PeerJ Computer Science

Hybrid computing framework security in dynamic offloading for IoT-enabled smart home system

Sheharyar Khan[1], Zheng Jiangbin[1], Farhan Ullah[1], Muhammad Pervez Akhter[2], Sohrab Khan[3], Fuad A. Awwad[4] and Emad A.A. Ismail[4]

[1] School of Software, Northwestern Polytechnical University, Xi'an, China
[2] Department of Computer Science, National University of Modern Languages, Faisalabad, Pakistan
[3] Elementary and Secondary Education Department, Government of Khyber Pakhtunkhwa, Haripur, Pakistan
[4] Department of Quantitative Analysis, College of Business Administration, King Saud University, Riyadh, Saudi Arabia

ABSTRACT

In the distributed computing era, cloud computing has completely changed organizational operations by facilitating simple access to resources. However, the rapid development of the IoT has led to collaborative computing, which raises scalability and security challenges. To fully realize the potential of the Internet of Things (IoT) in smart home technologies, there is still a need for strong data security solutions, which are essential in dynamic offloading in conjunction with edge, fog, and cloud computing. This research on smart home challenges covers in-depth examinations of data security, privacy, processing speed, storage capacity restrictions, and analytics inside networked IoT devices. We introduce the Trusted IoT Big Data Analytics (TIBDA) framework as a comprehensive solution to reshape smart living. Our primary focus is mitigating pervasive data security and privacy issues. TIBDA incorporates robust trust mechanisms, prioritizing data privacy and reliability for secure processing and user information confidentiality within the smart home environment. We achieve this by employing a hybrid cryptosystem that combines Elliptic Curve Cryptography (ECC), Post Quantum Cryptography (PQC), and Blockchain technology (BCT) to protect user privacy and confidentiality. Additionally, we comprehensively compared four prominent Artificial Intelligence anomaly detection algorithms (Isolation Forest, Local Outlier Factor, One-Class SVM, and Elliptic Envelope). We utilized machine learning classification algorithms (random forest, k-nearest neighbors, support vector machines, linear discriminant analysis, and quadratic discriminant analysis) for detecting malicious and non-malicious activities in smart home systems. Furthermore, the main part of the research is with the help of an artificial neural network (ANN) dynamic algorithm; the TIBDA framework designs a hybrid computing system that integrates edge, fog, and cloud architecture and efficiently supports numerous users while processing data from IoT devices in real-time. The analysis shows that TIBDA outperforms these systems significantly across various metrics. In terms of response time, TIBDA demonstrated a reduction of 10–20% compared to the other systems under varying user loads, device counts, and transaction volumes. Regarding security, TIBDA's AUC values were consistently higher by 5–15%, indicating superior protection against threats. Additionally, TIBDA exhibited the highest trustworthiness with an uptime percentage 10–12% greater than its competitors. TIBDA's Isolation Forest

Submitted 24 January 2024
Accepted 30 June 2024
Published 23 August 2024

Corresponding author
Sheharyar Khan,
shksherry@mail.nwpu.edu.cn,
shksherry@gmail.com

Academic editor
Vicente Alarcon-Aquino

Additional Information and Declarations can be found on page 56

DOI **10.7717/peerj-cs.2211**

How to cite this article Khan S, Jiangbin Z, Ullah F, Pervez Akhter M, Khan S, Awwad FA, Ismail EAA. 2024. Hybrid computing framework security in dynamic offloading for IoT-enabled smart home system. *PeerJ Comput. Sci.* **10**:e2211 http://doi.org/10.7717/peerj-cs.2211

algorithm achieved an accuracy of 99.30%, and the random forest algorithm achieved an accuracy of 94.70%, outperforming other methods by 8–11%. Furthermore, our ANN-based offloading decision-making model achieved a validation accuracy of 99% and reduced loss to 0.11, demonstrating significant improvements in resource utilization and system performance.

Subjects Computer Networks and Communications, Cryptography, Data Mining and Machine Learning, Security and Privacy, Internet of Things
Keywords Smart home, Internet of Things (IoT), Big data, Offloading, Artificial intelligence, Machine learning, Data security and privacy, Hybrid computing, Cryptography, Blockchain

INTRODUCTION

The Internet of Things (IoT) has increased into numerous services and applications that permeate our daily activities. This expansion has brought innumerable benefits and added value to human society (*Khan et al., 2022*). IoT-based smart home systems are one of the best examples of the epitome of future living through IoT integration. Communities worldwide actively embrace this paradigm as a core component of modernization endeavours (*Wang et al., 2013*). The fusion of wireless sensor network technologies (*Hsu et al., 2017*) with the IoT heralds a transformative era of global interconnectivity, uniting many smart devices boasting cutting-edge functionalities (*Katuk et al., 2018*; *Froiz-Míguez et al., 2018*). The wireless home automation network is at the core of this technological shift, a dynamic system of sensors and actuators that work together, sharing resources and creating connections. This network is a critical technology that will enable the development of competent smart homes.

Moreover, the dependence on network and wireless communication for data exchange and remote control functionalities introduces additional layers of complexity and vulnerability (*Xu et al., 2019*). The seamless operation of these systems relies on stable and secure connectivity, which network outages or cyber threats can disrupt. To address these concerns, research focuses on enhancing home automation networks' resilience and security (*Abu-Tair et al., 2020*). This includes developing robust encryption protocols and implementing redundant communication pathways to maintain functionality during disruptions. Furthermore, as smart homes generate vast amounts of data, advanced data analytics and machine learning algorithms optimize system performance and predict maintenance needs, preempting potential failures (*Nguyen et al., 2021*).

In a smart home, data from temperature and smoke detection sensors for real-time fire detection and electricity and gas consumption to efficiently manage the homes' power, gas, and water use is sent to a central monitoring station for continuous home monitoring. Within these constantly connected residences, a wealth of valuable data is continually generated by many smart devices and appliances integrated into the Internet of Things (IoT) ecosystem (*Signoretti et al., 2021*). IoT devices enable remote control of appliances, energy management, and enhanced security systems, providing convenience and energy savings (*El-Sayed et al., 2017*).

Privacy has become crucial in the rapidly changing IoT ecosystem (*Mocrii, Chen & Musilek, 2018*; *Haney, Furman & Acar, 2020*; *Edu, Such & Suarez-Tangil, 2020*). However, because IoT-based tools and devices are primarily meant to gather data about users' behaviours, vital signs, surrounding environments, and more, they are tempting targets for hackers and present several security and privacy risks (*Geneiatakis et al., 2017*). Because IoT devices have limited energy, storage, communication, and processing power, traditional security and privacy methods are poorly designed (*Haney, Acar & Furman, 2021*). Motivated by this, researchers have been forced to develop creative fixes and algorithms to deal with these limitations.

In parallel with these technological advancements, the ethical implications of data privacy and security in smart home environments are gaining attention. The continuous collection and transmission of personal data through IoT devices pose significant risks if not adequately protected (*Bajaj, Sharma & Singh, 2022*). Researchers advocate for adopting privacy-by-design principles, ensuring that data protection measures are integrated into developing home automation technologies from the outset (*Shouran, Ashari & Priyambodo, 2019*). This includes anonymizing data, securing data storage, and giving users transparent control over their information. As smart homes become more prevalent, addressing these privacy concerns is crucial to building trust and ensuring the widespread acceptance of these technologies.

Cloud computing has enabled many innovative operations (*Padhy, Patra & Satapathy, 2011*). Yet, these frequently fail to consider security and privacy issues that could affect these cloud-enabled services and apps. Because of these difficulties, researchers have been forced to come up with novel approaches and paradigms that attempt to provide safe and private mechanisms and services while taking into account the resource limitations of IoT devices and the open nature of cloud computing-based services (*Krishna et al., 2016*). In addition, more processing and analysis are required for the data gathered in particular applications to provide the desired services.

The need for processing power is very high. Thus, to satisfy the processing and storage needs of users and applications, it is imperative to integrate IoT peripherals with cloud service providers. However, there are several difficulties with this kind of integration, especially in terms of security and privacy. Cloud service providers entirely own and control users' data, which can track their actions and activities and identify the IoT devices they use and their types, usage patterns, access hours, and recording frequencies. This invasion of privacy has the potential to make users transparent. Therefore, setting up procedures and systems that anonymize and secure user data is critical to preserve their security and privacy. Additionally, because of their mobility and limited energy sources, IoT devices have limited Internet connectivity, which makes short-range communication protocols like WiFi, Bluetooth, and ZigBee increasingly popular (*Bulgurcu, Cavusoglu & Benbasat, 2010*).

In many scenarios, edge and fog computing has been added as a middle layer between the cloud and IoT layers (*Achar, 2022*). This strategy is based on the fact that these devices have much more computing, storage, and energy capacity than IoT devices. As a result, they can use more sophisticated technologies that enable longer distances,

such as LoRaWAN, LTE, and LTE-M, and offer long-range communication features to the Internet (*Raghunath & Rengarajan, 2019*). These devices also provide low-power, short-range wireless communication to other Internet of Things devices. Additionally, they usually have higher computing and storage power than IoT devices, which allows them to run programs that need more resources, like algorithms and functions related to security and privacy. As a result, edge and fog computing are trusted entry points to the cloud and are directly connected to IoT devices.

Research motivation

Distributed computing and the IoT have combined to create shared platforms that offer simultaneous data and effortless access to services. The Internet of Things has made it possible to create shared platforms by integrating online computing capabilities, giving users instant access to their data and resources from anywhere. When IoT devices are integrated into shared systems, new security risks surface; hence, system design security must be guaranteed. Given that several users will be accessing and using comparable data and capabilities, this is crucial.

For multiple-user-friendly applications, the integration of IoT inside a secure computing environment is extremely important. A secure computing system is required to host and manage IoT devices and data safely and effectively. To reduce possible threats to data integrity and privacy, the architecture should incorporate a number of security measures, including threat detection, access control, authorization, and authentication. When designing a system that includes several users, the secure framework for computing must consider users with different rights and capabilities. Therefore, an efficient and scalable access control system must be implemented to manage user rights and restrict data access to authorized parties while maintaining data integrity and privacy. The massive volumes of data produced by Internet of Things devices necessitate a safe computing architecture. Scalable storage solutions that can manage massive data volumes without compromising security and accessibility are essential.

The goal of designing a secure hybrid computing platform with IoT for multi-user systems is general and presents opportunities for research in several fields of study. The proposed study's goal is to develop a secure, scalable, and efficient dynamic offloading computing architecture. Massive data volumes and multi-user systems should be handled without sacrificing network performance or security. The primary goal of the study's first phase is to design a computing architecture that is reliable, scalable, and capable of processing data from Internet of Things devices. Important design goals include scalability, efficiency, and the ability to handle massive amounts of data.

Additionally, the proposed framework should be flexible in order to support Internet of Things devices that gather data from several sensing sources. The gathered data can be safeguarded and kept secure during the integration process. Moreover, the architecture uses cryptographic approaches to guarantee the security and privacy of user data. Post-quantum cryptography (PQC) and blockchain technology are combined in a combination cryptosystem. The integrated PQC-blockchain technology uses PQC encryption to protect data from serious threats. This encryption technique helps in preventing the compromise

of sensitive data. The method uses blockchain technology to make sure that immutable, distributed records are used to store sensitive data. Data in the hybrid PQC-blockchain system is securely and reliably examined before being integrated into the blockchain.

This secure computing architecture should be developed by integrating IoT and the hybrid PQC-blockchain system, which allows for the efficient administration of numerous users with fine-grained access permissions; additionally, by ensuring that data is only available to authorized individuals, a strong and scalable access control approach may be implemented, reducing a variety of security threats. Designed distributed systems, which utilize a PQC-blockchain hybrid system, provide users immediate access to data and services and protect their infrastructure from a wide range of threats.

Research objectives

To follow the motivation of our research, we proposed a framework named Trusted IoT Big Data Analytics (TIBDA), which aims to create a scalable, safe, and effective hybrid computing architecture that can manage massive amounts of data in a smart home system and enable multi-user systems while simultaneously maintaining network performance and security. Using a hybrid cryptosystem that integrates Elliptic Curve Cryptography (ECC), Post-Quantum Cryptography (PQC), machine learning algorithms like random forest and isolation forest, and blockchain technology, a primary objective is to implement cryptographic techniques to ensure user information privacy and confidentiality (*Zhang et al., 2023*).

Sensitive data is encrypted and decrypted by ECC and PQC to guarantee its integrity and confidentiality (*Nouioua & Belbachir, 2023*). When detecting anomalies during data transmission, the random forest and Isolation Forest algorithms can be used to ensure that encrypted data is safe from unwanted access. Utilizing blockchain technology preserves private information in distributed, unchangeable records (*Abbas et al., 2021*). Through consensus processes and mining techniques, data is safely and reliably examined before being included in the blockchain within the hybrid computing system. By integrating ECC-PQC with blockchain technology and machine learning techniques like random forest and Isolation Forest in a hybrid computing environment, we may be able to create a comprehensive security framework that addresses various security concerns and threats. This technique combines advanced anomaly detection and encryption techniques with blockchain-based immutability to safeguard sensitive data and ensure the integrity of the computing environment. A more linked and productive world is encouraged by the increasing use of IoT devices inside common infrastructures (*Azzaoui, Sharma & Park, 2022*).

Furthermore, with the help of an artificial neural network (ANN) dynamic algorithm (*Khan et al., 2024*), the TIBDA framework designs a hybrid computing system that integrates edge, fog, and cloud architecture and efficiently supports numerous users while processing data from IoT devices in real-time. Scalability, efficiency, and the capacity to manage enormous volumes of data are important design goals. Additionally, the architecture is flexible enough to accommodate IoT devices that gather data from different

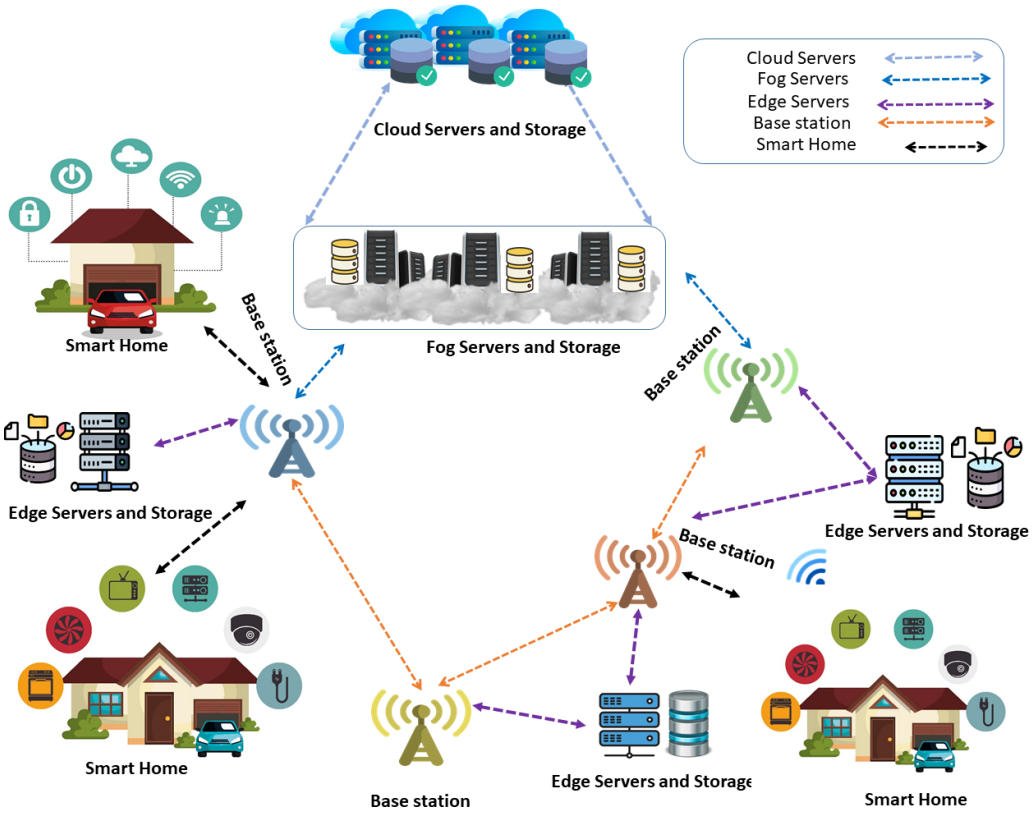

Figure 1 **Smart home devices connect to edge, fog, and cloud servers *via* a base station for data processing and analytics. Edge servers are preferred if nearby; otherwise, fog servers are used. For critical processing, requests are forwarded to cloud servers.** Image source credits: Smart home technology set icons, FreePik, https://www.freepik.com; Cloud server, Database free icons, Wifi, FlatIcon, https://www.flaticon.com; Cloud server, https://freepngimg.com/png/11351-cloud-server-png-file#google_vignette, CC BY-NC 4.0; Vector Server Icon, server icons PNG Designed By EncoderXSolutions from https://pngtree.com/freepng/vector-server-icon_4973694.html?sol=downref&id=bef; Database, https://icon-icons.com/icon/database-data/19664, CC BY-NC 4.0.

Full-size DOI: 10.7717/peerjcs.2211/fig-1

sensing sources (*Kuldeep & Zhang, 2022*), and the data is safely saved and safeguarded throughout the integration process.

In a world where these technologies' personalized and safe use is critical, our method guarantees that homeowners can enjoy a level of convenience and safety appropriate to their requirements. The results of our work confirm the proposed approach's improved cost-effectiveness and added convenience while highlighting its potential to influence smart living in the future. The basic architecture of the IoT-enabled smart homes systems process with hybrid computing (Edge, Fog, Cloud) is shown in Fig. 1.

