in their editorial article **"Guest Editorial: Special Issue on Aerial Computing for the Internet of Things (IoT),"** published in *IEEE Internet of Things Journal* (Volume 10, Issue 7, April 2023, Pages 5623–5625), introduce and discuss the significance of aerial computing in IoT, highlighting emerging research trends, challenges, and future directions in this evolving field.

- **P. Sun, S. Shen, Y. Wan, Z. Wu, Z. Fang, and X.-Z. Gao**, in their research article **"A Survey of IoT Privacy Security: Architecture, Technology, Challenges, and Trends,"** published in *IEEE Internet of Things Journal* (Early Access, March 1, 2024), provide a comprehensive review of IoT privacy security, exploring its architecture, technological advancements, key challenges, and emerging trends.

- **R. Singhai** and **R. Sushil**, in their research article **"An Investigation of Various Security and Privacy Issues in Internet of Things,"** published in *Materials Today: Proceedings* (Volume 80, January 2023, Pages 3393–3397), examine the critical security and privacy challenges in IoT, highlighting potential vulnerabilities, existing solutions, and future research directions.

- **M. Ahmid** and **O. Kazar**, in their research article **"A Comprehensive Review of the Internet of Things Security,"** published in *Journal of Applied Security Research* (Volume 18, Issue 3, July 2023, Pages 289–305), provide an extensive analysis of IoT security, discussing key threats, existing security mechanisms, and future challenges in safeguarding IoT ecosystems.

- **F. Xie, Z. Yin, A. Luo,** and **J. Yuan**, in their conference paper **"Prediction of Distribution Network Line Loss Based on Grey Relation Analysis and XGBoost,"** presented at the *IEEE 2nd International Conference on Big Data, Artificial Intelligence, and Internet of Things Engineering (ICBAIE)* in March 2021 (Pages 279–284), propose a predictive model using grey relation analysis and XGBoost to enhance the accuracy of distribution network line loss estimation.

AUTHOR	TITLE	YEAR	METHODOLOGY
G. Rathee, **R. Iqbal**, **C. A. Kerrache, H. Song**	TrustNextGen: Security Aspects of Trustworthy Next-Generation Industrial Internet of Things	2024	To increase confidence in IIoT systems, it combines blockchain-based identity management, cryptography approaches, and AI-driven anomaly detection.
I. H. Sarker, **A. I. Khan**, **Y. B. Abushark**, **F. Alsolami's**	Internet of Things (IoT) Security Intelligence: A Comprehensive Overview, Machine Learning Solutions, and Research Directions	2023	It investigates machine learning-based approaches to intrusion prevention, anomaly detection, and threat identification. The efficiency of different machine learning algorithms in improving IoT security is assessed.
A. Yazdinejad, **A. Dehghantanha**, **G. Srivastava**, **H. Karimipour**, **R. M. Parizi**	Hybrid Privacy-Preserving Federated Learning Against Irregular Users in Next-Generation Internet of Things	2024	To improve data secrecy, it combines secure aggregation, differential privacy, and cryptographic approaches. The robustness of the system is assessed using simulations and actual IoT scenarios.
A. Rejeb, K. Rejeb, H. Treiblmaier, A. Appolloni, S. Alghamdi, Y. Alhasawi, and M. Iranmanesh	The Internet of Things (IoT) in Healthcare: Taking Stock and Moving Forward	2023	Through case studies and industry reports, it looks at implementation difficulties, security issues, and technical improvements.

Q.V. Pham, M. Zeng, O. A. Dobre, Z. Ding, L. Song	Guest Editorial: Special Issue on Aerial Computing for the Internet of Things (IoT)	2023	It evaluates upcoming scientific trends through a selection of contributed papers. Security issues, and computational frameworks in aerial IoT systems are highlighted in the study.
P. Sun, S. Shen, Y. Wan, Z. Wu, Z. Fang, and X.Z. Gao	A Survey of IoT Privacy Security: Architecture, Technology, Challenges, and Trends	2024	It classifies current privacy-preserving methods, such as access control systems, anonymization, and encryption.
R. Singhai, R. Sushil	An Investigation of Various Security and Privacy Issues in Internet of Things	2023	It examines existing security frameworks, encryption methods, and authentication techniques to address vulnerabilities
M. Ahmid, O. Kazar	A Comprehensive Review of the Internet of Things Security	2023	It categorizes security mechanisms such as encryption, authentication, and intrusion detection to assess their effectiveness.
F. Xie, Z. Yin, A. Luo, J. Yuan	Prediction of Distribution Network Line Loss Based on Grey Relation Analysis and XGBoost	2021	It utilizes the XGBoost machine learning algorithm to develop a predictive model for accurate loss estimation.

CHAPTER 3

METHODOLOGY

3.1 OVERVIEW OF PROPOSED METHODOLOGY

The rapid proliferation of Internet of Things (IoT) devices has led to significant security challenges, as these devices often operate in resource-constrained environments with limited built-in security features. This research aims to enhance IoT security by developing a trust management framework that integrates advanced ensemble learning algorithms, specifically XGBoost and AdaBoost, to detect and mitigate malicious activities within IoT networks [3].

The Zero Trust security concept, which is the foundation of this methodology, holds that no network object, user, or device should be taken for granted. Rather, to guarantee a secure IoT ecosystem, ongoing authentication, verification, and trust assessment are crucial.

The suggested approach is especially made to tackle the following major IoT security issues:

1. **Recognizing IoT Nodes That Are Compromised:**

    - Device compromise, in which malevolent actors take over devices to carry out illegal operations, data exfiltration, or extensive cyberattacks, is a serious risk to IoT networks.
    - In order to identify possibly compromised nodes, the trust management framework uses machine learning-based behavioral analysis to identify unusual patterns in network traffic, authentication attempts, and device activity.
    - By learning from past attack patterns and current network activity, the combination of XGBoost and AdaBoost allows for extremely precise classification of trusted and untrusted devices.