Research contributions

- Firstly, we developed an IoT-enabled smart home system using a hybrid computing framework (edge-fog and cloud) with IoT devices, sensors, and appliances. For security purposes, our study presents an innovative, comprehensive framework for protecting hybrid computing for IoT data in smart homes. We achieve this by employing a

hybrid cryptosystem that combines ECC, PQC, and BCT to protect user privacy and confidentiality. By incorporating trust mechanisms, this innovative solution ensures secure processing and confidentiality of user information while addressing data privacy and reliability.

- To detect the anomaly in the smart home systems, We comprehensively compared four prominent Artificial Intelligence anomaly detection algorithms: Isolation Forest, Local Outlier Factor, One-Class SVM, and Elliptic Envelope. Our research findings revealed that Isolation Forest outperformed other anomaly detection algorithms, which are presented in detail.
- Additionally, in our study, we have utilized Machine Learning classification algorithms such as random forest (RF), k-nearest neighbors (KNN), support vector machines (SVM), linear disriminantc analysis (LDA), and quadratic discriminant analysis (QDA) for detecting malicious and non-malicious activities. Based on the deep analysis, random forest performed better in detecting malicious and non-malicious activities, as presented in our study.
- Finally, our framework incorporates an artificial neural network (ANN) dynamic algorithm. This pioneering optimization technique dynamically allocates computational tasks among Edge, Fog, and Cloud resources in response to real-time conditions. This innovative feature guarantees operational efficiency, responsiveness, and resource utilization, thereby greatly enhancing the framework's smart home data processing capabilities and overall system performance.

Orgnaization

The rest of the article is structured as follows: 'Related Work' summarizes related works. 'Proposed Methodology' presents our proposed approach, problem formulation, and algorithms. In 'Results and Discussion', we present the obtained results and provide a discussion based on the results and significance of our proposed model. Finally, 'Conclusion & Future Work' discusses future works and concludes the article.

RELATED WORK

Some of the significant previous research studies on the same topic are included in this section. In Table 1, we have summarized previous approaches, references, methodology, contributions, and limitations.

To solve the current issues with authentication in cloud-hosted Internet of Things systems, *Irshad & Chaudhry (2021)* presented SAS-Cloud, a novel authentication technique built on the ElGamal framework. This method combined biometric information and user passcodes for identity verification. The authors carefully evaluated SAS-Cloud's security and effectiveness, demonstrating its resilience to possible intrusions and greater effectiveness compared to other options then on the market. By combining biometric characteristics and passcodes, the authentication approach provided increased security. The study emphasized how important secure authentication was for cloud-based Internet of Things applications and presented SAS-Cloud as a ground-breaking way to meet this crucial requirement.

Table 1 Summary of previous approaches: methodologies, contributions, and limitations.

Reference	Methodology	Contribution	Limitation
Irshad & Chaudhry (2021)	SAS-cloud authentication method	Integrating passcodes and biometric data offers enhanced security and efficiency compared to existing methods.	The study's findings lack extensive real-world testing and deployment.
Sharma et al. (2015)	SHCEF framework	The potential of IoT-cloud integration proposes a novel framework, SHCEF, to address security challenges and improve the efficiency of IoT deployments.	Further research and real-world testing are necessary to evaluate the practical effectiveness and scalability of the SHCEF framework in diverse IoT environments.
Jalasri & Lakshmanan (2023)	Clustering algorithm and cryptography method	The research contributes to enhancing data security in fog systems by proposing a clustering algorithm and cryptography method.	The study needs further validation through empirical testing and evaluation in real-world distributed environments to assess the scalability and effectiveness.
Unal et al. (2021)	Safe cloud storage system (SCSS)	The research advances cloud storage security by introducing the SCSS, which integrates IBC and decentralized key administration. The system offers improved protection by addressing scalability issues and enhancing security through multiple PKGs.	Although the SCSS architecture shows promise, its limitations may include the need for further validation through real-world deployment and testing to assess its scalability, performance, and resilience against potential security threats in diverse cloud environments.
Selvarajan et al. (2023)	AILBSM model	The research improves the privacy and security of Industrial IoT systems by introducing the AILBSM model, which blends COSNN with blockchain authentication.	The AILBSM model may have limitations, such as the need for further evaluation in real-world industrial IoT environments to assess its scalability and robustness.
Uppuluri & Lakshmeeswari (2023)	MHE-IS-CPMT protocol	By implementing the MHE-IS-CPMT protocol with ECC for identification and key exchange, the research improves security in residential frameworks.	Limitations may include evaluating its scalability, efficiency, and resilience against potential security threats in diverse home environments.
Ahmad, Mehfuz & Beg (2023)	Hybrid cryptographic Methodology	The research enhances security in cloud environments by proposing a hybrid cryptographic methodology that combines ECC and AES for KAS.	Need further evaluation in diverse cloud scenarios to assess its scalability, adaptability, and resilience against potential security threats.
Sharma et al. (2023a)	Proposed application with Blockchain	The research enhances security and efficiency in healthcare certificate management by implementing a blockchain-based Proposed Application (PA).	Additionally, research needs regulatory compliance and privacy protection considerations should be addressed for broader adoption in the healthcare industry.
Proposed approach	**TIBDA framework**	The research investigates challenges in smart home technology, specifically, the concerns related to data security, privacy, processing efficiency, storage limitations, and interconnected Internet of Things (IoT) devices.	Further study is needed to integrate auto-scaling characteristics with security algorithms. It requires comprehensive testing across multiple datasets to ensure a reliable, secure architecture.

Sharma et al. (2015) discussed the evolution of communication technologies, from digitization and the Internet to pervasive computing and the IoT. They emphasized the advantages of integrating IoT with Cloud Computing to handle vast amounts of data from diverse devices. However, this integration presented security challenges. To address these challenges, they proposed a Secure, Hybrid, Cloud-Enabled architecture for IoT (SHCEF), leveraging both public and private clouds. This architecture aimed to ensure data security within domains while tackling scalability and interoperability issues. Additionally, the paper outlined research challenges in implementing this combined Cloud-IoT architecture.

Jalasri & Lakshmanan (2023) addressed data security challenges from storing sensitive information on third-party cloud servers, particularly data collected through IoT devices. Integration of IoT with fog computing aimed to manage data security, latency, and privacy issues. To enhance security in this distributed environment, they proposed a cryptographic algorithm combined with a clustering algorithm. Cluster heads were identified based on power probabilistic criteria, and data transmission was optimized through clustering in fog systems. Data security was ensured using the noise protocol framework encryption process, employing various cryptographic functions. A comparison with the energy-efficient Heterogeneous Clustering Algorithm (EEHCA) demonstrated the effectiveness of the proposed approach in minimizing intermediate attacks and reducing node energy consumption during data transmission.

Unal et al. (2021) addressed the challenge of data security in cloud services, where end-users lack control over their data once transmitted to the cloud. Conventional PKI-based solutions were inadequate for large-scale cloud systems, necessitating efficient, scalable, and secure key management. Key requirements included scalable encryption, authentication, non-repudiation services, data sharing, and forensic investigation support. Existing solutions lacked secure Type-3 pairings, Encryption-as-a-Service (EaaS), and multiple Public Key Generators (PKGs). To address these gaps, they proposed a Secure Cloud Storage System (SCSS) based on efficient identity-based cryptography (IBC). SCSS supported distributed key management, encryption mechanisms, and multiple PKGs. During forensic investigations, legal authorities could utilize the multiple PKG mechanism for data access, while an account locking mechanism prevented single authority access. Performance evaluations demonstrated SCSS's superiority in scalability and efficiency for encryption and decryption operations, streamlining forensic analysis on encrypted cloud data.

The Industrial Internet of Things (IIoT) held immense promise for revolutionizing business operations across diverse sectors, yet traditional architectures faced significant security vulnerabilities. *Selvarajan et al. (2023)* introduced an Artificial Intelligence-based Lightweight Blockchain Security Model (AILBSM) to bolster the privacy and security of IIoT systems. By leveraging lightweight blockchain and Convivial Optimized Sprinter Neural Network (COSNN) AI mechanisms, their model enhanced security operations while mitigating the impact of attacks. Central to its effectiveness was the use of Authentic Intrinsic Analysis (AIA) to encode features, reducing vulnerability to attacks. Extensive experimentation validated the system's efficacy, showcasing improved execution time,

classification accuracy, and detection performance. Notably, the inclusion of auto-encoder-based transformation and blockchain authentication significantly enhanced anomaly detection compared to existing techniques.

The emergence of IoT-based applications necessitated robust communication protocols to ensure secure interactions among smart home devices. Addressing the critical concern of authentication and access control, *Uppuluri & Lakshmeeswari (2023)* proposed a novel protocol named Modified Honey Encryption using Inverse Sampling-Conditional Probability Model Transform (MHE-IS-CPMT) with ECC. The protocol comprised five main steps: initialization, registration, login and data access request, authentication and session key agreement, and key update. During initialization, users and devices were initialized at the home network head (H). Registration involved users and devices registering with H *via* the smart gateway (SG), ensuring secure transmission using MHE-IS-CPMT with ECC. In the login process, registered users connected to the smart home system requested device access *via* SG and underwent authentication, receiving device access control *via* the private key. The system also supported secure key updates for users as needed. By leveraging ECC and MHE-IS-CPMT, their protocol enhanced security against various attacks, offering superior protection compared to existing methods.

Ahmad, Mehfuz & Beg (2023) research work was on the significance of secure data management in cloud computing environments, and they mainly emphasized the role of Key Management Systems (KMS) in ensuring data security across various domains, including e-healthcare, information security, and large-scale organizations. By utilizing a secure key generation approach involving random prime numbers and master secret keys, along with robust encryption techniques such as ECC and Advanced Encryption Standard (AES), their research study aimed to enhance data security and authentication processes. Their proposed Hybrid Cryptographic Approach for Key Management System (HCA-KMS) demonstrated improved efficiency compared to existing methods, addressing security vulnerabilities and reducing computational complexity. Through comparative analysis, the efficacy of HCA-KMS was evaluated in terms of confidentiality, integrity, time complexity, storage overhead, resource utilization, and security, showcasing its potential for enhancing data security in cloud environments.

Sharma et al. (2023a) research work explored the integration of blockchain technology into healthcare systems with an aim to improve security, privacy, and efficiency. They introduced a secure blockchain-based proposed application (PA) designed to generate and verify medical certificates, ensuring confidentiality, authentication, and access control through smart contracts. Through comparative analysis, their study highlighted the superiority of the proposed solution over existing schemes, providing a robust framework for managing healthcare data securely and efficiently.

Comparative analysis of proposed and existing approaches

In the rapidly evolving landscape of smart home technology, ensuring robust data security, privacy, and efficient processing capabilities has become paramount. This comparative analysis aims to assess the efficacy of existing approaches in addressing these challenges and

introduce our proposed model, the Trusted IoT Big Data Analytics (TIBDA) framework, as a superior solution.

Existing approaches

The comparative analysis of our proposed model with existing studies reveals the evolving landscape of distributed computing and the persistent IoT challenges that researchers aim to address. Previous research has explored various aspects of data security, privacy, processing efficiency, and storage limitations within the domains of distributed computing and the IoT. For instance, studies such as *Irshad & Chaudhry (2021)* presented authentication techniques like SAS-Cloud, emphasizing security in cloud-hosted IoT systems, while *Sharma et al. (2015)* proposed the SHCEF framework to address security challenges in IoT-cloud integration. Similarly, *Jalasri & Lakshmanan (2023)* and *Unal et al. (2021)* introduced cryptographic methods and secure storage systems, respectively, to enhance data security in fog and cloud environments. Furthermore, *Selvarajan et al. (2023)* and *Uppuluri & Lakshmeeswari (2023)* focused on security models and protocols for IIoT systems and residential frameworks, respectively, to mitigate security vulnerabilities. Additionally, *Ahmad, Mehfuz & Beg (2023)* and *Sharma et al. (2023a)* and *Sharma et al. (2023b)* explored hybrid cryptographic methodologies and blockchain-based applications to enhance data security and management in cloud and healthcare systems. However, further research and testing were deemed necessary to evaluate its scalability and effectiveness in diverse IoT environments.

Proposed approach

In contrast, our proposed TIBDA framework offers a comprehensive solution to the multifaceted challenges of smart home technology. The framework integrates advanced trust mechanisms, hybrid cryptosystems, and machine learning algorithms to ensure enhanced data security, privacy, and anomaly detection capabilities.

- Key advantages of TIBDA
  1. Holistic approach: Firstly, while previous studies have focused on individual aspects such as authentication techniques or cryptographic methods, TIBDA takes a holistic approach by integrating advanced trust mechanisms, hybrid cryptosystems, and machine learning algorithms. This integration ensures not only enhanced data security and privacy but also improved anomaly detection and classification capabilities, essential for safeguarding smart home environments against emerging threats.
  2. Dynamic resource allocation: Moreover, TIBDA stands out in its ability to efficiently process data across edge, fog, and cloud computing architectures. By incorporating an artificial neural network (ANN) dynamic algorithm, our framework dynamically allocates computational tasks, optimizing resource utilization and responsiveness in real time. This dynamic approach to computing ensures scalability, efficiency, and the ability to handle enormous volumes of data, surpassing the capabilities of existing models.

3. Practical applicability: Unlike some existing frameworks that face scalability and interoperability challenges, TIBDA seamlessly integrates edge, fog, and cloud resources, making it suitable for diverse IoT environments.
4. Comprehensive evaluation: Our research rigorously evaluates TIBDA's performance across multiple datasets and scenarios, ensuring its reliability and effectiveness in real-world applications.

Through this comparative analysis, it is evident that the TIBDA framework offers significant advancements in smart home technology. By addressing critical security, privacy, and processing efficiency concerns, TIBDA sets a new standard for secure, efficient, and interconnected IoT ecosystems. As we continue to refine and validate our framework, we are confident that TIBDA will emerge as the preferred solution for ensuring the safety, privacy, and convenience of smart living environments.

In this section, we reviewed studies that have demonstrated the evolving landscape of the topic and the persistent challenges researchers have sought to address. As we delve into the methodology and findings of our study, we draw upon the insights gained from these previous works. The knowledge and limitations in this section have paved the way for our innovative approach to advance the field further. Compared to existing models, TIBDA stands out due to its holistic approach to addressing the multifaceted challenges of smart home technology. While other solutions may focus on individual aspects, such as data security or processing efficiency, our framework offers a comprehensive solution encompassing these concerns.

PROPOSED METHODOLOGY

The TIBDA methodology, with its comprehensive approach, aims to create a trustworthy, efficient, and adaptive framework for IoT big data and its analytics within smart home systems. The overall functionality of the proposed architecture is explained in subsections and represented in Fig. 2.

Objective function

The maximization of trust (T) and security (S), while ensuring data utility (U), is encapsulated in an objective function J. This formulation is crucial for balancing the competing priorities within the system. The weights w_1, w_2, w_3 represent the importance assigned to each component.

$$J = w_1 \cdot T + w_2 \cdot S - w_3 \cdot U. \tag{1}$$

This objective function combines trust, security, and data utility, providing a comprehensive metric for system optimization.

Data sources

Our research framework begins with the foundational layer of the smart home system, where many IoT devices, sensors, and appliances are strategically installed. These components collectively form the data generation infrastructure, producing a continuous stream of

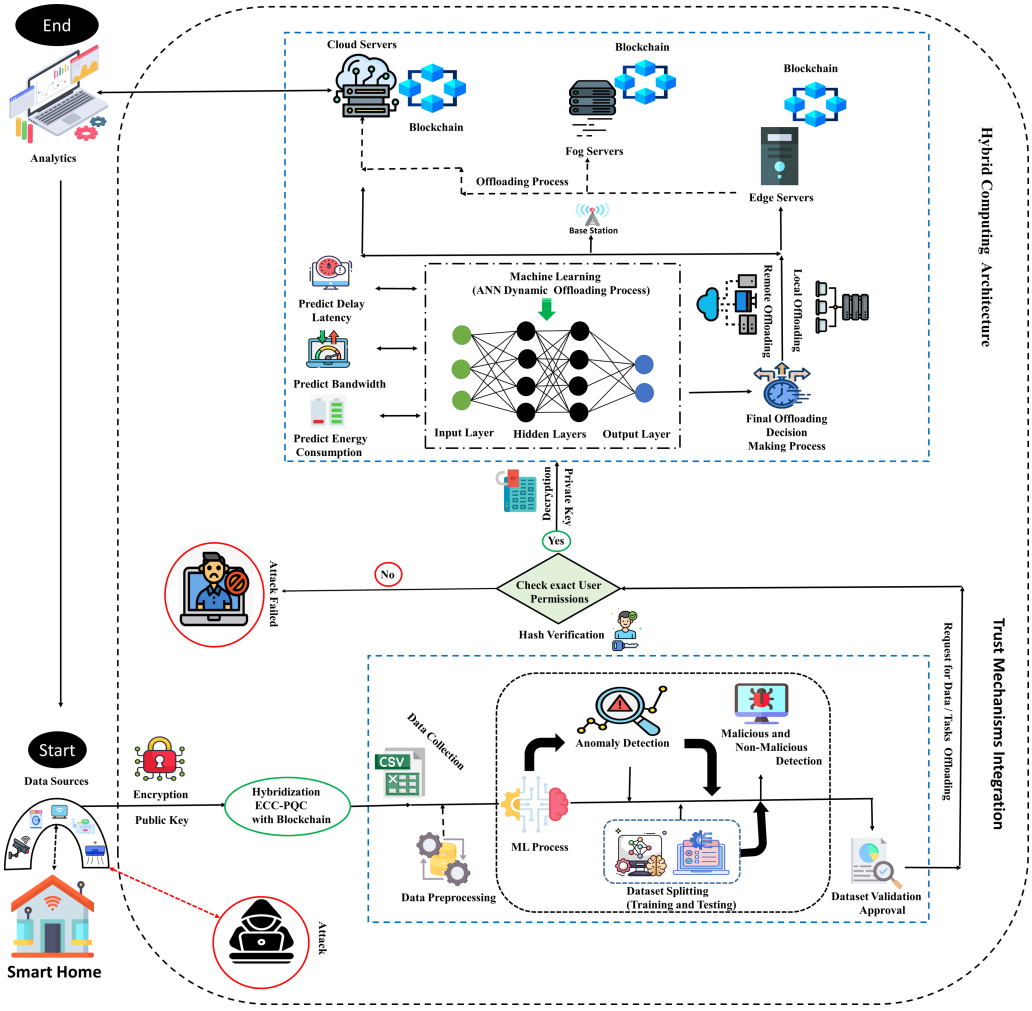

Figure 2 **Smart home user process flow with proposed TIBDA methodology.** It begins with user initiation, followed by data encryption with ECC-PQC and Blockchain. Anomaly detection is conducted using Isolation Forest and Random Forest algorithms. Hash verification confirms user permissions, and servers are selected *via* an ANN algorithm, secured with Blockchain. Finally, the process concludes with analytics processing, ultimately directing data for further processing, analytics, or action to the exact user. Image source credits: Smart home created by Freepik - Flaticon; Cctv Camera, created by Tru3 Art - Flaticon; Smart House, created by Tru3 Art - Flaticon; Smart TV, created by Corner Pixel - Flaticon; Internet Of Things, created by Chattapat - Flaticon; Router, created by Graficon - Flaticon; Hacker, created by Andrean Prabowo - Flaticon; Csv File, created by surang - Flaticon; Machine Learning, created by Iconjam - Flaticon; Result, created by Uniconlabs - Flaticon; Profile, created by Paul J. - Flaticon; Decryption, created by Nhor Phai - Flaticon; Bandwidth, created by Flat Icons - Flaticon; Latency, created by Vectors Tank - Flaticon; Decision, created by GOWI - Flaticon; Cloud Server, created by turkkub - Flaticon; Networking, created by Konkapp - Flaticon; Cpu Tower, created by Freepik - Flaticon; Broadcast, created by Freepik - Flaticon; Cloud Server, created by Uniconlabs - Flaticon; Blockchain, created by Freepik - Flaticon; Data Analytics, created by vectorsmarket15 - Flaticon; Encryption, Data Processing, Mind development, Testing, Malware, Block user, Batteries, Fog, www.freepik.com.