2. **Reducing Online Dangers:**

- Numerous security risks might affect IoT setups, such as data breaches, in which private information is intercepted or disclosed.
- Unauthorized access occurs when hackers take advantage of lax authentication procedures. Distributed Denial-of-Service (DDoS) attacks occur when hacked Internet of Things devices (botnets) overload network capacity, disrupting services.

- To detect these threats in real time and implement the appropriate mitigation measures, the suggested architecture combines adaptive response mechanisms with machine learning-based anomaly detection.
- The technology can prevent security breaches by isolating or limiting access to suspect devices by continuously analyzing device trust scores and network activities.

3. Ensuring Secure and Efficient Decision-Making:

- Radiational rule-based security mechanisms often fail to adapt to evolving cyber threats in IoT environments.
- XGBoost and AdaBoost work together to refine these trust scores by learning from historical attack patterns and continuously updating security models.
- The trust management framework facilitates automated security decisions, such as restricting access to low-trust devices, triggering security alerts, or enforcing additional verification steps

This approach guarantees that IoT networks continue to be safe, robust, and able to identify and address security risks in real time by fusing ensemble learning techniques with Zero Trust principles. The framework's adaptability enables it to change with new attack trends, offering a strong security solution for contemporary IoT settings [4].

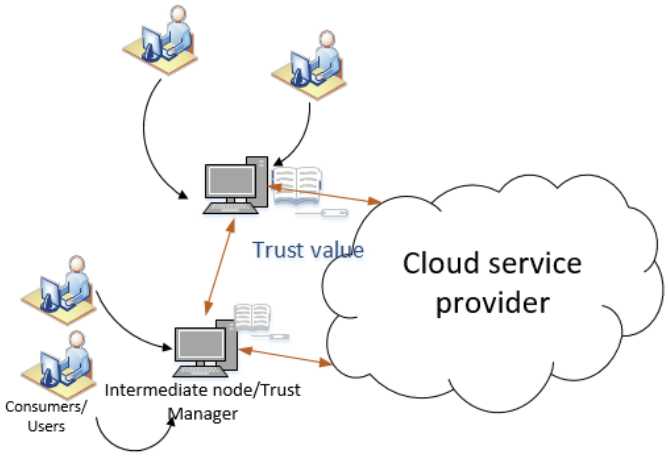

Fig 3. 1 Trust Management Framework

3.2 DATASET COLLECTION AND PREPARATION

A strong dataset is necessary for training and evaluating the machine learning models in order to create an IoT security framework that is both efficient and broadly applicable. Four significant IoT security datasets are used in this study; they each address a different facet of network security, anomaly detection, and trust assessment in IoT contexts. In order to ensure that the models can effectively generalize to actual IoT risks, the datasets offer a wide variety of attack scenarios, traffic patterns, and security events [5].

1. **IoT-23: Classified as Benign and Malicious IoT Network Traffic:**

   The Stratosphere Lab created the IoT-23 dataset, which is a publicly accessible compilation of actual IoT network traffic. It distinguishes between harmful and benign activity with 23 tagged captures.

   Key Features:

   - Covers various IoT malware families, including Mirai, Gafgyt, and Tsunami.
   - Captures both inbound and outbound network traffic for IoT devices.
   - Contains network-level features such as protocols, packet sizes, timestamps, and connection attempts.

   Relevance to This Study:

   - The dataset helps in identifying malicious behaviors in IoT traffic.
   - Enables training the machine learning models to distinguish between legitimate and suspicious communications.
   - Assists in building real-time intrusion detection mechanisms for IoT networks.

Fig 3. 2 Iot-23

2. N-BaIoT: IoT Device Botnet Attack Dataset:

For the purpose of researching botnet-driven cyberattacks in IoT ecosystems, the N-BaIoT dataset (Network-Based Anomaly Detection for IoT Botnets) was created. Network traffic logs from nine distinct IoT devices—including webcams and baby monitors—that were infected by botnets are included.

Key Features:

- Captures botnet attack traffic related to Mirai and Bashlite malware.
- Provides network flow statistics, including packet rates, bandwidth usage, and flow durations.
- Includes attack-specific features, such as command-and-control (C&C) traffic, scanning attempts, and denial-of-service (DoS) activities.

Relevance to This Study:

- Enables early detection of botnet-infected IoT devices.
- Helps train ensemble learning models (XGBoost and AdaBoost) to classify botnet-induced anomalies.
- Supports developing real-time mitigation strategies to prevent large-scale botnet attacks.

3. Industrial IoT (IIoT) Security Risks: Edge-IIoTset

The Edge-IIoTset dataset focuses on cybersecurity threats in Industrial IoT (IIoT) environments, where smart sensors, actuators, and edge computing devices play a crucial role in critical infrastructure such as manufacturing plants, energy grids, and smart cities [6].

Key Features:

- Covers five major attack categories:
  - **Reconnaissance Attacks:** Scanning and probing for vulnerabilities.
  - **Denial-of-Service (DoS) Attacks:** Overloading network resources.
  - **Man-in-the-Middle (MITM) Attacks:** Eavesdropping and altering communications.
  - **Injection Attacks:** Malicious code or command execution.
  - **Cross-Site Scripting (XSS) Attacks:** Targeting web applications.
- Contains real-world sensor and device telemetry data.
- Includes labeled normal and attack traffic for training intrusion detection models.

Relevance to This Study:

- Enhances model capability to detect sophisticated cyber threats in industrial IoT ecosystems.

- Provides insights into behavioral patterns of compromised IIoT devices.
- Improves trust scoring mechanisms by incorporating device-specific threat profiles.

4. **AutoTrust-IoTDS: Trust-Based Intrusion Detection:**

The AutoTrust-IoTDS dataset is designed for trust-based security evaluation in IoT networks. It integrates trust models with machine learning techniques to detect anomalous activities and potential intrusions based on device trustworthiness.

Key Features:

- Utilizes trust-based scoring metrics to classify IoT devices as trusted, suspicious, or malicious.
- Includes network behavior features such as authentication logs, access attempts, and historical interactions.
- Provides labeled attack types, including spoofing, replay attacks, and unauthorized access attempts.

Relevance to This Study:

- Helps in dynamic trust score computation for IoT nodes.
- Enhances adaptive security mechanisms based on real-time trust evaluation.
- Supports Zero Trust-based security models by ensuring continuous verification of device behavior.

Dataset Preprocessing and Feature Engineering

Before training the machine learning models, the datasets undergo the following preprocessing steps to ensure data quality and consistency:

1. Data Cleaning:
   - Removing duplicate records, missing values, and inconsistencies.
   - Filtering irrelevant traffic to focus on IoT-specific security threats.
2. Feature Selection & Extraction:
   - Identifying key network traffic features, such as packet frequency, protocol distribution, and anomaly scores.

- Extracting trust-based attributes from AutoTrust-IoTDS for dynamic trust scoring.

3. Data Normalization & Encoding:

   - Scaling numerical features using Min-Max normalization to improve model performance.
   - Encoding categorical variables (e.g., attack types) using one-hot encoding.

4. Data Splitting:

   - Dividing each dataset into training (70%), validation (15%), and testing (15%) sets.
   - Applying stratified sampling to maintain class balance.

3.2.1 DATA PREPROCESSING

Before training machine learning models, data preparation is an essential step in guaranteeing the dataset's consistency and quality. Good preprocessing improves model performance and generalizability by removing noise, superfluous data, and inconsistencies. This study preprocesses IoT security datasets using an organized methodology, emphasizing feature selection, data cleaning, addressing missing values, and normalization [7].

1. **Data Cleaning:**
   The raw datasets may contain duplicate, inconsistent, or incomplete records due to the nature of IoT network traffic logs and security event monitoring systems. Cleaning the dataset involves:

- Removing duplicate records that could bias the model. For example, duplicated network traffic flows might arise due to repeated log captures.
- Eliminating inconsistent or corrupted entries where data is unreadable or contains erroneous values (e.g., timestamps that do not align with expected patterns).
- Filtering out irrelevant features that do not contribute to trust evaluation (e.g., irrelevant metadata fields).

2. **Handling Null Values:**
   Missing values can negatively impact model training and prediction accuracy. To handle null values, different strategies are applied depending on the feature type:

- Numerical Features:

  - Median imputation is used, replacing missing values with the median of the respective column.

- Why median? Unlike the mean, the median is less sensitive to outliers in network security datasets.
- Example: If a feature "Packet Transmission Rate" has missing values, the median of all non-missing packet transmission rates will be used to fill them.

- Categorical Features:
  - Mode imputation (most frequent value) is applied.
  - This ensures that the most common category is used as a replacement without altering feature distribution.
  - Example: If an "IoT Device Type" field has missing values, the most frequently occurring device type (e.g., "Smart Camera") is used to fill them.

3. Normalization:

By guaranteeing that numerical features are scaled to a uniform range, normalization keeps models from being skewed toward higher values. Min-max scaling is used to rescale values between 0 and 1 since IoT security datasets include variables with varying scales (such as packet size, transmission frequency, and anomaly scores).

$$x_{norm} = \frac{x - min(X)}{max(X) - min(X)}$$

Where:

- x = original value
- $min(X)$ = minimum value of the feature
- $max(X)$ = maximum value of the feature
- x_{norm} = normalized value (ranges from 0 to 1)

4. Feature Selection:

Feature selection is crucial to enhance model interpretability, reduce computational complexity, and eliminate irrelevant variables. The goal is to select highly correlated features that significantly influence IoT trust scores.

- **Feature Importance Calculation**

  To determine the most relevant features, a trust-based feature significance score is computed using correlation analysis. The importance of each feature f(i) is defined as:

$$S(f_i) = \sum_{j=1}^{M} w_j \cdot Corr(f_i, T_j)$$

Where:

- $S(f_i)$ = significance score of feature f_i.
- M = total number of **trust parameters** (e.g., device authentication rate, anomaly score, access patterns).
- w_j = weight assigned to **trust parameter** T_j (weights reflect the relative importance of different trust factors).
- $Corr(f_i, T_j)$ = **correlation coefficient** between feature f_i and trust parameter T_j.

Feature Selection Process:

1. Calculate correlation scores:
   - Compute Pearson correlation coefficient between each feature and trust-related parameters.
   - Features with strong positive or negative correlations are prioritized.
2. Assign weights to trust parameters:
   - Assign different weights (wjw_jwj) based on their impact on security assessment.
   - Example: Anomaly detection score may have a higher weight than packet size, as anomalies are stronger indicators of attacks.
3. Select Top-N Features:
   - Features with the highest significance scores are retained for model training.
   - Features with low correlation or redundant information are removed.

3.3 ENSEMBLE LEARNING MODELS FOR IOT TRUST MANAGEMENT

To ensure robust and adaptive security in IoT networks, this study employs ensemble learning techniques to classify IoT nodes as trusted or untrusted based on their behavior, communication patterns, and security events. Ensemble learning improves model performance by combining multiple weaker models (base learners) to create a more accurate and reliable predictive model.

Two potent ensemble learning methods are used for IoT trust management:

- **XGBoost (Extreme Gradient Boosting) –** A high-performance tree-based model that uses gradient boosting to improve accuracy and reduce errors.
- **AdaBoost (Adaptive Boosting) –** A boosting algorithm that sequentially adjusts model weights to focus on hard-to-classify samples, enhancing overall security assessment.