Full-size DOI: 10.7717/peerjcs.2211/fig-2

valuable information. The data generated by these sources serve as the bedrock for our comprehensive security and behavioral analytics.

Smart home systems leverage IoT devices, sensors, and appliances to enhance automation, convenience, and efficiency. The seamless operation of these systems relies on the continuous collection of data from various sources, providing valuable insights for smart decision-making and personalized user experiences. IoT (Internet of Things) devices form the backbone of a smart home ecosystem. These devices are embedded with sensors, actuators, and communication capabilities, enabling them to interact with the surrounding environment and communicate with each other. Common IoT devices in a smart home include smart thermostats, lighting systems, security cameras, and voice-activated assistants. Sensors capture real-time information about the physical environment within and around the smart home. These sensors can be embedded in various devices or strategically placed to monitor specific areas. Common sensors include motion sensors, door/window sensors, and light sensors. Smart appliances contribute valuable data points, enhancing the overall understanding of user behavior and optimizing energy consumption. Examples of smart appliances include refrigerators, washing machines, and ovens, which are equipped with sensors and communication interfaces.

Communication

In a smart home system, we integrated hybrid computing for further processing, a variety of IoT devices, sensors, and actuators collaborate to create an intelligent and responsive environment. IoT devices such as smart thermostats, light bulbs, and power outlets interact with sensors to receive data and instructions, while actuators respond to commands to perform actions like adjusting lighting levels or setting temperatures. Sensors continuously monitor parameters like temperature, humidity, and motion, collecting data that is locally processed and aggregated (*Zaidan et al., 2018*).

Communication within the smart home network is designed by using lightweight protocols like ZigBee or Bluetooth Low Energy (BLE), enabling efficient interaction between devices while conserving energy (*Bin Aftab, 2017*). Aggregated data is transmitted to a central hub or gateway, which communicates with the hybrid computing environment using protocols like MQTT or CoAP over Wi-Fi or Ethernet connections (*Yalcinkaya et al., 2020*; *Zaidan et al., 2018*; *Kashyap, Sharma & Gupta, 2018*; *Froiz-Míguez et al., 2018*). The hybrid computing environment, comprising local edge computing resources and cloud-based servers, further processes the data, leveraging protocols like HTTP for communication between components. Analysis results may trigger feedback or control signals sent back to the smart home system, enabling automated actions or adjustments based on the processed data. This collaborative process, facilitated by efficient communication protocols, allows the smart home system to optimize energy usage, enhance comfort, and improve overall functionality (*Huang, Chen & Zhang, 2020*; *Díaz, Martín & Rubio, 2016*).

Device-to-hub interaction: enabling processing, feedback, and integration

Equations (2)–(5) elucidate the intricacies of Device-to-Hub interaction, enabling robust processing, comprehensive feedback mechanisms, and seamless integration in smart home

systems.

$$\text{Communication}_{\text{Devices-Hub}} = \sum_{i=1}^{|D|} \text{Protocol}_{\text{Device}_i-\text{Hub}} + \sum_{j=1}^{|S|} \text{Protocol}_{\text{Sensor}_j-\text{Hub}} + \sum_{k=1}^{|A|} \text{Protocol}_{\text{Actuator}_k-\text{Hub}} \tag{2}$$

where:

- $|D|$ is the cardinality of the set of IoT devices.
- $|S|$ is the cardinality of the set of sensors.
- $|A|$ is the cardinality of the set of actuators.
- $\text{Protocol}_{\text{Device}_i-\text{Hub}}$, $\text{Protocol}_{\text{Sensor}_j-\text{Hub}}$, and $\text{Protocol}_{\text{Actuator}_k-\text{Hub}}$ represent the communication protocols (*e.g.*, ZigBee, BLE) used between each device (IoT device, sensor, or actuator) and the hub/gateway.

$$\text{Processing}_{\text{Aggregated Data}} = \text{Local Edge Computing} + \text{Cloud-based Computing} \tag{3}$$

$$\text{Feedback}_{\text{Analysis Results}} = \text{Feedback}_{\text{Control Signals}} \tag{4}$$

$$\text{Smart Home System} = \text{Communication}_{\text{Devices-Hub}} + \text{Processing}_{\text{Aggregated Data}} + \text{Feedback}_{\text{Analysis Results}}. \tag{5}$$

Securing smart homes: modeling and mitigating adversarial sensor threats

The preservation of data integrity and security within the smart home ecosystem is imperative. It is essential to fortify the protection of collected data against adversarial manipulations that could otherwise jeopardize the optimal functioning of the smart home system.

Let S be the set of sensors in the smart home system, and A be the set of adversaries. Each sensor s_i has an associated legitimacy variable x_i, where $x_i = 1$ indicates legitimacy and $x_i = 0$ indicates compromise. The presence of a compromised sensor s_k is indicated by $x_k = 0$ compromised state as shown in Eq. (8). The overall legitimacy vector is represented as:

$$\mathbf{x} = [x_1, x_2, \ldots, x_k, \ldots, x_n] \tag{6}$$

where n is the total number of sensors.

Additionally, the influence of compromised sensor s_k on the system can be captured by an impact function $f_k(\mathbf{x})$, which evaluates the effect of s_k's compromised state on the

performance and reliability of the system. The security concern is mathematically expressed as:

$$\min_{\mathbf{x}} \left(\sum_{k=1}^{|S|} f_k(\mathbf{x}) \right) \tag{7}$$

Subject to the constraint:

$$x_k = \begin{cases} 1 & \text{if sensor } s_k \text{ is legitimate} \\ 0 & \text{if sensor } s_k \text{ is compromised} \end{cases} \tag{8}$$

This formulation encapsulates the objective of minimizing the cumulative impact of compromised sensors on the smart home system's performance and reliability, considering the integration of various components such as IoT devices, sensors, actuators, and hybrid computing resources. This entails designing and implementing security measures, such as anomaly detection or attack mitigation algorithms, to ensure data integrity and security within the smart home ecosystem.

Multi-layered secure cryptographic integration across hybrid computing

To ensure the reliability and privacy of data within the smart home ecosystem (*Yang & Sun, 2022*; *Buil-Gil et al., 2023*). We embedded the trust mechanisms within our comprehensive framework. In a hybrid computing environment for IoT-enabled smart home systems, the combination of ECC, PQC, and Blockchain-centric security paradigms can work synergistically to enhance security (*Irshad et al., 2023*).

In a hybrid computing environment, these components can work together to provide a multi-layered security approach for IoT-enabled smart home systems. For example, ECC can be used for device authentication and secure communication, while PQC algorithms can be deployed to protect sensitive data against quantum attacks. Meanwhile, Blockchain technology can be leveraged to securely manage device identities, access control policies, and data transactions within the smart home ecosystem. By integrating ECC, PQC, and Blockchain-centric security paradigms, smart home systems can achieve robust security against various threats, ensuring data and services' privacy, integrity, and availability.

To understand the basic operation of the proposed MSCHC in TIBA's framework, we conduct a thorough examination of its architecture and thread detection mechanism, as illustrated in Fig. 2. This Figure illustrates how the MSCHC model, which includes detectors, actuators, regulators, and sensors, is carefully designed and developed to function as a central location for controlling the Internet of Things devices. These devices produce data, which is easily sent to hybrid servers. Many important parts of the system work together to create a safe and dependable environment that meets the needs of many users. The following is a description of the operational dynamics of these various components inside the proposed model:

(a) Hybrid computing infrastructure: Offloading server farms play a pivotal role, comprising interconnected physical servers tasked with managing processing and storage needs for IoT devices. This setup guarantees the server's platform's high availability and

scalability, ensuring seamless operations even under varying workloads (*Zhang et al., 2018*; *Yar et al., 2021*).

(b) Virtual servers (VS): VS constitute the infrastructure's main part, facilitating the hosting and oversight of IoT devices and their accompanying data. These virtualized resources (Edge, Fog, Cloud) furnish the requisite computational potency and storage capacity for seamless operations.

(c) Elliptic curve cryptography (ECC): It is a robust and efficient cryptographic tool for securing communications, transactions, and data across diverse applications. Its versatility extends to safeguarding mobile devices, IoT systems, and integral communication protocols like TLS/SSL, ensuring robust security measures are in place. ECC offers efficient and robust cryptographic primitives suitable for resource-constrained devices like IoT sensors and actuators commonly found in smart homes. ECC can provide secure authentication, data confidentiality, and integrity protection, ensuring that only authorized devices can access the system and that communication channels remain secure (*Yusoff et al., 2022*; *Majumder et al., 2021*; *Sanaa et al., 2020*; *Ullah et al., 2023*).

(d) Attack detection Isolation Forest and random forest are both machine learning algorithms widely employed for anomaly detection, particularly within cybersecurity contexts. Isolation Forest, an unsupervised learning algorithm, excels at isolating outliers by randomly partitioning data points until isolation is achieved, enabling rapid identification of anomalies. Similarly, random forest, an ensemble learning algorithm, constructs multiple decision trees using random subsets of features and data points, providing robust classification capabilities for identifying abnormal patterns in data. In cybersecurity applications, both algorithms play pivotal roles in scrutinizing network traffic, system logs, and user behaviors to detect potential security threats and intrusions, contributing to enhanced threat detection and mitigation strategies (*Bilgin, Kilinc & Zaim, 2022*; *Kandhoul, Dhurandher & Woungang, 2021*).

(e) PQC encryption methods: It plays a vital role in safeguarding communication content by traversing multiple users and servers. Utilizing algorithms resistant to quantum attacks, PQC ensures the confidentiality and integrity of transmitted data, thereby fortifying the system's security. PQC algorithms are designed to resist quantum computer attacks, threatening traditional cryptographic schemes like RSA (Rivest-Shamir-Adleman). By integrating PQC into smart home security systems for hybrid computing purposed, we can future-proof them against emerging quantum threats, ensuring that sensitive data remains secure even as quantum computing technology advances (*Kumari et al., 2022*; *Lee et al., 2021*; *Sharma et al., 2023b*).

(f) Blockchain nodes: They play a crucial role within the network, actively engaging in the blockchain infrastructure to uphold a distributed registry responsible for recording and overseeing IoT device data. Leveraging its secure and transparent characteristics, the blockchain guarantees the accuracy and immutability of data, instilling trust and reliability within the ecosystem. Data integrity and immutability are guaranteed by blockchain technology's decentralized, tamper-resistant infrastructure. In the context of smart homes, a Blockchain-centric security paradigm can securely store device data, manage access control permissions, and facilitate secure transactions between devices. Each block in the

Blockchain contains a cryptographic hash of the previous block, creating a secure chain of blocks that cannot be altered without consensus from the network participants (*Lin et al., 2019*; *Ammi, Alarabi & Benkhelifa, 2021*; *Tchagna Kouanou et al., 2022*).

Various security techniques are integrated into the architectural design to ensure the safety of data transfer, storage, and administration on administration in the hybrid computing paradigm. These techniques are strategically positioned at various stages in the data flow process to ensure data availability, confidentiality, and integrity from Internet of Things devices.

- Authorization serves (AS): It is an important part of the system in charge of confirming and giving users the necessary access rights. Only authenticated users may communicate with the IoT devices and hybrid computing services through this method, guaranteeing strong security and control over the system's functioning (*Chifor et al., 2018*).
- Access management (AM): It is essential for controlling and monitoring user access to the many resources and functionalities that are offered by the hybrid computing platform. User rights must be carefully managed to prevent illicit activities, safeguard data privacy, and ensure system integrity (*Mohammad et al., 2021*).
- Attacks detection: The proposed MSCHC strategically combines Isolation Forest, random forest, and PQC Algorithms to use intrusion detection and anomaly detection capabilities inside its hybrid-based computing for offloading infrastructure. Its resistance against attacks is strengthened by this integration, which improves its capacity to recognize and proactively address possible security threats (*Saxena, Sodhi & Singh, 2017*; *Anthi, 2022*).
- Security measures: The system then uses data encryption and decryption procedures, further enhancing security protections. These preventive actions ensure the strong integrity and confidentiality of the transmitted data by serving as a disincentive to unwanted changes or tampering efforts (*Ahanger et al., 2020*).
- Access control mechanisms (ACM): These are crucial for managing the privileges and permissions awarded to users. These solutions guarantee data privacy and security as they travel from edge-cloud servers to many users *via* IoT devices by imposing stringent control over user behavior (*Sikder et al., 2020*). The core of the proposed MSCHC system's security architecture is a hybridized cryptosystem that combines ECC, PQC, and Blockchain technology. To ensure robust security protocols are implemented across the system, this sophisticated cryptographic system combines a number of essential components, such as random forest and Isolation Forest for attack detection, PQC algorithms for optimal key generation, and Blockchain for cryptography. Through the incorporation of Blockchain methodology with PQC and attack detection mechanisms like random forest, Isolation Forest, and PQC Algorithms, the proposed hybrid computing environment, the hybridized cryptosystem establishes a solid foundation for secure communication, data storage, and access control. It strengthens the system's defences against any assaults and ensures the confidentiality of sensitive data while it is being transmitted.

Elliptic curve cryptography

Elliptic curve features are used in public-key cryptography to offer security. It functions more like other public-key cryptography systems, such as Rivest-Shamir-Adleman (RSA), due to its reduced key sizes. This makes it more practical in limited settings like Internet of Things devices and smartphones.

In our study, we integrate ECC into our system to ensure secure communication and data exchange between the smart home IoT devices and hybrid-based computing offloading infrastructure. Here is how we describe the ECC implementation along with the corresponding equations:

1. Key generation: ECC key pairs are generated for each entity in the system, including IoT devices and offloading servers. Private keys (SK) are randomly generated integers, while public keys (PK) are computed points on the elliptic curve as represented in the Eqs. (9) and (10).

$$\text{Private Key (SK)} : \text{Randomly generated integer} \tag{9}$$

$$\text{Public Key (PK)} : \text{Computed point on the elliptic curve} \tag{10}$$

2. Encryption and decryption: Asymmetric encryption and decryption are performed using ECC. Encryption involves the use of the recipient's public key ($\text{PK}_{\text{recipient}}$) to encrypt the message (M), resulting in the ciphertext (C). Decryption utilizes the recipient's private key ($\text{SK}_{\text{recipient}}$) to recover the original message from the ciphertext, which is shown in Eqs. (11) and (12).

$$C = \text{Encrypt}(M, \text{PK}_{\text{recipient}}) \tag{11}$$

$$M = \text{Decrypt}(C, \text{SK}_{\text{recipient}}) \tag{12}$$

3. Digital signatures: ECC-based digital signatures are employed to ensure message authenticity and integrity. Signing involves the use of the sender's private key ($\text{SK}_{\text{sender}}$) to generate a signature (σ) for the message (M) as shown in Eq. (13). Verification is performed by the recipient using the sender's public key ($\text{PK}_{\text{sender}}$) to verify the signature as represented in Eq. (14).

$$\sigma = \text{Sign}(M, \text{SK}_{\text{sender}}) \tag{13}$$

$$\text{Verify}(\sigma, M, \text{PK}_{\text{sender}}) \tag{14}$$

4. Security and efficiency: ECC provides a high level of security with smaller key sizes compared to other cryptographic techniques. This ensures efficient computation and reduced resource consumption, making it suitable for resource-constrained environments like IoT devices and hybrid-based computing systems. The overall process of ECC can be summarized with the following Eq. (15):

$$\text{ECC}(M, \text{PK}_{\text{recipient}}, \text{SK}_{\text{recipient}}) \rightarrow C = M + \text{PK}_{\text{recipient}} - \text{SK}_{\text{recipient - 1}} \tag{15}$$

where M represents the message, $PK_{recipient}$ is the recipient's public key, $SK_{recipient}$ is the recipient's private key, and C is the resulting ciphertext. This equation encapsulates ECC's encryption and decryption process, ensuring secure communication between parties.