These models analyze network traffic, authentication logs, anomaly detection scores, and device-specific attributes to classify IoT nodes as:

- **Trusted Nodes –** Devices exhibiting normal behavior.
- **Untrusted Nodes –** Devices showing signs of malicious activity, unauthorized access, or data anomalies.

The trust management framework delivers high detection accuracy, adaptability to changing threats, and improved interpretability for security specialists by combining the two models.

1. XGBoost (Extreme Gradient Boosting)

Gradient boosting is used by the tree-based ensemble learning algorithm XGBoost to maximize model performance. In complicated security situations, such identifying botnet assaults, illegal access attempts, and unusual device behavior, it works incredibly well.

Key Features of XGBoost:

- **Gradient Boosting Framework:** Sequentially improves weak learners (decision trees) by reducing errors.
- **Regularization Mechanism:** Prevents overfitting by using L1/L2 regularization (Ridge & Lasso techniques).
- **Feature Importance Evaluation:** Ranks security-related features to improve interpretability.

- **Handles Large-Scale Datasets:** Efficiently processes massive IoT security logs.

Fig 3. 3 XGBoost

2. AdaBoost (Adaptive Boosting)

A boosting technique called AdaBoost builds a powerful classifier by combining several weak classifiers. By dynamically modifying model weights to prioritize instances that are challenging to categorize, it improves IoT security.

Key Features of AdaBoost:

- **Adaptive Learning:** Gives more weight to **misclassified IoT nodes**, ensuring better learning.
- **Boosts Weak Learners:** Sequentially improves simple models (e.g., decision stumps).
- **Enhanced Interpretability:** Provides security analysts with transparent decision-making insights.

3.3.1 XGBOOST IN THE PROPOSED METHODOLOGY

XGBoost is an advanced "gradient boosting algorithm" designed for high accuracy and efficiency. It works by:

- Training a series of "decision trees sequentially", where each tree corrects errors from the previous one.
- Utilizing "regularization techniques" to prevent overfitting.
- Employing "parallelized learning" to handle large-scale IoT datasets efficiently.

Mathematical Representation of XGBoost:

By improving a loss function and reducing model complexity through regularization, XGBoost improves IoT security categorization. High precision, generalizability, and resilience to hostile attacks are thereby guaranteed.

1. Loss Function Optimization

The core objective of XGBoost is to minimize a loss function L(ϕ), which measures how well the model predicts IoT device behaviour. The general form of the loss function is:

$$L(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Where:

- $l(y_i, \hat{y}_i)$ – Measures the difference between the actual (y_i) and predicted ($\hat{y}_i$) values.
  - Example: **Log loss** for classification.
- $\Omega(f_k)$ – Regularization term that **prevents overfitting** and controls model complexity.
- n – Number of training samples.
- K – Total number of boosting trees.

2. Regularization Term (Model Complexity Penalty)

XGBoost controls model complexity using a regularization term:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda ||\omega||^2$$

Where:

- **T** – Number of leaves in a tree (larger trees increase complexity).
- **ω** – Scores assigned to leaves (weighting decision outcomes).
- **γ (Gamma)** – Controls how new leaves are added. Higher values prune unnecessary branches, reducing overfitting.
- **λ (Lambda)** – Controls L2 regularization, reducing large weights to improve generalization.

Implementation of XGBoost in IoT Trust Management

XGBoost is implemented in IoT trust management using two distinct approaches:

1. **Centralized IoT Trust Model** – A central server collects and processes security data from all IoT nodes, training a global trust model to classify devices.
2. **Decentralized IoT Trust Model** – Each IoT node computes its own trust score locally and shares trust insights with neighbouring devices, improving network-wide security.

1. Centralized IoT Trust Model

In the centralized approach, a centralized security server is responsible for:

- Collecting real-time IoT network traffic and device logs.
- Training an XGBoost-based trust model using aggregated security data.
- Classifying IoT nodes as trusted or untrusted based on their behavior.
- Broadcasting security policies and trust scores to all IoT devices.

Implementation Steps

- **Data Collection**
  - IoT nodes send network traffic logs, authentication data, and anomaly scores to the central server.
- **Feature Engineering**
  - Extract key security features (e.g., request frequency, authentication failures, anomaly scores).
  - Apply feature selection techniques to enhance model efficiency.
- **Model Training with XGBoost**
  - Train an XGBoost classification model using labelled IoT security datasets.
  - Optimize hyperparameters for better detection accuracy.
- **Trust Score Computation**
  - The trained model assigns a trust score to each IoT node.
  - Devices with low trust scores are flagged as untrusted.
- **Global Security Policy Enforcement**
  - The central server blocks or isolates malicious nodes.
  - Sends updated trust scores to all IoT devices.

Advantages

- **High accuracy** – Uses global data for precise anomaly detection.
- **Efficient central control** – Security policies are enforced uniformly.
- **Scalability** – Handles thousands of IoT devices in a single architecture.

2. Decentralized IoT Trust Model

In a decentralized paradigm, every IoT node calculates its trust score on its own and shares security intelligence with peers. Federated learning and blockchain-based security frameworks served as inspiration for this architecture.

Implementation Steps

- **Local Feature Extraction**
  - Each IoT device monitors its network behaviour and extracts security features.
- **Local XGBoost Model Training**
  - Devices train lightweight XGBoost models using local security logs.
  - Trust scores are calculated without sending raw data to a central server.
- **Collaborative Trust Sharing**
  - IoT nodes exchange trust scores with neighbours via peer-to-peer (P2P) communication.
  - A consensus mechanism (e.g., majority voting or blockchain-based validation) determines trusted/untrusted nodes.
- **Autonomous Security Enforcement**
  - Each node decides whether to trust or block other nodes based on received trust scores.

Advantages

- **No Single Point of Failure** – Each device makes independent security decisions.
- **Privacy-Preserving** – IoT devices do not share raw data, reducing hacking risks.
- **Lower Communication Overhead** – Trust decisions are made locally, reducing bandwidth usage.

Comparison of Centralized vs. Decentralized Models

Feature	Centralized Model	Decentralized Model
Data Collection	Collected at a central server	Each IoT device processes its own data
Trust Computation	Global trust model trained using all IoT data	Local trust models computed on devices
Communication Overhead	High (data constantly sent to server)	High (data constantly sent to server)
Security Risk	Single point of failure if the server is compromised	Distributed trust, no single failure point
Scalability	Scales well with cloud computing resources	Scales well in distributed networks
Privacy	Less privacy (all data is centralized)	More privacy (data stays on devices)

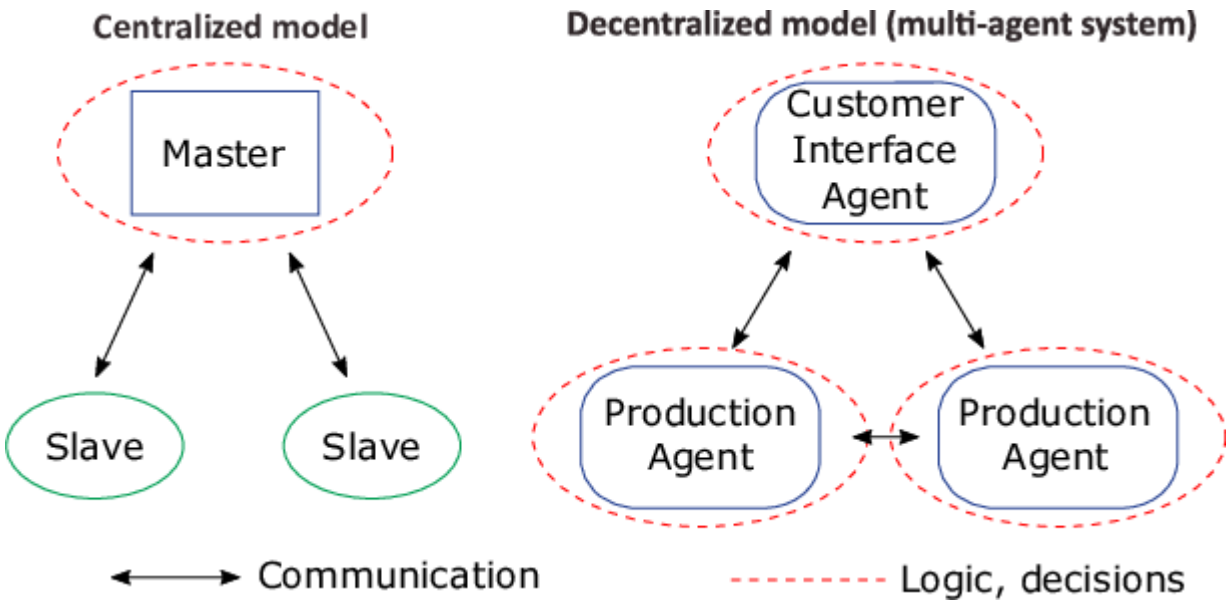

Fig 3. 4 XGBoost Centralized and Decentralized Model

3.3.2 ADABOOST IN THE PROPOSED METHODOLOGY

To increase classification accuracy, the 'adaptive boosting algorithm" AdaBoost combines several weak classifiers, such as decision stumps. To improve detection, it gives "higher weights to misclassified instances" in every iteration [8].

Mathematical Representation of AdaBoost:

By merging several weak classifiers into a single, powerful classifier, the ensemble learning technique known as AdaBoost (Adaptive Boosting) enhances IoT security categorization. By modifying instance weights, it reduces classification error and guarantees that misclassified IoT nodes are given greater attention in the following iteration.

1. Instance Weight Update in AdaBoost

In each iteration, AdaBoost updates the weight of each training instance based on its classification accuracy. The weight update formula is:

$$w_i^{(t+1)} = w_i^{(t)} \times e^{-\alpha_t y_i h_t(x_i)}$$

Where:

- $w_i^{(t)}$ – Weight of instance i at iteration t.
- α_t – Weight of the weak classifier $h_t(x)$, computed as:

  $$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$$

  where ϵ_t is the classification error rate of $h_t(x)$.
- y_i – True label of instance i (**+1 for trusted, -1 for untrusted**).
- $h_t(x_i)$ – Prediction by weak classifier h_t for input x_i (**+1 or -1**).
- $e^{-\alpha_t y_i h_t(x_i)}$ – Adjusts weight; misclassified instances get **higher weights** to be prioritized in the next iteration.

2. **Final Prediction in AdaBoost**

   Once all weak classifiers are trained, AdaBoost combines them to make the final prediction:

$$F(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

Where:

- $F(x)$ – Final classification score for IoT node x.
- T – Total number of weak classifiers.
- α_t – Importance weight of classifier $h_t(x)$.
- $h_t(x)$ – Output of weak classifier at iteration t (+1 or -1).

Implementation of AdaBoost in IoT Security

AdaBoost (Adaptive Boosting) is implemented in IoT security to enhance threat detection and trust evaluation by combining multiple weak classifiers into a strong ensemble model. It continuously improves classification performance by reweighting misclassified samples, making it effective for identifying malicious IoT nodes.

This section presents two approaches for AdaBoost-based IoT security:

1. **Centralized Model** – A global IoT security system where a central server trains a single AdaBoost model using data from multiple IoT devices.
2. **Decentralized Model** – Each IoT node locally evaluates its trust score and shares security insights within the network.

1. Centralized AdaBoost Model

In the centralized model, AdaBoost is used to analyze global IoT security data, improving threat detection accuracy. A central security server collects network traffic, authentication logs, and anomaly reports from IoT devices, then trains an AdaBoost model to classify IoT nodes as trusted or untrusted.