By integrating ECC into our system architecture and leveraging its security features, we ensure the confidentiality, integrity, and authenticity of data exchanged between entities. These equations and descriptions highlight the effective implementation of ECC in our system for secure communication and data protection.

Isolation forest anomaly detection

There should be security checks that investigate the security and privacy concerns. To detect anomalies in smart home systems, we comprehensively compared four prominent artificial intelligence anomaly detection algorithms: Isolation Forest, Local Outlier Factor, One-Class SVM, and Elliptic Envelope. Our research findings revealed that Isolation Forest outperformed other anomaly detection algorithms presented in detail. The subsequent step in our study involves selecting the Isolation Forest algorithm as a robust approach for anomaly detection (*Shah et al., 2023*; *Dilraj, Nimmy & Sankaran, 2019*; *Tang et al., 2019*; *Chithaluru et al., 2024*; *Kang et al., 2020*).

In 2008, *Liu, Ting & Zhou (2008)* presented Isolation Forest (iForest), a pioneering anomaly detection method emphasizing anomaly isolation over regular instance profiling. While iForest demonstrates remarkable efficiency and effectiveness in detecting anomalies, the study identifies limitations such as needing an additional attribute selector in high-dimensional datasets and susceptibility to certain types of anomalies or noise. Our research aims to address these limitations by exploring techniques to enhance iForest's performance in high-dimensional settings and mitigate the impact of anomalies and noise, ultimately contributing to the advancement of anomaly detection methodologies.

So, we specifically designed it to identify instances that deviate from the norm within a dataset. Leveraging the principles of decision trees, this algorithm helps us to excel in efficiently isolating anomalies based on their sparsity in the data. The algorithmic flow of the Isolation Forest algorithm can be observed from Algorithm 1. Utilizing the Isolation Forest algorithm, our research employs this powerful tool for anomaly detection across various use cases, including fraud detection, intrusion detection in cybersecurity, and identifying rare events in diverse domains. The key concept guiding the Isolation Forest is its adeptness in efficiently isolating anomalies. Anomalies, being infrequent and having distinct characteristics, are anticipated to necessitate fewer partitions in a tree structure for isolation.

Anomaly score calculation

The anomaly score for each instance is computed based on the average path length to isolate the instance across multiple trees (*Jain et al., 2021*). The calculation formulas for the anomaly score can be represented through the following Eq. (16):

Split condition equation:

$$\text{Split Condition}(x, F_i, P_i) = \begin{cases} 1 & \text{if } x[F_i] < P_i \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

Algorithm 1: Isolation forest algorithm for anomaly detection

Inputs: data: $D'_{i,j}$

Output: Classification of anomalous and nominal.

Build Isolation Trees:

for *i* **to** *n_trees* **do**

Randomly select a feature F_i;

Randomly select a split value P_i;

Construct an isolation tree using recursive splitting based on the SplitCondition:

$$\text{SplitCondition}(x, F_i, P_i) = \begin{cases} 1 & \text{if } x[F_i] < P_i \\ 0 & \text{otherwise} \end{cases}$$

end for

Calculate Anomaly Scores:

for *x* **to** *data* **do**

Compute the average path length in the isolation trees, denoted as $h(x)$, using:

$$h(x) = 2^{\frac{-\bar{c}(n)}{\bar{c}(n-1)}} - 1$$

end for

Decision Threshold:

Set a decision threshold based on the contamination parameter:

$$\text{threshold} = \text{Percentile}(\text{average path lengths}, 1 - \texttt{contamination})$$

Calculate Anomaly Scores:

for *x* **to** *data* **do**

If $h(x) <$ threshold, mark x as an anomaly;

Otherwise, mark x as a normal instance;

end for

Equation (16) represents the condition for splitting a node in an isolation tree. For a given data point x, feature F_i, and split value P_i, if the value of the feature $x[F_i]$ is less than the split value P_i, the condition is true (1); otherwise, it is false (0). This binary decision determines whether the data point goes to the left or right subtree during tree construction.

- Average path length equation:

$$h(x) = 2^{\frac{-\bar{c}(n)}{\bar{c}(n-1)}} - 1 \tag{17}$$

In Equation (17), $h(x)$ represents the average path length for a data point x in the isolation trees. $\bar{c}(n)$ is the average path length of an unsuccessful search in a binary tree with n nodes. $\bar{c}(n-1)$ is the average path length with $n-1$ nodes, considering the unsuccessful search includes the root node. The equation calculates the ratio of average path lengths for n and $n-1$ nodes, applies it as an exponent to 2, and subtracts 1.

- Decision threshold equation:

$$\text{threshold} = \text{Percentile}(\text{average path lengths}, 1 - \text{contamination}) \tag{18}$$

The decision threshold is determined based on the contamination parameter in Eq. (18), which represents the expected fraction of anomalies in the dataset. The threshold is set as the $(1 - \texttt{contamination})$ percentile of the distribution of average path lengths. This helps identify data points with path lengths indicating potential anomalies.

These equations collectively form the core calculations in the Isolation Forest algorithm, aiming to isolate anomalies based on their sparsity in the data efficiently. The use of random splits and averaging over multiple trees makes the algorithm robust and effective for anomaly detection.

Random forest for malicious vs. non-malicious attack detection

It is imperative to acknowledge that, given the integration of the smart home system with local networks such as Wi-Fi, potential alterations or attacks may transpire (*Alalade, 2020*; *Yang & Sun, 2022*; *Buil-Gil et al., 2023*; *Tanwar, Ramani & Tyagi, 2018*). Consequently, it becomes essential to systematically conduct a rigorous process for detecting both malicious and non-malicious activities within the system (*Ul Hassan et al., 2022*; *Andelić, Baressi Šegota & Car, 2023*). In the realm of cybersecurity, distinguishing between malicious and non-malicious activities is a critical task. Applying machine learning algorithms like random forest provides an effective automated detection and classification method. Random forest is an ensemble learning method that leverages the power of multiple decision trees to enhance predictive accuracy and robustness. The algorithmic flow of the random forest model is shown in Algorithm 2.

The next crucial step is the distribution of datasets into subsets tailored for training and testing. The dataset is typically divided into two main segments:

- Training set:

This subset constitutes most of the dataset and serves as the input for training our machine-learning models. The algorithms learn patterns, relationships, and features from this data portion.

- Testing set:

The testing set is reserved for evaluating the trained models' performance. It acts as an unseen dataset that allows us to assess how well our models generalize to new, previously unseen data.

The allocation for this research work involves utilizing 70% of the dataset for training and 30% for testing, ensuring a comprehensive evaluation of our models' capabilities on both learned and unseen data. This split ratio was chosen based on considerations of dataset size, model complexity, and the need for a robust evaluation framework. We have utilized machine learning classification algorithms such as random forest (RF), k-nearest Neighbors (KNN), Support Vector Machines (SVM), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA) for detecting malicious and non-malicious activities. Based on the deep analysis, random forest exhibited superior performance in detecting both malicious and non-malicious activities, as presented in our study. This algorithm distinguishes between different classes, such as malicious and non-malicious

activities, based on patterns observed in the training data. The random forest algorithm, characterized by its ensemble approach, captures intricate patterns within complex datasets. This attribute makes it particularly adept at making accurate predictions, rendering it a valuable asset in the cybersecurity domain. Its capacity to distinguish between malicious and non-malicious activities within a smart home system underscores its relevance and effectiveness in safeguarding against security threats.

Algorithm 2: Random forest for malicious vs. non-malicious attack detection

Inputs:
X: Feature matrix representing the dataset.
y: Labels indicating malicious (1) or non-malicious (0) for each instance.
Output:
Predicted labels for detection.
Initialize Random Forest Classifier:
Initialize a Random Forest classifier:

$$\text{RandomForest} = \text{RFC}(n\_estimators = N, max\_depth = D, \ldots)$$

Split the Dataset:
Split the dataset into training and testing sets:

$$X_{\text{train}}, X_{\text{test}}, y_{\text{train}}, y_{\text{test}} = \text{train\_test\_split}(X, y, test\_size = 0.3)$$

Train the Classifier:
Train the Random Forest classifier using the training dataset:

$$\text{RandomForest.fit}(X_{\text{train}}, y_{\text{train}})$$

Make Predictions:
Make predictions on the testing dataset:

$$\hat{y}_{\text{pred}} = \text{RandomForest.predict}(X_{\text{test}})$$

return Predicted labels for detection ($\hat{y}_{\text{pred}}$).

Hybridization of ECC-PQC and blockchain system

In this section, we leverage ECC-PQC and blockchain technology benefits to improve PQC algorithms' performance for data and transactions stored in the blockchain. Additionally, we investigate how to maximize key identification using ECC, enhancing the effectiveness and security of the system's cryptographic operations (*Lara-Nino, Morales-Sandoval & Diaz-Perez, 2021*). The goal is to create a coherent link with PQC systems and combine blockchain technology with an enhanced ECC methodology. To improve IoT-based hybrid computing systems, we create and integrate the hybrid ECC-PQC with the Blockchain system. This attempt involves thoroughly adjusting and changing pertinent equations within these systems to ensure a cohesive and seamless structure. Our study has carried

out a thorough investigation of an Internet of Things-based hybrid system intended for many user devices. By using this strategy, we succeeded in creating reliable and trustworthy communication routes between IoT devices and hybrid servers by utilizing the strong encryption capabilities provided by ECC-PQC.

- Mathematically presentation:

Suppose an environment in which there are many n IoT devices, each identified as follows:

$$Dev_1, Dev_2, Dev_3, \ldots, Dev_n \tag{19}$$

As in the equation, several dedicated devices are assigned. Each Internet of Things device is given a unique identity, U_i, and generates a personalized symmetric key (k_i). This key (k_i), is essential for encrypting data before sending it to the appropriate hybrid servers.

Before sending sensitive data to the offloading server, the symmetric crucial k_i is carefully encrypted using the powerful Post-Quantum Cryptography (PQC) technique to fortify security measures. Notably, the ECC approach serves as the basis for the complex problem of estimating optimal keys. The thorough security protocol is implemented employing a series of carefully planned steps:

- **Key generation process of ECC:**

In cryptography, the process of generating a pair of keys, a public key and a confidential private key, is essential. These keys are carefully generated with secure strategies that follow certain cryptographic techniques. The public key, as its name implies, is essential to encryption because it is freely shared among users and allows anyone with it to encrypt data securely. On the other hand, the private key needs to be kept confidential and known by the authorized user alone. Encrypted data is decrypted using this private key, guaranteeing that only those with permission can view the original data. Techniques for generating random or pseudorandom numbers can also be combined with key derivation algorithms, which produce keys from a primary key to boost security. It is important to note that the strength and unpredictability of the produced keys significantly impact the overall security of cryptographic systems. Each entity (Dev_n), generates a public–private key pair for Post-Quantum Cryptography (PQC) using the ECC technique. The public key (k_{PQC_i}) is shared freely, while the private key ($k_{PQC_{i-1}}$) is kept hidden. This key combination is necessary for securely encrypting and decrypting sensitive data in the context of post-quantum computing.

- **Data view:**

Usually, the data to be encrypted is converted into a numerical format, most frequently a bit sequence or vector. This transformation allows the data to carry out mathematical operations, which makes processing and manipulation easier while encrypting data.

- **Public key matrices:**

Public key matrices enhance numerical vectors obtained from the data during encryption. These matrices are purposefully made publicly accessible and are produced

throughout the key creation process. These matrices' attributes and format are determined by the encryption method, and the ECC process further shapes them.

- **Quantum encryption:**

Information is securely transformed into an encrypted form through a series of processes in the encryption process. The symmetric key (k_i) is encrypted using the PQC public key (k_{PQC_i}). This initiates the use of the encryption function $E(k_i, k_{PQC_i})$ to create a ciphertext (x_i). This resulting ciphertext x_i represents the encrypted form of the symmetric key (k_i), which guarantees the key's confidentiality throughout storage or transmission. The encryption process consists of multiple phases to convert data into an encrypted format safely. In order to generate a ciphertext (x_i), we employ the encryption function ($E(k_i, k_{PQC_i})$) to encrypt the symmetric key (k_i). We do this by using the PQC public key (k_{PQC_i}). As a result, the symmetric key (k_i) is represented by the encrypted ciphertext (x_i), which ensures the key's confidentiality throughout storage or transfer as shown in the Eq. (20).

Encryption of symmetric key:

$$x_i = E(k_i, k_{PQC_i}) \tag{20}$$

- **Vector augmentation:**

The encrypted vector (EV), which comprises the altered and encrypted version of the original data, is created *via* the vector augmentation (VA) method. Due to the extra information added during the augmentation process, the encrypted vector usually has a bigger size than the original vector.

- **Encryption vector:**

The encrypted vector (EV), or the original data in its modified and encrypted form, is the result of the vector augmentation (VA) process. Because more data is added during the augmentation step, the encrypted vector is typically bigger than the original vector.

- **Security and confidentiality:**

The ECC grants information security and confidentiality acquired by augmentation using the best public key matrices within the ECC. Without the corresponding private key, it is not possible to computationally reverse the encryption process or retrieve the original data from the encrypted vector. Once it gets access, the offloading server may efficiently encrypt and decrypt the data received by the IoT device using the initial symmetric key (k_i). Thanks to this method, sensitive information will be safeguarded throughout its journey from the IoT device to the destination server. Within the framework of an IoT-based hybrid servers system with several users, the ECC-PQC encryption process can be briefly represented as follows:

- **Hash verification:**

A blockchain eliminates the requirement for a single point of failure by transparently recording transactions *via* a distributed, immutable electronic registry and several

computations. Within its blocks, the blockchain technology safely stores encrypted data. A cryptographic hash function (H) is used to calculate the hash value of block (β_i) to preserve the security and integrity of a blockchain. The input for this hash function is the data of block (β_i) concatenated with a nonce value (η_i) that users choose to satisfy particular criteria.

The generated hash value (H), which uniquely identifies block (β_i) on the blockchain, including the one that is now active, is a fixed-length string of bits. Users mine in a dangerous and computationally demanding way to meet requirements, like having a certain amount of trailing zeros. The challenge at present involves finding a nonce value (η_i) that, when combined with the data in the block (β_i), provides a hash value satisfying the required conditions. Miners need a lot of computing power and cost to compete with one another and produce a legitimate nonce.

The first miner to find a nonce value that meets the requirements receives a reward for their mining contribution. The use of the hash function and nonce value guarantees the immutability of the blockchain. Because it is computationally difficult to alter any data in a previous block without also recalculating the hash values of the succeeding blocks, the blockchain's integrity is preserved. Let $H(\beta_{i-1})$ stand for the previous block's hash value ((β_{i-1})). P_i indicates the plaintext data stored in the current block. $f(E)$ is the ideal ECC-PQC encryption function, and k_{PQC_i} is the ideal PQC public key. Let β_i be a random nonce value. Next, the following formula can be used to get β_i, the hash value of the current block. The representation of β_i to calculate the hash value of the current block is as follows:

$$\beta_i = H(\beta_{i-1}||P_i||f(\hat{E})P_i, (k_{PQC_i})||\eta_i) \tag{21}$$

Here,

- $H(\beta_{i-1})$ is a representation of β_{i-1}, the hash value of the previous block.
- P_i represents the plaintext material stored in this block.
- $f(\hat{E})$ symbolizes the best encryption function for ECC-PQC.
- k_{PQC_i} stands for the ideal PQC public key.
- η_i indicates a nonce value that is randomized.

On calculating the hash value of the current block, this equation concatenates the previous block's plaintext data, the current block's hash value, the result of applying the optimum ECC-PQC encryption function on the plaintext data, and the randomized nonce value.

A complex mathematical challenge needs to be resolved for the hybrid ECC-PQC with the Blockchain system's consensus and mining process to add a new block (β_i) to the blockchain. Let S_c be the current state of the blockchain, and let N be the group of network nodes. To add a new block (β_i) to the blockchain, a node (miner) must discover a nonce value (η_i) such that the hash value of the current block $H(i)$ satisfies specific conditions, including having a specified amount of leading zeros. A representation of the mining process in the hybrid ECC-PQC with Blockchain system is as follows:

$$\beta_i = \text{argmin}_i\{H(Sc||\beta_{i-1}||P_i||f(\hat{E})(P_i, k_{PQC_i}))||(\eta_i)|H(Sc||\beta_{i-1}||P_i||f(E(P_i, k_{PQC_i}))||\eta_i)$$

satisfies criteria} (22)

Here,

- β_i Indicates the next block that will be entered into the blockchain.
- Sc depicts the blockchain's current state.
- β_{i-1} represents the previous block.
- P_i symbolizes the plaintext information that will be kept in the current block.
- $f(\hat{E})$ represents the optimal ECC-PQC encryption function.
- k_{PQC_i} represents the optimal PQC public key.
- η_i represents the nonce value to be found by the miner.
- H represents the cryptographic hash function.
- Specific criteria for the hash value $H(i)$ may include having a certain number of leading zeros.

This equation reflects the process where a node (miner) aims to find a nonce value (η_i) that satisfies specific criteria for the hash value of the current block, allowing the addition of a new block (β_i) in the blockchain.

The combination is shown in the context given by ||, and the encrypted data that has to be added to β_i is represented by x_i. The miner needs to constantly cycle through a number of β_i values in order to identify a nonce that generates a suitable H value. Once a nonce satisfies the requirements, the miner broadcasts the new β_i to the network for verification. The two steps involved in verification are verifying that the value stored in the state $S_c(\beta_i)$ matches the nonce value (η_i) of the previous block (β_{i-1}) in the blockchain and validating the nonce value used to generate the hash value of the current block (β_i) as indicated by Eq. (23).

$$S_c(\beta_i) = \eta_i \tag{23}$$

Here,

- $S_c(\beta_i)$ represents the state of nonce value (η_i) for block (β_i) in the blockchain.

This process ensures the integrity and validity of each blockchain block, maintaining the system's security and reliability.

Nodes in the network assess the optimum PQC encryption technique ($f(\hat{E})$) used to encrypt the data in block β_i and make sure the PQC public key (k_{PQC_i}) used is dependable, in addition to confirming the nonce value and the hash criteria. This has the following mathematical representation shown in Eq. (24).

$$f(\hat{E})(P_i, k_{PQC_i}) = x_i \tag{24}$$

Here, $f(\hat{E})(P_i, k_{PQC_i})$ represents the encryption function applied to the plaintext data (P_i) using the optimal PQC public key (k_{PQC_i}), resulting in the encrypted data (x_i). This validation step is crucial for maintaining the security and integrity of the blockchain system, as it ensures that only trusted encryption processes and keys are used to protect the data within each block. The computational difficulties of the mathematical problems involved

in the blockchain consensus mechanism and the ECC-based PQC encryption process define the security of the hybrid ECC-PQC with the Blockchain system. These puzzles are made to be difficult for both conventional and quantum computers, providing a strong protection against security breaches.

- **Quantum decryption:**

The original information is recovered by decrypting the encrypted data using the confidential private key. The ECC-PQC private key (k_{PQC_i-1}) connected to the device (Dev_n) must be accessible to the offloaded server in order to decrypt the ciphertext (x_i). The offloading server carries out the decryption operation ($d = x_i, k_{PQC_i-1}$) using this private key. The original symmetric key (k_i), which was used to encrypt the data, is produced by this operation. The offloading server can access and retrieve the encrypted data if it has the original symmetric key.

Decryption of symmetric Key:

$$k_i = d(x_i, k_{PQC_{i-1}}) \tag{25}$$

Using the original symmetric key (k_i), the offloading hybrid server (edge, fog, or cloud) can generate ciphertext and extract the original data transmitted by the IoT device. Through this procedure, the shared data of the offloading server and IoT devices is kept safe and out of the hands of unauthorized individuals. Any offloading server, including edge, fog, and cloud, can securely encrypt and decrypt data while maintaining the sent information's confidentiality and integrity by using a symmetric key.

Algorithm 3 delineates a comprehensive process for hybridizing ECC and PQC with blockchain technology. It begins by initializing the system, generating ECC-PQC key pairs for IoT devices and a nonce for hashing. Next, data from IoT devices undergoes ECC-PQC encryption, followed by server selection based on predictions from an artificial neural network (ANN). The output includes the encrypted data and the chosen server. Subsequently, the algorithm computes hash values necessary for blockchain, applies additional cryptographic operations for enhanced security, and securely stores the resulting hash value. Verification procedures ensure the integrity of the hash during retrieval, handle any collisions, and update cryptographic standards. Finally, encrypted data is decrypted using ECC-PQC decryption on offloading servers. This comprehensive approach ensures secure communication and data integrity in a hybrid computing environment comprising edge, fog, and cloud components.

Our research methodology is not static; it involves continuous iteration and improvement. As new data becomes available and the smart home environment evolves, our models are retrained and refined to adapt to emerging patterns and potential security threats. The distribution and classification of data within the smart home system security framework are crucial components that contribute to the overall success of our research. Through thoughtful data preprocessing, strategic dataset distribution, and the application of robust classification models, we aim to enhance the security posture of smart home environments and provide adequate protection against potential cyber threats. This research acknowledges the necessity for ongoing efforts in adapting security measures

Algorithm 3: Hybridization of ECC-PQC and blockchain algorithm

Input: IoT devices, Encryption

Initialization:
Generate ECC-PQC key pairs for IoT devices
Generate nonce R for hashing

ECC-PQC Encryption:
for *each IoT device* Dev_n **do**
  Generate ECC-PQC key pair (k_{PQC_i}, k_{PQC_i-1})
  Encrypt data: $x_i = f(\hat{E})(P_i, k_{PQC_i})$
  ▷ Based on Eq. (20) and (24)
end for
Output: Encrypted data

Input: Private key (SK), Public key (PK), Offloading servers available for selection
Hashing:
Compute hash of previous block: $H(\beta_{i-1})$
▷ Based on Eq. (21)
Compute hash of encrypted data: $H[f(\hat{E})(\beta_{i-1}, k_{PQC_i})]$
Choose randomized nonce: η_i
▷ Based on Eq. (22)
Estimate blockchain state hash: $H[\eta_i||(H(\beta_{i-1})||H(k_{PQC_i}, P_i))]$
Hash and concatenate the public key PK, private key SK, and nonce R:
$H = \text{Hash}(H_{SK}||H_{PK}||R)$
▷ Based on Eq. (23)
Additional Cryptographic Operations:
Compute the Random Forest and Isolation Forest algorithms for enhanced security

Store Hashed Verification:
To improve security, do more cryptographic operations on H.

Verification and Integrity Check:
Verify integrity of H during retrieval
Handle collisions, if any
Update security-related cryptography standards.

ANN Server Selection:

Compute the ANN algorithm for suitable server selection
$\{D_i^L, D_i^E, D_i^F, \ldots, D_i^C\} \in [0, 1]$
▷ Based on Eq. (33)
Output: Hashed confirmed, Server Selected with ANN

Input: IoT data encrypted, private key, and hashed verification value, Utilized offloading servers
ECC-PQC Decryption:
for *each hybrid server* H **do**
  Obtain k_{PQC_i-1} for decryption
  Decrypt encrypted data: $x_i = d(x_i, k_{PQC_{i-1}})$ ▷ Based on Eq. (25)
end for
Output: Decrypted data, Data Offloaded for : (process, storage, and analysis)

to confront evolving threats and align with emerging technological paradigms. Future considerations encompass periodic security audits, updates to security policies under regulatory standards, and the integration of cutting-edge security technologies. The meticulous implementation of these strategies reinforces the security of offloaded data and establishes a resilient foundation. This foundation is integral for authorized entities, facilitated by ANN algorithms, to conduct analytics operations securely. This proactive approach, outlined in this research, underscores our commitment to upholding the highest data security standards in the intricate landscape of smart home technology.

Hybrid computing architecture

We developed a comprehensive and adaptive computing architecture that optimizes the processing of IoT data in smart home systems. This hybrid computing architecture combines the strengths of edge, fog, and cloud computing to create a resilient and efficient framework for IoT big data analytics in smart homes. This approach optimizes data processing, reduces latency, and balances real-time responsiveness.

Dynamic offloading in edge, fog, and cloud with artificial neural network algorithm

Dynamic offloading optimizes system performance across Edge, Fog, and Cloud computing environments. Each layer in this hierarchical architecture, including Edge devices, Fog nodes, and Cloud servers, requires efficient workload distribution to ensure optimal resource utilization, minimize response times, and adapt to varying workloads (*Khan et al., 2023*; *Hossain et al., 2022*; *Joseph, Kwak & Iosifidis, 2019*; *Hoa et al., 2023*; *Mu & Zhong, 2020*; *Yang, Lee & Huang, 2022*; *Bajaj, Sharma & Singh, 2022*).

The problem can be formally represented as follows:

$$R = \{R_1, R_2, \ldots, R_{sen}, \ldots, R_n\} \tag{26}$$

In Eq. (26), R represents the set of input observations that includes user goals, system states, and environmental conditions. The user goals or requests may include data storage, processing, and analysis.

The constraints for incoming requests are represented by R_1, R_2, and R_n in Eq. (26). In particular, the usual approach is to offload the requests to the cloud, even if they can be handled locally or on remote servers, due to the sensitivity indicated by the constraint R_{sen}.

The formulation of R_{sen} within our equation framework directly results from the demands or requirements of the devices within our computational ecosystem. Specifically, R_{sen} encapsulates the conditions or constraints necessitating offloading computational tasks to cloud infrastructure. This decision is driven by the inherent needs or capabilities of the devices involved in our system. Our methodology entails a rigorous analysis of R_{sen} to discern the underlying factors prompting the offloading of tasks to the cloud. By meticulously examining the demands and capabilities of the devices within our computational environment, we gain valuable insights into the rationale behind offloading decisions. Through this comprehensive analysis, our methodology aims to develop robust offloading strategies that effectively leverage cloud resources to meet the demands of the devices while optimizing overall system performance. By aligning offloading decisions with device requirements, our approach ensures efficient resource utilization and enhances the scalability and responsiveness of the system. In essence, the formulation and analysis of R_{sen} within our methodology serve as foundational elements in developing dynamic offloading strategies tailored to the specific demands and needs of the devices within our computational ecosystem.

$$SMD's = \{S_1, S_2, \ldots, S_N\} \tag{27}$$

In Eq. (27), SMD represents a set of smart devices that includes user edge devices such as mobile devices, wearable devices, and IoT devices. While 'S' represents a smart device symbol.

In our research work, we present a comprehensive framework detailing the intricate process by which a smart home system seamlessly integrates with an Edge-Fog-Cloud (EFC) offloading structure. The operational sequence initiates with data generation at edge devices, encompassing sensors, cameras, and smart appliances within the home environment. Subsequently, edge servers strategically located within or near the home undertake time-sensitive computations, facilitating real-time analysis of sensor data, immediate control responses, and local automation logic. The paradigm then transitions to fog computing, where local nodes aggregate data from diverse edge devices, effectively reducing the volume of data requiring transmission to the cloud. Edge-to-fog communication channels, employing wireless protocols such as Wi-Fi, facilitate the seamless exchange of information between edge devices and nearby fog nodes. Fog nodes, acting as intermediate layers, execute additional processing and decision-making tasks based on the aggregated data. Tasks that are non-time-sensitive or necessitate in-depth analysis are subsequently offloaded to the cloud for extensive computation. The cloud computing layer, hosted on centralized servers, engages in advanced analytics, machine learning, and long-term data storage. This analytical prowess encompasses trend analysis, energy consumption predictions, and personalized recommendations tailored to user preferences.

The communication network is bidirectional, encompassing various layers of the EFC architecture. Cloud servers relay insights and instructions back to fog nodes, ensuring that the local fog layer is continually updated with the latest global analytics. Similarly, fog nodes communicate refined instructions to edge devices, enabling localized processing that adapts dynamically based on insights gleaned from the entire system. User interaction is facilitated through interfaces such as mobile apps or voice assistants, with local responses to user commands efficiently handled at the edge or fog layer, ensuring rapid and responsive actions. A critical aspect of our research focuses on the dynamic offloading decision-making process, wherein intelligent algorithms ANN Dynamic Offloading (*Khan et al., 2024*) across the edge, fog, and cloud layers collaborate to optimize task distribution based on real-time requirements, system loads, and evolving user preferences.

Our research comprehensively explains the collaborative interplay between edge, fog, and cloud components in a smart home system. This Edge-Fog-Cloud offloading structure not only enhances system responsiveness but also optimizes resource utilization, offering a scalable and intelligent automation framework that aligns with the specific requirements of smart home environments.

Let W represent the total workload, T the response time, and N the number of resources. The efficiency (E) metric for dynamic workload allocation can be expressed as:

$$E = \frac{W}{N \times T}. \tag{28}$$

This formula remains applicable at each layer of the architecture, emphasizing the need for efficiency in workload distribution at the Edge, Fog, and Cloud levels.

Load balancing algorithms are critical in dynamically allocating workloads across Edge, Fog, and Cloud resources. Tailoring algorithms to the characteristics of each layer ensures optimal performance. For instance, at the Edge, algorithms may consider device capabilities, while in the Cloud, they may prioritize data center resources. Predictive modeling becomes especially relevant in Edge and Fog computing, where local decision-making is crucial. Machine learning models predict workloads based on historical data, allowing proactive allocation adjustments before data reaches the Cloud. Auto-scaling policies are instrumental in all layers of the architecture. Edge devices dynamically adjust resources based on local demand, Fog nodes scale horizontally or vertically, and Cloud servers dynamically provision or de-provision virtual machines based on global demand.

$$D_i = \{D_i^L, D_i^E, D_i^F, \ldots, D_i^C\}. \tag{29}$$

In Eq. (29), D_i^L represents local execution, D_i^E represents execution on edge servers, D_i^F represents execution on Fog servers, and D_i^C represents execution on cloud servers if it passes the condition. The decision-making process aims to reduce overall latency and energy consumption and consider cost considerations. To evaluate energy usage E and calculation delay D under the decision D_i, the following equations are used:

$$E = E(D_i^E) \tag{30}$$

$$D = D(D_i^E). \tag{31}$$

The optimization problem can be defined as follows:

$$P_o = \min(D, E) \tag{32}$$

$$\{D_i^L, D_i^E, D_i^F, \ldots, D_i^C\} \in [0, 1]. \tag{33}$$

When applying ANN to P_o, the offloading issues often have a time complexity on the order of $O(e \cdot n \cdot p \cdot m^{nl_i})$. The ANN Dynamic Offloading Algorithm (*Khan et al., 2024*) is designed to determine the final offloading decision based on the available input data.

Optimization objectives:

- Minimizing response times is a shared objective across Edge, Fog, and Cloud. Efficient workload allocation ensures that tasks are processed swiftly at the Edge when the latency is critical and workloads seamlessly transition to the Fog and Cloud when scalability and computational power are paramount.
- Each layer aims to optimize resource utilization. Edge devices seek to utilize local processing power efficiently, Fog nodes strive to use intermediate computational resources effectively, and Cloud servers focus on efficient data center resource utilization.
- Adaptability to changing workloads is a key criterion. Dynamic workload allocation strategies should allow Edge, Fog, and Cloud components to scale up or down based on fluctuations in demand, ensuring that the entire architecture remains responsive.

Algorithm 4: ANN dynamic offloading algorithm

Data:
Input data set = (x,y)
Get training data set x, y: $(x, y) \leftarrow$ dataset
Get $(\boldsymbol{x}test, \boldsymbol{y}test)$
Run ANN model
$D_i = D_i^L, D_i^E, D_i^F, ..., D_i^C$
1ex ▷ Based on Eq. (29)
$P_o = \min(D, E)$
1ex ▷ Based on Eq. (32)
Calculate mean μ_k and standard deviation σ_k of each feature k in x
Calculate prior probability $P(y = c)$ for each class c
for $k = 1$ *to* K **do**
Calculate class-conditional mean $\mu_{k,c}$ and standard deviation $\sigma_{k,c}$ for feature k in class c
for $c = 1$ *to* C **do**
Calculate $P(x_k|\mu_{k,c}, \sigma_{k,c})$ using Gaussian probability density function
end for
end for
Get observation sequence of the training data set:

$$x_1, x_2, \ldots, x_n :$$

x_i is a sequence of features observations :
Predict the parameters:

$$\hat{\theta} = \arg\max_{\theta \in \Theta} P(X_{1:T}, Y_{1:T}|\theta)$$

Create a hybrid model based on the results:

$$\hat{y}hybrid = w_1 \hat{y}arima + w_2 \hat{y}lstm$$

Final Offloading Decision Scenario:

$$\begin{cases} \text{Remote offloading;} \\ \text{otherwise;} \\ \text{Local offloading;} \end{cases}$$

$(D_i^L, D_i^E, D_i^F, ...D_i^C \epsilon [0, 1])$
▷ Based on Eq. (33)
if *The predict value* = 0 **then**
$D_i = D_i^L \epsilon [0]$ ▷ Decision is Local
else
$D_i = , D_i^E, D_i^F, D_i^C \epsilon [1]$
▷ Decision is Edge, Fog, and Cloud server
end if-else
Result:
Run ANN modeling with training data
Predict delay latency
Predict energy consumption
Predict bandwidth with test data
Send it as the input request
Final Offloading Decision: Local / Remote

Our research significantly advances dynamic workload allocation strategies by addressing the nuances of Edge, Fog, and Cloud computing. By tailoring algorithms, embracing predictive modeling, and optimizing resource usage, Our work enhances the efficiency of workload allocation in this multi-layered computing paradigm. Figure 3 illustrates our proposed methodology's sequential steps and processes.

RESULTS AND DISCUSSION

This section discusses the results of the proposed approach in detail, taking into account several performance measures. These measures include the following: analysis of the system's scalability, security evaluation, reliability, anomaly detection, malicious and

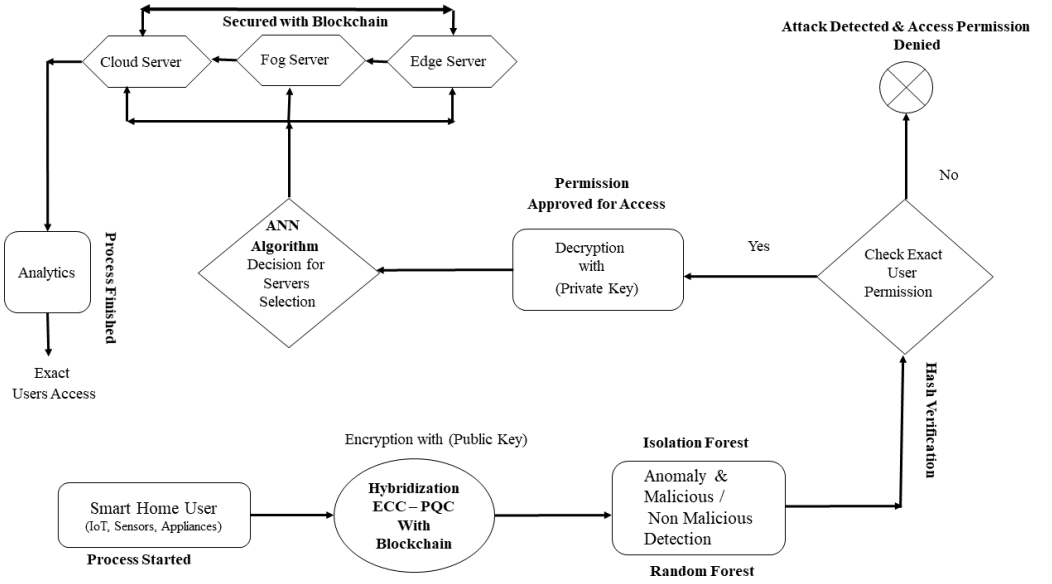

Figure 3 **Framework process flow diagram.**

Full-size DOI: 10.7717/peerjcs.2211/fig-3

non-malicious analysis with all classification algorithms for accuracy, and lastly, dynamic offloading decisions, which greatly increase latency throughput and bandwidth. We also provided an extensive overview of comparative studies comparing our suggested TIBDA architecture with well-known state-of-the-art security techniques. This combined presentation offers a comprehensive analysis of the proposed TIBDAs compared to existing approaches.