Implementation Steps

- **Data Collection & Preprocessing:** The central server aggregates security logs, authentication attempts, and network traffic data from IoT devices.
- **Feature Engineering:** Key features (packet behaviour, login anomalies, device interactions) are extracted.
- **Model Training:** The AdaBoost classifier is trained using labelled attack datasets (IoT-23, N-BaIoT, Edge-IIoTset).
- **Threat Detection & Mitigation:**
  - Malicious nodes are isolated from the network.
  - Access control rules are enforced based on trust scores.

Advantages of Centralized Model

- **High Accuracy** – Aggregates global IoT network data for better anomaly detection.
- **Efficient Training** – Uses powerful cloud resources for model training.
- **Simplified Deployment** – Can be integrated into cloud-based IoT security solutions.

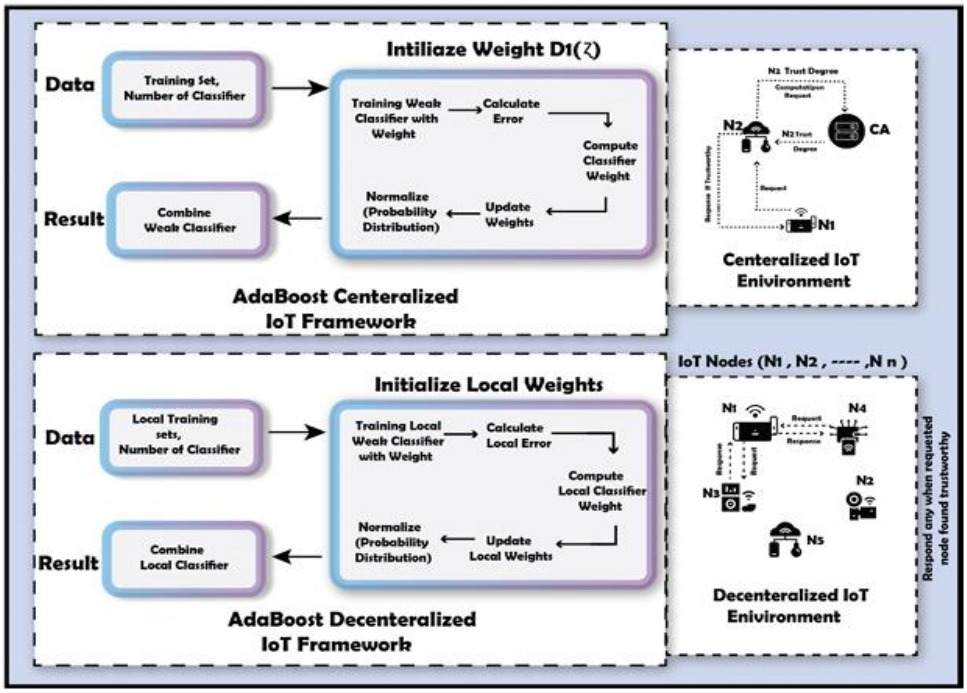

Fig 3. 5 The Implementation Workflow of AdaBoost in IoT Environment

2. Decentralized AdaBoost Model

IoT devices assess their own reliability and communicate their trust scores to nearby devices under the decentralized paradigm. As a result, the approach becomes less reliant on a single server, increasing its scalability and resilience to network outages [9].

Implementation Steps

- **Local Weight Initialization:** For each IoT node, initialize local weights for training instances.
- **Local Iterative Weak Classifier Training:**

  For each iteration:
  - Train a local weak classifier using the current weights.
  - Calculate the local error rate of the classifier.
- **Local Classifier Weight Calculation:** Compute the weight for the local classifier based on its accuracy.
- **Local Weight Update:** Update the local weights, focusing on misclassified instances.
- **Final Local Classifier Combination:** Combine all local weak classifiers to form a strong final classifier for each node.

Advantages of Decentralized Model

- **No Single Point of Failure** – Each IoT device evaluates security independently**.**
- **Low Latency & Bandwidth Usage** – Security decisions are made locally**.**
- **Scalability** – Works well in large-scale IoT networks (e.g., smart cities, autonomous systems**).**
- **Real-Time Threat Detection** – Each node adapts quickly to local threats**.**

CHAPTER 4

MERITS, DEMERITS AND APPLICATION

4.1 MERITS OF ENSEMBLE LEARNING IN IOT SECURITY

Ensemble learning techniques, such as XGBoost and AdaBoost, significantly enhance trust management in IoT networks by improving classification accuracy, scalability, adaptability, and threat detection [10].

1. High Accuracy

- Ensemble learning combines multiple weak classifiers to build a strong predictive model, which enhances the accuracy of detecting trusted and untrusted IoT devices.
- XGBoost uses gradient boosting, iteratively refining the model by minimizing errors.
- AdaBoost assigns higher weights to misclassified instances, ensuring that hard-to-classify data gets more focus in subsequent iterations.
- This results in a higher precision rate for detecting malicious IoT nodes.
    - **Example:** If an IoT node behaves suspiciously, XGBoost can identify it with high confidence by analyzing past behaviour trends.

2. Scalability

- Both models efficiently handle large-scale IoT networks where thousands or even millions of devices are connected.
- The boosting framework ensures efficient training without requiring an excessive amount of computational resources.
- Parallel computation in XGBoost speeds up the process, making it suitable for real-time IoT security applications.
    - Example: A smart city with millions of IoT sensors (traffic lights, CCTV cameras, pollution monitors) can rely on XGBoost/AdaBoost to scale trust assessment efficiently.

3. Adapts to Changes

- IoT environments are highly dynamic, with devices constantly joining or leaving the network.
- Ensemble models adapt to new behavioral patterns by continuously learning from new data.
- Concept drift handling ensures that the model stays relevant as threats evolve.