System configuration for experiment and performance testing analysis

A dedicated computing system is configured with a Core i9 (CPU) 13th generation, 12GB of RAM, and a 1TB SSD storage device with an Intel Iris Xe (GPU) card to simulate the computing capabilities. This configuration is optimized to unravel TIBDA's operational efficiency. Performance testing involves assessing the system's processing speed, responsiveness, and resource utilization. The system completes its tasks in just a few minutes, showcasing its capability to handle data from multiple IoT devices in smart homes seamlessly. This performance insight underscores the efficiency of a hybrid model, paving the way for enhanced scalability and responsiveness. Comprehensive details regarding the simulation setup and parameters can be found in Table 2.

We employed a network traffic generator using Java programming language within the Eclipse integrated development environment (IDE). This combination allowed us to leverage the capabilities of CloudSim (to simulate the behavior of hybrid servers), a cloud computing simulation toolkit written in Java, for generating, analyzing, and testing the performance of these servers in a simulated cloud environment for network traffic. Eclipse provided a user-friendly platform for writing, debugging, and executing simulation code, facilitating the research on performance testing and analysis of network systems. To

Table 2 Simulation setup and parameters.

Simulation parameters	Description
Operating system	Core i9 (CPU) 13th generation
RAM	12 GB
Hard drive	1TB SSD
GPU	Intel Iris Xe (GPU) card
Simulation platform	Eclipse, Jupyter, Attack simulator (BS, Snort, and Zee), Ethereum, QKG
Types of servers	Edge, Fog, Cloud
Latency (RTT/ms)	0–100
Edge devices	Raspberry Pi
Devices energy (V)	1.5–5.0
IoT Devices	Sensors, Actuators, Cameras
Dataset	Smart home, NAB
Programming language	Java, Python
CloudSim	Test (ANN) and simulate hybrid computing environment
Pandas	Data manipulation and analysis
Numpy	Array operations and mathematical functions
Matplotlib.pyplot	Visualizing data through plots and charts
Scikit-learn	Prominent machine learning library
Training and testing	70%–30% for ANN model evaluation and cryptosystem efficiency
Offloading decision scenarios	Local' and 'Remote' for offloading computational tasks

simulate the behavior of hybrid servers, we utilize the CloudSim package in Java, allowing us to model. Edge devices act as bridges between IoT devices and the cloud, facilitating connectivity, analytics, and data collection. Intelligent sensors, cameras, and actuators are among the numerous smart devices that are used by IoT devices to collect data and facilitate remote control.

Our study utilizes an attack simulator for security analysis, evaluating the efficacy of Quantum Cryptography in providing safe data transfer *via* encryption and key distribution mechanisms. Additionally, we use a Blockchain System, which utilizes a distributed and impenetrable ledger system corresponding to Ethereum, to monitor and validate transactions. Thanks to this thorough strategy, we possess the ability to assess the resistance of our suggested security framework to fictitious cyberattacks. We incorporate tools such as Burp Suite, Snort, and Zeek (formerly known as Bro) into our simulation environment to create a representative testbed for assessing the performance and effectiveness of our system in real-world scenarios. By identifying vulnerabilities, validating security controls, and enhancing the resilience of our system, we aim to safeguard sensitive data and transactions effectively.

We developed our proposed system by using advanced tools and the latest machine-learning libraries in Python. Specifically, we implemented algorithms such as Isolation Forest for anomaly detection and Forestry Random for distinguishing between malicious and non-malicious data. To accomplish this, we utilized the open-source platform Jupyter

Notebook to write source code for data preprocessing, modeling, and training and to evaluate the visual performance of our model.

We leveraged a suite of essential Python libraries to develop and evaluate our anomaly detection model. The 'pandas' library facilitated seamless data manipulation and analysis, while 'numpy' provided fundamental support for array operations and mathematical functions. Visualizing data through plots and charts was made possible by the 'matplotlib.pyplot' library. Additionally, we capitalized on the capabilities of 'scikit-learn,' a prominent machine learning library, utilizing various submodules such as 'IsolationForest,' 'LocalOutlierFactor', 'OneClassSVM', and 'EllipticEnvelope' for anomaly detection in conjunction with 'train test split' for dataset partitioning.

For feature selection in malicious/non-malicious classification using RF, KNN, SVM, LDA, and QDA, we employed a variety of techniques including mutual information, recursive feature elimination, feature importance from tree-based models, L1 regularization, correlation-based selection, and model-based feature selection. This comprehensive integration of diverse libraries empowered us to investigate anomaly detection methodologies and classification purposes thoroughly, yielding valuable insights into our research domain.

In this research, we evaluated the efficacy of our proposed security model against several critical threat criteria in hybrid computing with IoT platforms using a wide range of datasets, including Smart Home and NAB, in our experimental setup.

- The dataset for smart homes comprises 15 columns and approximately 49,000 rows, encompassing a fictitious four-year period (2020–2023) in a real-world smart home situated in Xi'an, China. This dataset is stored as a CSV file named 'smart home dataset.csv.' We utilized this dataset to analyze energy consumption trends, behavior patterns in smart homes, and scenarios for identifying insights into energy consumption trends, smart home behavior, and decision-making regarding the delegation of computing tasks.
- An important component of our study involved the Numenta Anomaly Benchmark (NAB) dataset (*Benchmark, 2020*), which comprises real-world datasets designed specifically for evaluating anomaly detection techniques in streaming data. We used the NAB dataset as a benchmark to assess the performance of our anomaly detection algorithm, particularly in multi-user scenarios involving multiple IoT devices, users, and transactions. Through our evaluation, we examined the robustness, accuracy, and performance of our algorithm under various circumstances, confirming its effectiveness in practical contexts with the assistance of the NAB dataset.

In our study, the determination of simulation parameters is conducted through a meticulous process aimed at ensuring the robustness and relevance of our experimental setup. Firstly, we aligned our parameter selection with the objectives of our research, focusing on aspects such as system configuration, simulation platforms, security analysis, machine learning algorithms, and datasets. This alignment ensured that our simulations accurately reflected the goals and scope of our investigation. Secondly, we prioritized realism

and representation, closely mirroring real-world computing environments commonly found in IoT smart home setups. By doing so, we aimed to capture the complexities and constraints inherent in practical deployment scenarios, enhancing the validity of our findings. Thirdly, we employed a methodologically rigorous approach, drawing on established best practices and consulting domain experts. This rigorous process enabled us to identify and select parameters that would yield meaningful insights and reliable results. Fourthly, we selected simulation tools and platforms based on their suitability for our research objectives, prioritizing robust functionality and ease of use. By leveraging tools such as Eclipse, Jupyter, CloudSim, and attack simulators, we ensured that our experimental framework was well-equipped to meet the demands of our investigation. Lastly, to ensure comprehensive analysis, we utilized a diverse range of datasets, including synthetic and real-world data sources relevant to IoT smart home applications. This diversity allowed us to explore various scenarios and evaluate the performance of our framework under different conditions. Overall, our meticulous process of determining simulation parameters contributed to the validity and robustness of our findings, underscoring our commitment to clarity and transparency in research.

TIBDA response time and scalability analysis

Two key measures are used to evaluate the system architecture's scalability and performance: response time and user count. The initial statistic indicates how well the system can handle growing loads, while the alternative shows how well it can respond quickly. To guarantee that the system maintains acceptable performance levels under various loads, a crucial component for real-world usability and effectiveness, it is necessary to evaluate reaction times across a range of user counts. Additionally to the user measurement, the number of devices, transactions, or requests can also affect the system's scalability and performance. These variables can be shown on the X-axis with the specific use case and system requirements taken into account.

Response time is computed using the following equation, which takes into account several factors that can expressed:

$$\text{Response Time} = \frac{\text{Total Processing Time}}{\text{Number of Transactions or Requests}} \tag{34}$$

where:

- Total processing time: The cumulative time taken by the system to process all transactions or requests.
- Number of transactions or requests: The total count of transactions or requests performed within the specified time frame.

This equation provides a fundamental framework for quantifying response time, enabling comprehensive performance evaluation across different usage scenarios.

The research study employs a comprehensive analysis of response times across various systems, aiming to discern their performance under different user loads represented in Fig. 4. The investigation encompasses systems such as TIBDA, SAS-Cloud, SHCEF, AILBSM, MHE-IS-CPM, and PA, each assessed across user groups comprising 25, 50,

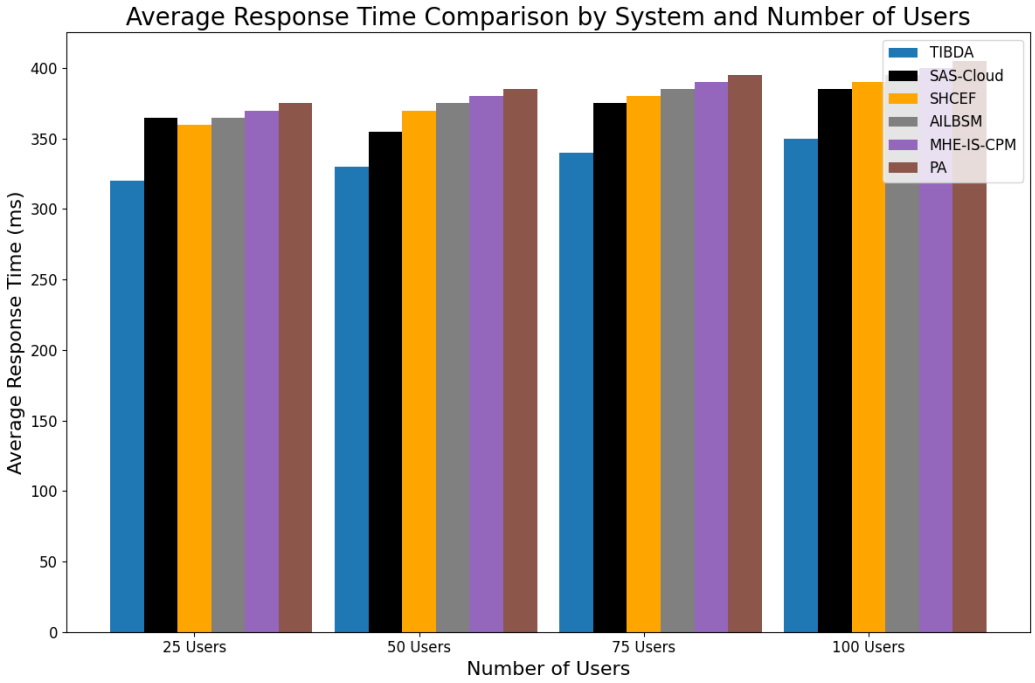

Figure 4 **Average response time (ms) and numbers of users comparison.**
Full-size DOI: 10.7717/peerjcs.2211/fig-4

75, and 100 individuals. Response time data, crucial for evaluating system efficiency, is meticulously collected and processed. A clear depiction of performance trends emerges through averaging response times for each system across user groups.

The analysis reveals discernible differences in the performance of the various systems under examination. Across different user loads, some systems consistently exhibit lower average response times than others. Specifically, the TIBDA system consistently demonstrates the lowest response times across all user groups, indicating superior responsiveness.

Conversely, other systems such as SAS-Cloud, SHCEF, AILBSM, MHE-IS-CPM, and PA exhibit slightly higher average response times across user groups, suggesting comparatively lower efficiency in handling user requests.

Therefore, based on the collected data and analysis, the TIBDA system emerges as the model that consistently delivers better results regarding response time performance across varying user loads. This finding underscores the importance of system selection and optimization in ensuring optimal performance and user satisfaction in real-world applications.

Figure 5 depicts our comprehensive analysis comparing the average response times of various systems under different numbers of devices. The response times, measured in milliseconds, were adjusted to reflect real-world scenarios for each system.

The TIBDA system consistently demonstrated lower response times across all device configurations, ranging from 250 to 1,000. Specifically, for 250 devices, the average response

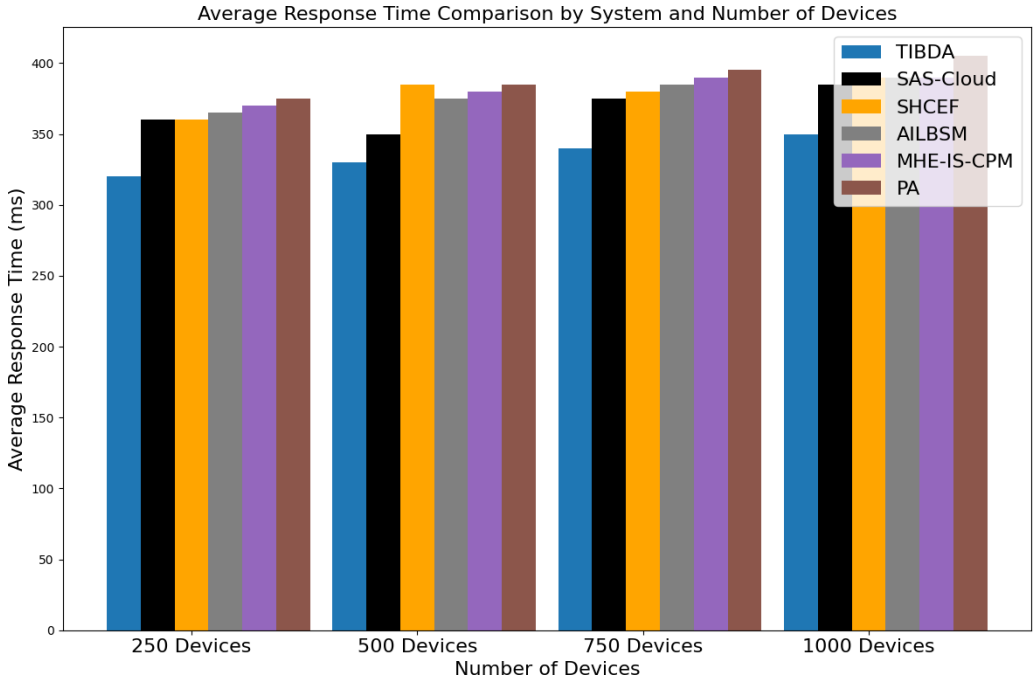

Figure 5 **Average response time (ms) and numbers of devices comparison.**
Full-size DOI: 10.7717/peerjcs.2211/fig-5

time was approximately 320 ms, and it gradually increased to around 340 ms for 1,000 devices.

In contrast, the SAS-Cloud system exhibited slightly higher response times than TIBDA but remained competitive. For instance, with 250 devices, the average response time was about 360 ms, increasing to around 375 ms for 1,000 devices. The SHCEF system's performance varied, showing fluctuations in response times across different device configurations. Initially, with 250 devices, the average response time was around 360 ms, but it improved to approximately 380 ms with 1,000 devices. Similarly, AILBSM and MHE-IS-CPM systems demonstrated response times comparable to SAS-Cloud, with slight variations across device configurations. Both systems exhibited average response times ranging from 370 to 385 ms for 250 to 1,000 devices. Lastly, the PA system consistently had higher response times than other systems across all device configurations. Starting from approximately 370 ms with 250 devices, the average response time increased to around 395 ms with 1,000 devices.

Overall, TIBDA outperformed other systems regarding response time across various device configurations, making it a promising choice for scenarios requiring low-latency responses.

As shown in Fig. 6, we investigated the performance of various systems regarding average response times under different transaction loads. The response times, measured in milliseconds, were adjusted to simulate realistic transaction scenarios for each system.

The TIBDA system consistently demonstrated lower response times across all transaction loads, ranging from 125 to 500 transactions. For instance, with 125 transactions, the average

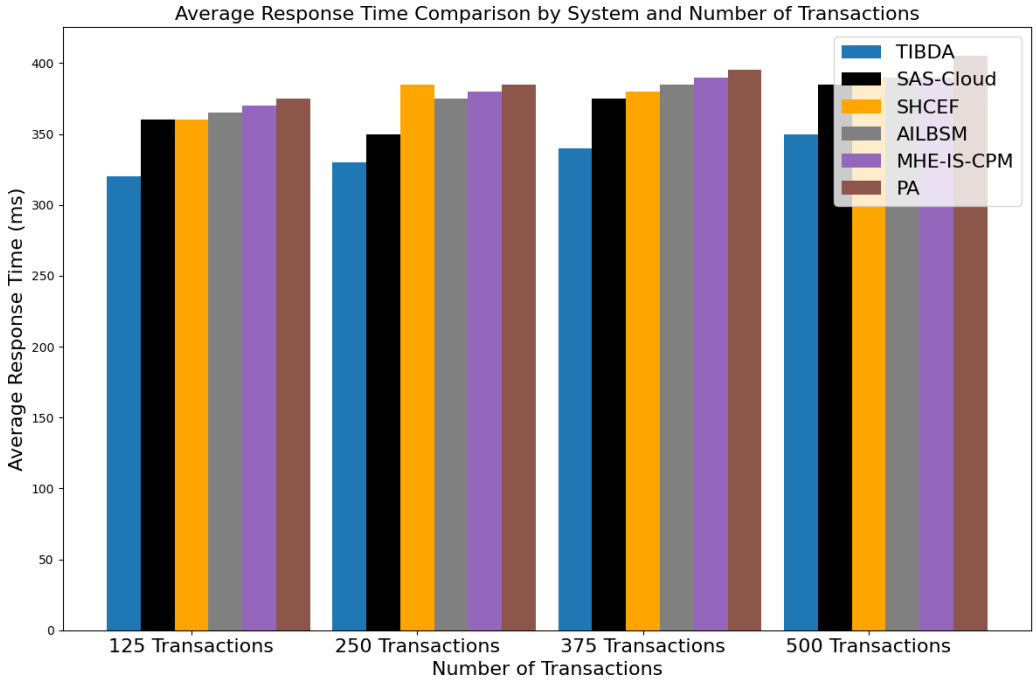

Figure 6 **Average response time (ms) and numbers of devices transactions comparison.**
Full-size DOI: 10.7717/peerjcs.2211/fig-6

response time was approximately 330 ms, and it gradually increased to around 340 ms with 500 transactions.

The SAS-Cloud system exhibited slightly higher response times than TIBDA but remained competitive. For example, with 125 transactions, the average response time was about 355 ms, increasing to approximately 375 ms with 500 transactions. The SHCEF system's performance showed fluctuations in response times across different transaction loads. Initially, with 125 transactions, the average response time was around 355 ms, but it improved to approximately 380 ms with 500 transactions. Similarly, both AILBSM and MHE-IS-CPM systems demonstrated response times comparable to SAS-Cloud, with slight variations across transaction loads. Both systems exhibited average response times ranging from 365 to 385 ms for 125 to 500 transactions. On the other hand, the PA system consistently had higher response times than other systems across all transaction loads. Starting from approximately 385 ms with 125 transactions, the average response time increased to around 400 ms with 500 transactions.

Overall, TIBDA consistently outperformed other systems regarding response time across various transaction loads, making it a promising choice for scenarios requiring low-latency responses.

Analytical reasoning:

- Efficient resource management:

TIBDA's lower response times are due to its optimized resource management strategies. The system efficiently allocates and utilizes resources, ensuring that processing times remain minimal even under heavy loads.

- Scalability:

The system's architecture is designed to scale seamlessly, allowing it to manage more users, devices, and transactions without significant performance degradation. This scalability is a key factor in TIBDA's superior performance compared to other models.

TIBDA security analysis

Evaluating a system's security is extensive and involves identifying its vulnerabilities and the effectiveness of the security measures to mitigate possible attacks. Maintaining the system's security design depends on several essential elements, such as secure communication protocols, using cryptographic algorithms, and managing cryptographic keys. These key elements collectively contribute to fortifying the system against malicious attacks and unauthorized access attempts, enhancing its resilience and reliability in the face of evolving cybersecurity challenges.

An assessment of a system's security (S) includes an evaluation of its vulnerabilities (V) and the effectiveness of the security measures in place (M) in dealing with potential threats (T). It could be accomplished to express this in the following form:

$$S = f(V, M, T) \tag{35}$$

where:

- S represents the system's security level.
- V represents the vulnerabilities inherent in the system.
- M represents the security measures implemented within the system.
- T represents the potential threats that the system may encounter.
- f denotes the function that quantifies the relationship between vulnerabilities, security measures, and threats in determining the overall security level of the system.

In Fig. 7, our analysis aims to evaluate and compare the security performance of various models using the AUC metric across different user levels. As mentioned in Fig. 7, the performance of different security models in terms of their AUC values at varying user levels. The vulnerability rate, representing the number of users, is plotted on the x-axis, while the degree of protection, measured by the AUC value, is described on the y-axis. Higher AUC values indicate better security performance, reflecting the ability of the models to mitigate security threats effectively.

TIBDA consistently exhibits the highest AUC values across all user levels, indicating superior security performance. This suggests that TIBDA effectively safeguards against security threats, making it a robust choice for security implementations. While SAS-Cloud demonstrates competitive AUC values, particularly at lower user levels, its performance slightly lags behind TIBDA. Nevertheless, it remains a viable option for security solutions. SHCEF shows moderate AUC values across different user levels, indicating its capability to

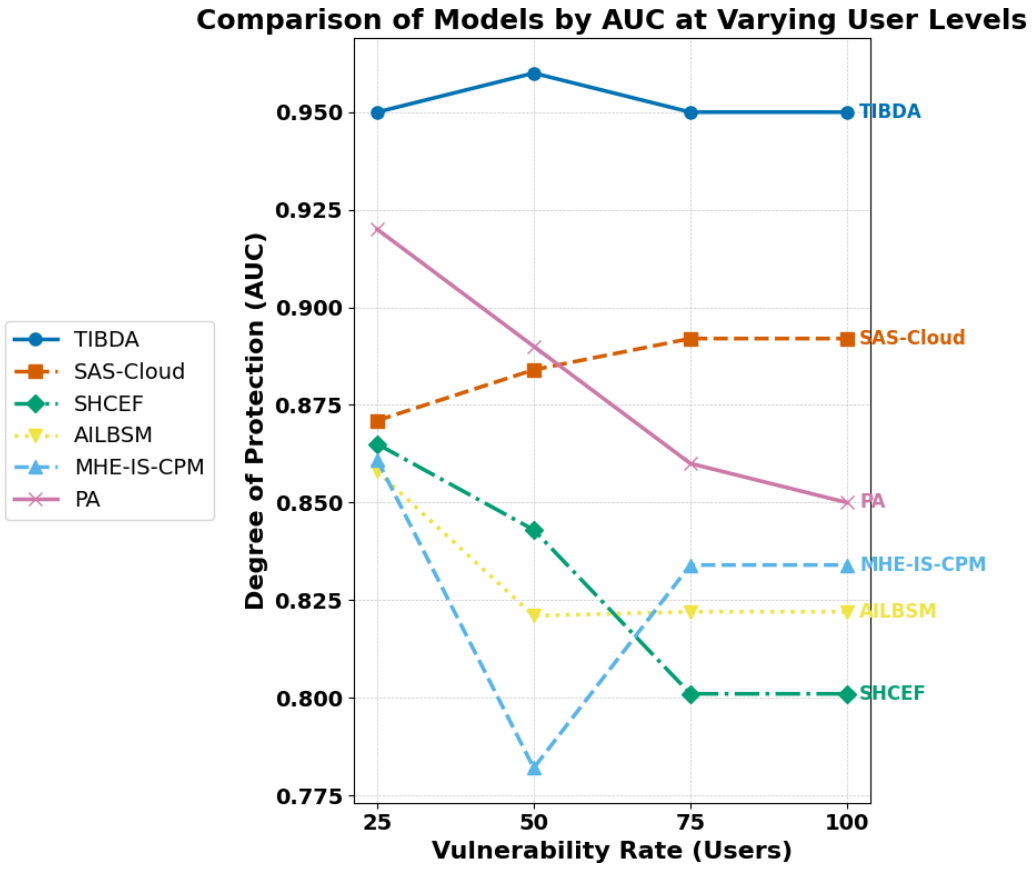

Figure 7 **A comparison of the security utilizing the AUC at different user levels.**
Full-size DOI: 10.7717/peerjcs.2211/fig-7

provide reasonable security protection. However, its performance may not match TIBDA's in higher user scenarios.

ALIBSM exhibits comparable AUC values to SHCEF, suggesting similar security performance. While it may offer adequate protection, it may not be as effective as TIBDA in mitigating security threats in more challenging environments. MHE-IS-CPM consistently demonstrates lower AUC values than other models, indicating relatively weaker security performance. While it may provide some level of protection, it appears less robust compared to TIBDA and other models. PA displays varied AUC values across different user levels, with its performance declining at higher user levels. This indicates that while PA may offer decent security protection, it may struggle to maintain effectiveness in more extensive user settings.

Analytical reasoning:

- Advanced security measures:

TIBDA employs robust cryptographic algorithms and secure communication protocols, significantly reducing vulnerabilities.

- Proactive threat management:

The system's design includes proactive threat detection and mitigation mechanisms, which enhance its resilience against attacks.

TIBDA trustworthiness or reliability analysis

The term "trustworthiness" alludes to the capability of a framework to execute its intended duties across time reliably. In the setting of information security, "trust ability" is an indicator of trustworthiness and is frequently articulated as a proportion reflecting the duration of inactivity encountered by the framework because of security occurrences or assaults. Consequently, trustworthiness is the pace at which a framework persists in operating properly without failure over time.

The trustworthiness of a system is quantified using the following equation:

$$\text{Trustworthiness} = \frac{\text{Total uptime}}{\text{Total time}} \times 100\% \tag{36}$$

where:

- **Total uptime** represents the duration during which the system operates without failure. It captures when the system functions as intended, fulfilling its tasks consistently.
- **Total time** represents the entire duration under consideration, including uptime and downtime. It accounts for all periods, whether the system is operational or experiencing failures.

By calculating the ratio of total uptime to total time and multiplying the result by 100%, we obtain a percentage value that signifies the system's trustworthiness. This percentage indicates the proportion of time the system spends functioning reliably compared to the overall duration observed.

In essence, this equation provides a structured method for evaluating a system's reliability by measuring its performance against total time, ultimately developing a numerical representation of its trustworthiness.

In Fig. 8, each point represents the performance of a specific approach in terms of system uptime and downtime. Different techniques were considered in the comparison, including TIBDA, SAS-Cloud, SHCEF, AILBSM, MHE-IS-CPM, and PA. Each data point in the scatter plot corresponds to one of the approaches and represents a combination of downtime and uptime percentages observed for that approach. The color of each data point corresponds to the specific approach it describes.

The x-axis represents the downtime experienced by each approach, while the y-axis represents the uptime percentage. Observing the graph, we can see some points clustered toward the lower left corner, indicating approaches with higher downtime and lower uptime. Conversely, points closer to the upper right corner represent approaches with lower downtime and higher uptime.

To determine which approach performed better, we typically look for points closer to the upper right corner, which indicates approaches with higher uptime and lower downtime. In this case, the "TIBDA" approach tends to have more points clustered towards the upper right corner compared to the other approaches, suggesting that it performs better regarding reliability.

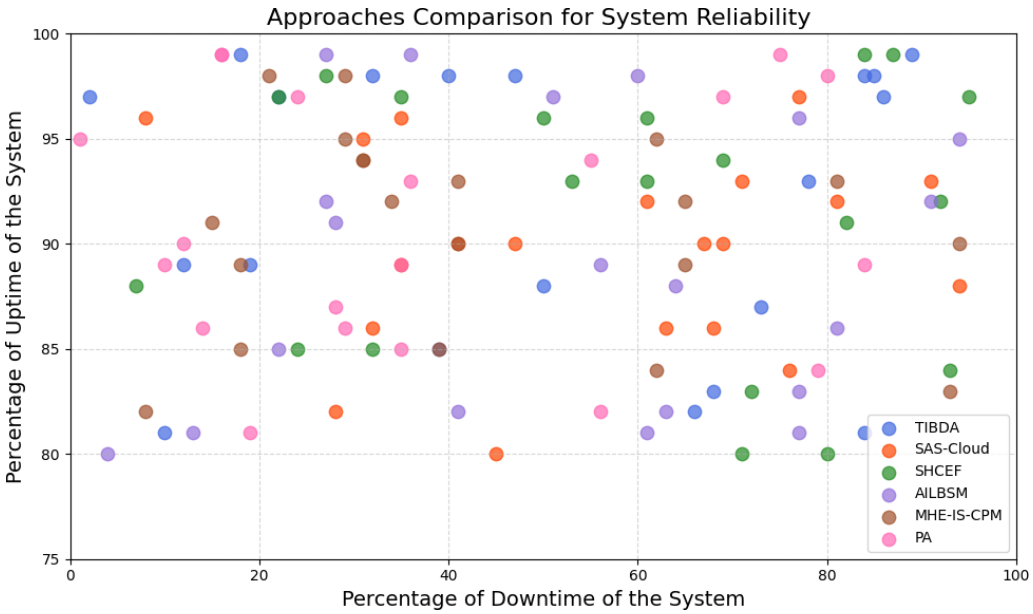

Figure 8 **Approaches comparison analysis for system reliability.**
Full-size DOI: 10.7717/peerjcs.2211/fig-8

The comparative analysis in the figure illustrates the performance of different system reliability approaches, particularly on TIBDA. The proposed "TIBDA" exhibits superior performance owing to its accurate design and implementation, incorporating advanced security measures tailored to effectively combat various types of attacks. Additionally, TIBDA boasts optimized resource allocation strategies, ensuring efficient utilization of system resources, and features fast recovery mechanisms that enable swift restoration of system functionality in the face of downtime events. The clustering of data points towards the upper left corner of the plot highlights TIBDA's ability to achieve higher uptime percentages while minimizing downtime, underscoring its robustness and effectiveness in maintaining system availability and resilience compared to other approaches.

Analytical reasoning

- Robust design:

TIBDA's architecture ensures continuous operation and swift recovery from failures. Its low complexity further contributes to its high reliability by minimizing the likelihood of errors and operational disruptions.

- Effective resource allocation:

Optimized resource allocation strategies ensure the system remains functional and reliable over extended periods.

Table 3 comprehensively compares various security approaches, each addressing distinct concerns in safeguarding systems against potential threats. The methodologies employed by these approaches vary significantly, ranging from cloud-based solutions like SAS-Cloud to frameworks such as SHCEF and protocols like MHE-IS-CPMT. These methodologies

Table 3 Overall comparative results analysis of TIBDA with existing security concerns approaches.

Reference	Methodology	Complexity	Reliability
Irshad & Chaudhry (2021)	SAS-Cloud	High	Moderate
Sharma et al. (2015)	SHCEF framework	High	Moderate
Jalasri & Lakshmanan (2023)	Clustering and cryptography	High	Low
Unal et al. (2021)	Safe Cloud Storage System (SCSS)	Moderate	High
Selvarajan et al. (2023)	AILBSM model	Moderate	Moderate
Uppuluri & Lakshmeeswari (2023)	MHE-IS-CPMT protocol	High	Low
Ahmad, Mehfuz & Beg (2023)	Hybrid cryptographic Methodology	High	High
Sharma et al. (2023a)	Application with Blockchain	High	Moderate
Proposed approach	**TIBDA framework**	**Very low**	**High**

encompass various techniques, algorithms, and strategies to address specific security challenges.

One crucial aspect evaluated in the table is the complexity of each approach. Complexity refers to the intricacy and sophistication of the system's design and implementation. Higher complexity often implies more intricate architectures or mechanisms, which may offer advanced features and increase the likelihood of encountering challenges during operation. For instance, approaches like SHCEF and the MHE-IS-CPMT protocol exhibit high complexity, indicating intricate designs that may pose operational challenges.

On the other hand, reliability emerges as a key consideration when assessing the effectiveness of these security approaches. Reliability is the system's capacity to carry out its intended tasks without error or disruption continuously. An approach's ability to withstand different threats and sustain operational continuity is reflected in its reliability. For instance, while some approaches like SAS-Cloud and AILBSM exhibit moderate reliability, others like the Safe Cloud Storage System (SCSS) demonstrate higher reliability, indicating their resilience to potential threats.

Among the various approaches assessed, the TIBDA framework stands out for its remarkable attributes. Notably, it exhibits significantly low complexity alongside high reliability, setting it apart as a promising solution compared to existing methodologies. Despite its simplicity in design and implementation, the TIBDA framework consistently delivers its intended functionalities, offering robust protection against security threats. This combination of low complexity and high reliability positions the TIBDA framework as a compelling option for addressing security concerns effectively.

Analysis and results of anomalies detection

In our research exploration of anomaly detection algorithms, we applied the Isolation Forest method to discern anomalous patterns within our dataset. The Fig. 9 subsequent visual analysis aimed to provide insights into the algorithm's effectiveness. The generated plots, presented side by side for training and testing data, depict the distribution of anomalies across two specified features. Each point in the scatter plots is color-coded to represent its classification as either a non-anomalous (0) or anomalous (1) data point. The legend on the plots clarifies the color-coding scheme, facilitating a clear interpretation

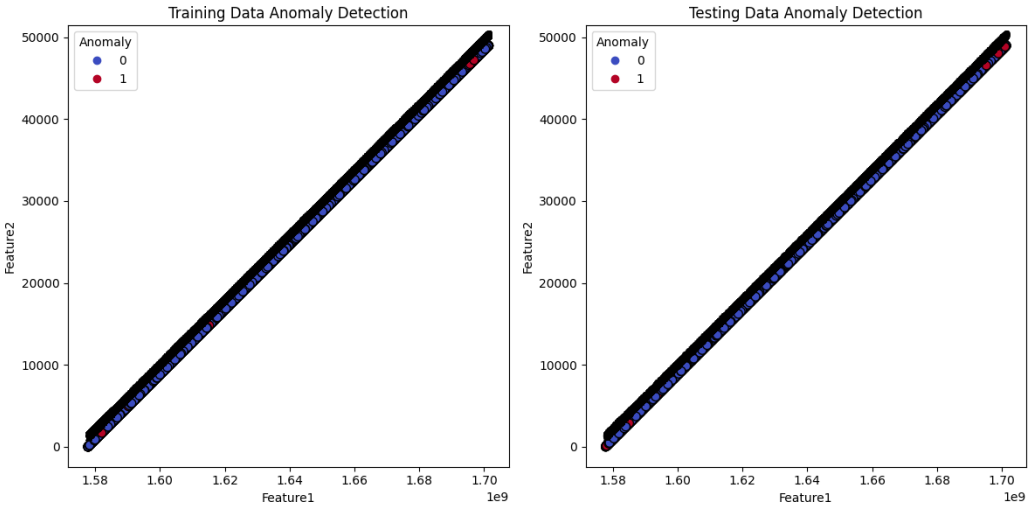

Figure 9 **Analysis and results of anomalies detection.**

Full-size DOI: 10.7717/peerjcs.2211/fig-9

of the results. Additionally, annotations accompany individual points, further aiding in identifying anomalies within the visual representation. This graphical exploration is a valuable component of our research, shedding light on the algorithm's performance in detecting anomalies within the specified feature space.