4. Better Threat Detection

- Combining multiple classifiers improves anomaly detection by reducing false positives and negatives.
- These models can detect botnet attacks, data breaches, unauthorized access, and DDoS threats more effectively than single classifiers.
- They provide real-time threat mitigation by flagging suspicious devices.

5. Combines Multiple Models for Better Performance

- By aggregating different models, ensemble learning benefits from each model's strengths while compensating for weaknesses.
- XGBoost is highly accurate and computationally efficient, whereas AdaBoost is adaptive and interpretable.
- Together, they provide robust, explainable, and precise trust management.

4.2 DEMERITS OF ENSEMBLE LEARNING IN IOT SECURITY

While ensemble learning techniques such as XGBoost and AdaBoost offer high accuracy and better threat detection, they come with certain challenges when applied in IoT security and trust management.

1. Complex to Implement

- Ensemble learning requires advanced knowledge of machine learning and optimization techniques.
- Setting up XGBoost or AdaBoost requires fine-tuning hyperparameters such as learning rate, depth of trees, and regularization terms to ensure the model performs optimally.

2. Needs More Resources (Computationally Expensive)

- XGBoost and AdaBoost require high computational power, especially for large-scale IoT networks.
- Tree-based models like XGBoost perform multiple iterations to refine the decision boundaries, consuming significant CPU and GPU resources.

3. Long Training Time

- Ensemble methods involve multiple iterations, where each classifier improves upon the errors of the previous one. This increases training time, especially for large datasets.
- Hyperparameter tuning (e.g., adjusting the number of weak learners, depth of trees, and learning rate) adds to the complexity and training duration.

4. Quality of Data Matters

- The performance of these models heavily depends on data quality—poor or noisy data can lead to inaccurate trust assessments.
- IoT data is often incomplete, imbalanced, or contains anomalies, making it difficult to train an effective ensemble model.

5. Less Interpretability

- Ensemble models, especially boosting techniques, act as "black boxes."
- Unlike decision trees or rule-based models, it is harder to understand why a specific device is classified as "trusted" or "malicious."

4.3 APPLICATION

- **Smart Home Security:** Protects smart home devices by ensuring only trusted devices can connect and communicate.
- **Industrial IoT Security:** Safeguards industrial equipment by detecting and stopping threats from compromised devices.
- **Healthcare Devices:** Secures medical devices that collect patient data, ensuring only trusted devices share sensitive information.
- **Autonomous Vehicles:** Ensures safe communication between self-driving cars and their surroundings to prevent cyberattacks.
- **Supply Chain Security:** Secures IoT devices used in tracking inventory, making sure only trusted devices can access data.
- **Smart Agriculture:** Secures devices used in farming, like soil sensors, to prevent unauthorized access and ensure data accuracy.

CHAPTER 5

SIMULATION AND RESULTS

Jupyter Notebook was used for both the training and testing stages of the experimental simulation and results of the suggested trust management mechanism in IoT contexts. Using the XGBoost and AdaBoost algorithms, the study assessed both centralized and decentralized trust implementations and contrasted them with pre-existing models.

1. Deployment and Integration in IoT Systems

- The IoT network consisted of 450 nodes (IoT devices).
- **Centralized Approach:** A central node collected data from all IoT nodes and applied XGBoost and AdaBoost models to compute trust scores.
- **Decentralized Approach:** Each IoT node computed its trust score independently using ensemble models.
- **Key evaluation metrics:** Detection accuracy, false positive rates, system latency, and scalability.

2. Performance Metrics and Results

- **AdaBoost Model Performance:**
    - Accuracy, precision, recall, and F1-score: 0.99 for both classes (trusted and untrusted IoT nodes).
    - Macro and weighted averages: 0.99, confirming its ability to consistently classify IoT nodes.
- **XGBoost Model Performance:**
    - Precision scores: 0.98 (trusted) and 1.00 (untrusted).
    - Recall scores: 1.00 (trusted) and 0.98 (untrusted).
    - F1-score: 0.99 for both classes, confirming the model's ability to accurately classify IoT nodes.
- The models showed high accuracy and robust classification performance, outperforming existing models.

3. Detection Rate Evaluation

- **Decentralized Approach:**
  - XGBoost detection rate: 99% of malicious nodes.
  - AdaBoost detection rate: 98.6%, nearly as effective as XGBoost.
  - Comparative models performed lower: SELM (96.5%), ETSM (97.2%), ELTB (95.8%), BTAMC (96.3%), RepuTE (97.5%).
- **Centralized Approach:**
  - XGBoost detection rate: 99.5%.
  - AdaBoost detection rate: 99.1%.
  - Outperformed existing methods, which had detection rates ranging from 96.1% to 97.8%.

4. Performance Against IoT Security Attacks

- **On-Off Attack:**
  - XGBoost: 95.4% detection rate.
  - AdaBoost: 94.7% detection rate.
- **Good Mouthing Attack (malicious nodes rate other bad nodes positively):**
  - XGBoost: 96.8% detection rate.
  - AdaBoost: 96.1% detection rate.
- **Bad Mouthing Attack (malicious nodes give negative feedback to good nodes):**
  - XGBoost: 97.5% detection rate.
  - AdaBoost: 96.9% detection rate.
- **White Washing Attack (attackers reset their bad history by rejoining the network):**
  - XGBoost: 98.3% detection rate.
  - AdaBoost: 97.7% detection rate.

- **Comparison with existing models:**
    - Other models had lower detection rates, ranging from 85.5% to 94.6%, confirming that XGBoost and AdaBoost outperform traditional IoT trust management models.

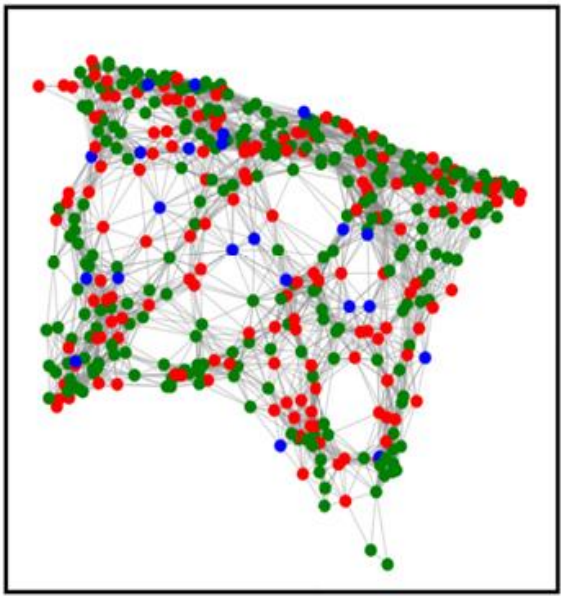

Fig 5. 1 Centralized implementation of the IoT Network

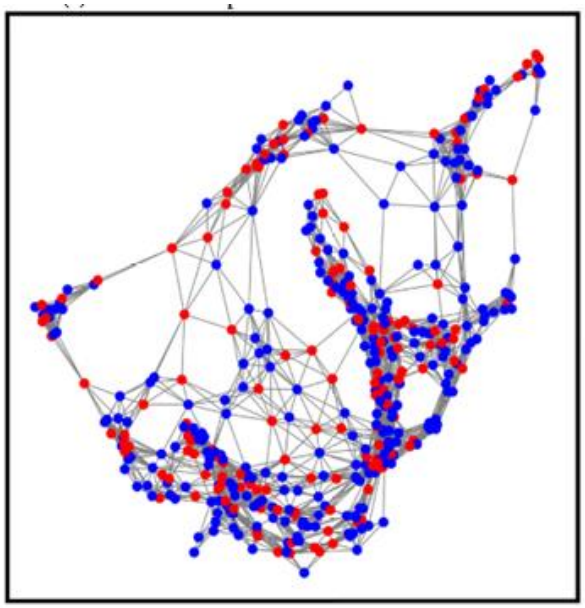

Fig 5. 2 Decentralized implementation of the IoT Network

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

This study's research focuses on using advanced ensemble learning techniques, specifically XGBoost and AdaBoost, to improve trust management in IoT contexts. These models were created to guarantee high security, effectiveness, and scalability while identifying, stopping, and mitigating hostile activity in IoT networks.

The rapid proliferation of connected devices and the growing complexity of IoT systems make it difficult for traditional security measures to maintain strong trust management. By introducing a Zero Trust architecture, the suggested method makes sure that no device is automatically trusted and that each transaction and communication is verified. This study greatly enhances threat detection, trust assessment, and decision-making in IoT networks by utilizing ensemble learning.

6.2 FUTURE SCOPE

By incorporating blockchain technology, the suggested trust management architecture for IoT security can be further improved by creating a decentralized, impenetrable trust mechanism that guarantees openness and resistance to malevolent changes. Federated learning can also be used to provide distributed trust computation among IoT nodes, which can lessen reliance on centralized systems while maintaining data privacy. The ability of AI-driven processes to continuously adapt to new security risks without the need for regular manual updates is another noteworthy improvement in real-time adaptive learning of trust models. Additionally, the framework is appropriate for low-power edge computing environments by optimizing the computational efficiency of the XGBoost and AdaBoost models, which can enable smooth deployment on resource-constrained IoT devices.

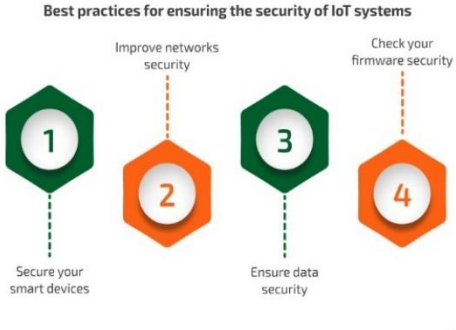

Fig 6. 1 Best practices for ensuring the security of IoT systems

REFERENCES

[1] M. Q. Husamuddin Mohammed, “ Internet of Things :A Study on Security and Privacy Threats,” vol. 2, no. 3, pp. 1-6, March 2017.

[2] D. N. K. D. R. C. Happy, “Security Issues in IoT Applications,” *Journal of Information Systems Engineering and Management,* vol. 3, no. 1, pp. 1-20, 05 Feb 2025 .

[3] O. R. O. A. J. B. Akinrotimi Akinyemi Omololu, “A Comparative Analysis of Adaboost and XGBoost Meta-Algorthms,” vol. 1, pp. 1-10, December 2024.

[4] S. U. R. a. I. U. Hanan Aldowah1, “Security in Internet of Things: Issues,,” vol. 2, no. 1, pp. 395-405, July 2019.

[5] L. E. Peterson, “Optimization of classifier ensemble diversity,” *ACADEMIA MOLECULAR BIOLOGYAND GENOMICS,* vol. 3, no. 1, pp. 1-8, 05 August 2024.

[6] M. Ţălu, “Security and privacy in the IIoT: threats, possible security countermeasures, and future challenges,” *Computing&AI Connect* , vol. 1, pp. 1-12, October 03, 2024 .

[7] T. Adewale, “ Enhancing Security in IIoT Networks: Comparative Analysis of Deep Learning,” vol. 1, August, 2024.

[8] T. O. K. R. Gunnar Ratsch, “ Regularizing AdaBoost,” vol. 2, pp. 4-8, 1999.

[9] L. Kuncheva, “ An Investigation into How ADABOOST Affects Classifier Diversity,” pp. 2-7, 2001.

[10] Z. W. P. K. S. F. A. A. a. J. W. Yongjun Ren, “ Zero Trust Networks: Evolution and Application from Concept to Practice,” pp. 1-21, 02 January 2025.