The effectiveness of the Isolation Forest anomaly detection algorithm is evident through the visual representation and analysis of the generated plots. The algorithm demonstrates a notable capability to isolate and identify anomalous patterns within the dataset, as showcased in both the training and testing data plots.

Performance comparison of anomaly detection algorithms accuracies

In this study, we comprehensively compared four prominent anomaly detection algorithms: isolation forest, local outlier factor, one-class SVM, and elliptic envelope. Figure 10 presents a visual representation of the accuracy achieved by each algorithm in detecting anomalies within our dataset. Each bar corresponds to a specific algorithm, with the accuracy percentage annotated. Notably, the Isolation Forest algorithm outperforms the others, boasting an accuracy of 99.30%. Local Outlier Factor follows with an accuracy of 90.00%, while One-Class SVM and Elliptic Envelope exhibit accuracies of 89.50% and 88.20%, respectively. Figure 10 provides a succinct overview of the comparative performance of these algorithms, laying the groundwork for insightful discussions on their suitability for anomaly detection tasks in our research.

As we delve into the complexity of the Isolation Forest algorithm, we see that its key strength lies in its ability to maintain a linear time complexity. The average-case time complexity for anomaly isolation in an Isolation Forest is $O(t * \log(n))$, where 't' represents the number of trees in the forest and 'n' is the number of data points. The algorithm's simplicity and effectiveness make it an attractive choice for applications demanding real-time processing and rapid identification of anomalies.

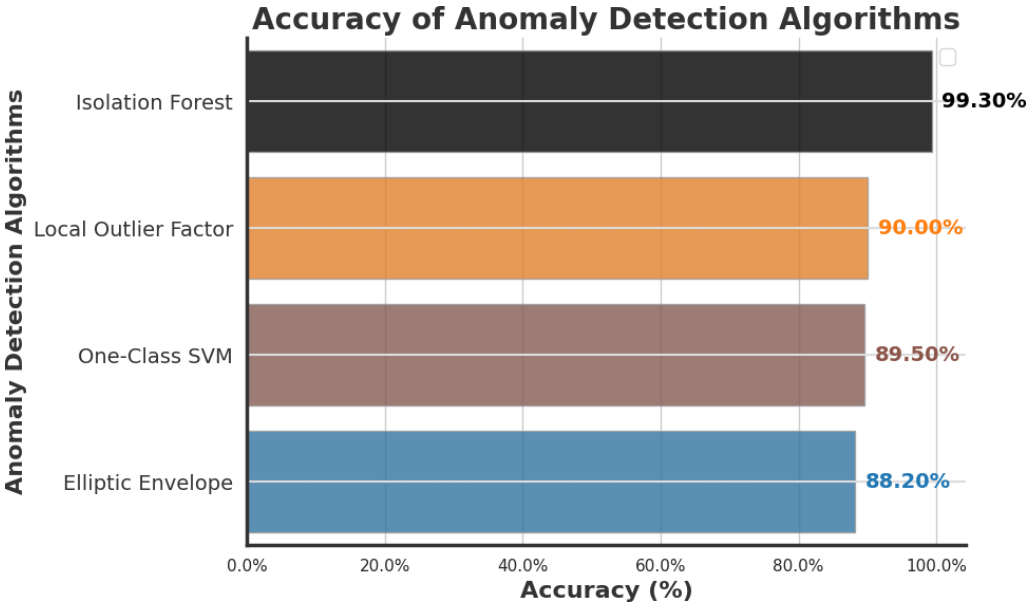

Figure 10 **Comparison of anomaly detection algorithm accuracies.**
Full-size DOI: 10.7717/peerjcs.2211/fig-10

Local Outlier Factor (LOF), One-Class Support Vector Machine (One-Class SVM), and Elliptic Envelope exhibit different computational characteristics. LOF's time complexity is $O(n^2)$, making it suitable for small to medium-sized datasets. One-class SVM's complexity is $O(n^3)$, rendering it less efficient for large datasets but powerful in separating data points in high-dimensional spaces. Meanwhile, the Elliptic Envelope's complexity depends on the underlying algorithm used for covariance estimation, typically ranging from $O(n^2)$ to $O(n^3)$. Despite their varying complexities, each technique offers unique advantages and applicability in anomaly detection across diverse domains.

Comparing these complexities, Isolation Forest's linear time complexity makes it more efficient, especially for large datasets, compared to LOF, One-Class SVM, and potentially Elliptic Envelope. This efficiency contributes to Isolation Forest's suitability for real-time processing and scalable anomaly detection tasks across diverse domains. Therefore, in terms of computational efficiency, Isolation Forest appears to be the most favorable option among the mentioned techniques.

Analysis and results of malicious and non-malicious detection

In this experiment, we assessed the performance of the random forest algorithm in a binary classification scenario within our dataset. The confusion matrix, depicted in Fig. 11, serves as a comprehensive tool to evaluate the classification performance of the random forest algorithm to achieve an accuracy of 94.70%.

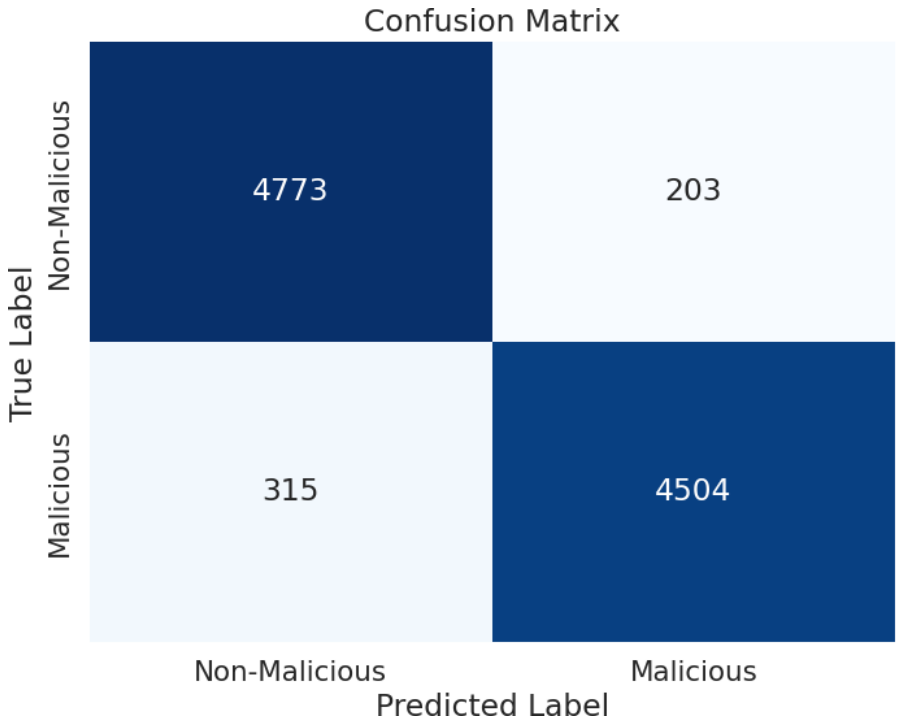

Figure 11 **Confusion matrix analysis for random forest algorithm.**
Full-size DOI: 10.7717/peerjcs.2211/fig-11

The confusion matrix is a statistical metric that provides a detailed breakdown of the classifier's performance. It is structured based on four fundamental features: True Positive (μ), False Positive (θ), True Negative (γ), and False Negative ($).

1. **True positive (μ):**
   - Represents the instances correctly identified as positive by the classifier.
   - In the context of this experiment, μ signifies the count of correctly predicted malicious instances.
2. **False positive (θ):**
   - Denotes instances incorrectly classified as positive by the algorithm.
   - In this scenario, θ indicates the count of non-malicious instances misclassified as malicious.
3. **True negative (γ):**
   - Signifies instances accurately classified as negative by the algorithm.
   - γ represents the count of correctly predicted non-malicious instances.
4. **False negative ($):**
   - Represents instances incorrectly predicted as negative when they are positive.
   - In the present study, ($) indicates the count of malicious instances erroneously classified as non-malicious.

The confusion matrix percentages, presented as integers for clarity, offer insights into the algorithm's strengths and weaknesses. The heatmap in Fig. 11 visually represents the distribution of correct and incorrect predictions.

The diagonal elements of the confusion matrix, corresponding to True Positive (μ) and True Negative (γ), reveal the algorithm's ability to identify both malicious and non-malicious instances correctly. Conversely, off-diagonal elements (θ and ($)) highlight areas where misclassifications occurred.

The results presented in Fig. 11 demonstrate the performance of the random forest algorithm in classifying instances as either non-malicious or malicious. With its detailed breakdown, the confusion matrix offers a comprehensive evaluation of the algorithm's accuracy and provides valuable insights for further refinement and optimization.

The heatmap's color intensity reflects the percentage of instances in each category, providing a nuanced understanding of the algorithm's predictive accuracy. The diverging color palette enhances visualization, emphasizing the distinction between different performance categories.

The true positive, false positive, true negative, and false negative values were found to be 4,773, 203, 315, and 4,504, respectively.

This pattern suggests that the algorithm accurately identified non-malicious and malicious instances in the hypothetical scenario described. The values in the confusion matrix are often expressed as counts or percentages, and the context of a specific application or experiment would determine the meaning and implications of these values.

By leveraging the distinctive characteristics captured by these features, we can evaluate the model's effectiveness through a comprehensive analysis of its performance metrics. The model's accuracy gauges its overall correctness in predicting positive and negative instances. Precision measures the precision of the optimistic predictions, indicating how many predicted positive instances are true positives. On the other hand, recall assesses the model's ability to capture all actual positive instances, highlighting its sensitivity. These performance metrics provide a nuanced understanding of the model's strengths and limitations, facilitating a more thorough assessment of its predictive capabilities.

Performance comparison of classification algorithms accuracies

In this comprehensive evaluation of classification algorithms for detecting malicious and non-malicious activities, we assess the efficacy of various machine learning models. The two classes under consideration are "Malicious" and "Non-Malicious", crucial for security and threat detection applications.

Each algorithm is meticulously scrutinized for its performance in classifying instances belonging to the "Malicious" and "Non-Malicious" categories. The y-axis of the plot signifies the accuracy percentage, presenting two bars for each algorithm representing its accuracy in detecting both "Malicious" and "Non-Malicious" instances.

Notable observations from the analysis:

- Random forest: Emerges as the top performer, exhibiting a remarkable accuracy of 94.7% for both "Malicious" and "Non-Malicious" instances. Its ability to handle complex relationships in the data renders it a robust choice for security applications.

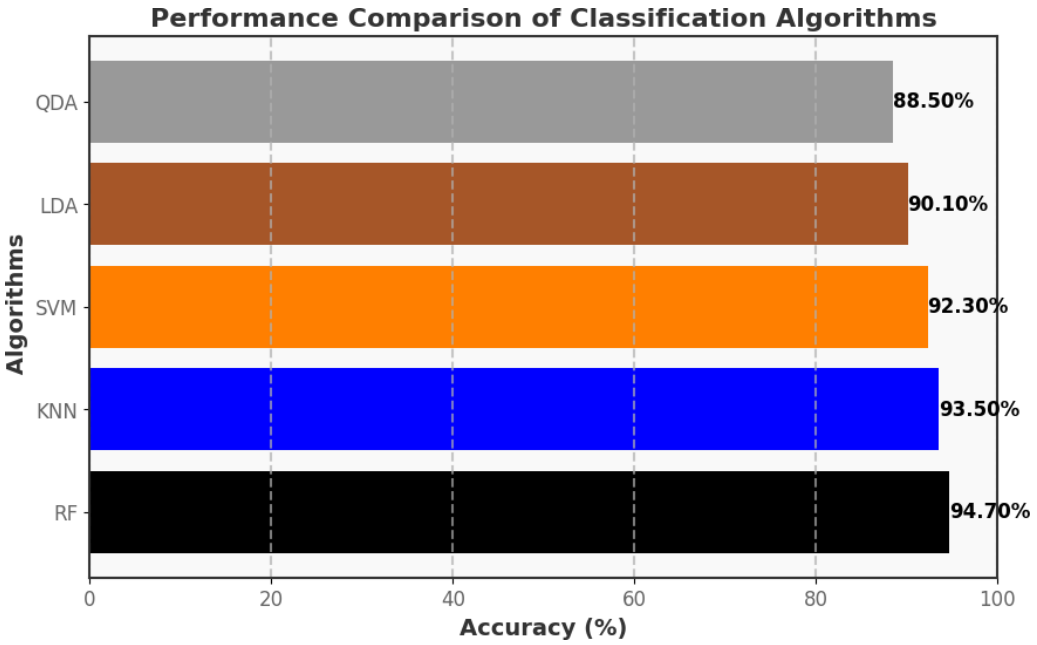

Figure 12 **Performance comparison of malicious and non-malicious classification.**
Full-size DOI: 10.7717/peerjcs.2211/fig-12

- k-nearest neighbors: Follows closely with an accuracy of 93.5% for both classes, showcasing its effectiveness in discerning patterns in the data.
- Support vector machines: Demonstrates a balanced accuracy of 92.3%, making it a reliable choice for scenarios where equal importance is assigned to both classes.
- Linear discriminant analysis: Achieves an accuracy of 90.1%, highlighting its proficiency in distinguishing between malicious and non-malicious instances.
- Quadratic discriminant analysis: While exhibiting an accuracy of 88.5%, it provides a pragmatic solution, especially when dealing with non-linear relationships in the data.

This Fig. 12 serves as a comprehensive guide for selecting an appropriate classification algorithm based on the specific requirements of a security system. The performance insights gleaned from this analysis contribute to informed decision-making in threat detection and security applications.

ANN offloading decisions-making performance analysis:

In smart home systems, the decision to offload data is a critical aspect that significantly influences resource utilization and overall system performance. Employing ANN (*Khan et al., 2024*) is a pivotal strategy for intelligent offloading decisions and weighing factors such as computational complexity, latency requirements, and resource availability. The process involves carefully selecting features, training the ANN with historical data to learn patterns, and deploying it for real-time decision-making based on current environmental conditions and system states.

In Fig. 13, the training and validation data are plotted for Epoch values ranging from 0 to 10. A gray line represents the train data, while the validation data is displayed with an

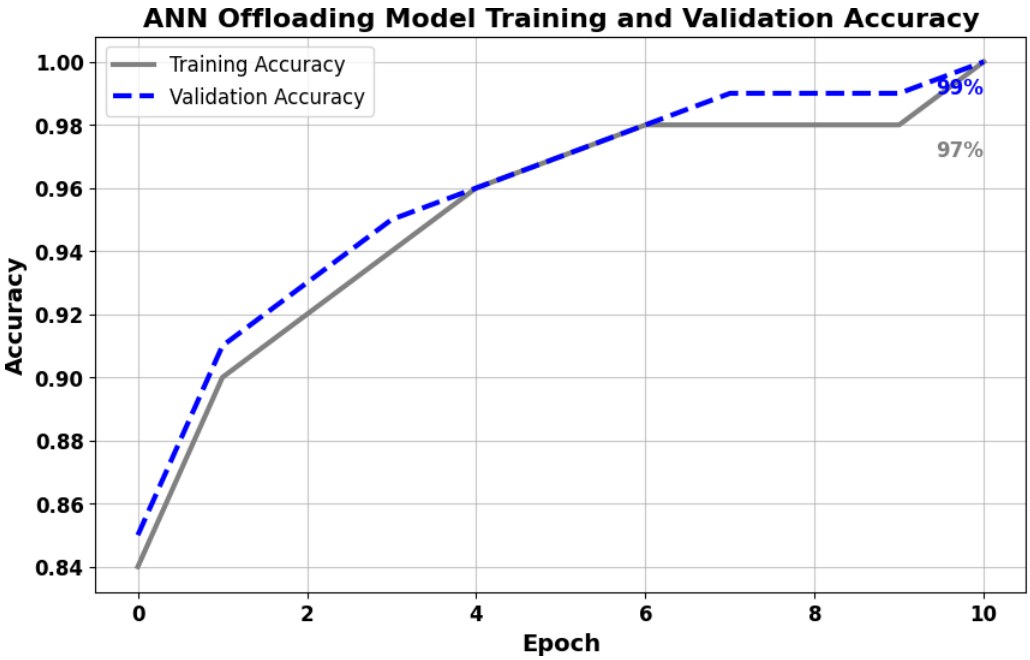

Figure 13 **ANN offloading model training and validation accuracy.**
Full-size DOI: 10.7717/peerjcs.2211/fig-13

blue line. The training accuracy plot shows how well the model performs on the training data over epochs. As the number of epochs increases, the model learns from the training data, increasing accuracy.

The validation accuracy plot indicates how well the ANN model generalizes to unseen data (validation set). A close alignment between training and validation accuracy suggests good generalization. In the plot, both training and validation accuracies start from relatively low values and gradually increase over epochs. The final values of 97% for training accuracy and 99% for validation accuracy indicate that the ANN model performs well on both the training and validation sets.

Figure 14 depicts the ANN model training and validation loss. The training loss plot illustrates how the loss (error) decreases over epochs on the training data. A decreasing loss indicates that the model is learning to make better predictions. The validation loss plot shows the loss on the validation set. In the plot, training and validation losses decrease steadily over epochs, which indicates that the model is learning effectively. The convergence of training and validation losses to low values (0.12 and 0.11) suggests that the model is well-trained and generalizes well to unseen data.

A confusion matrix is a helpful tool for summarizing the performance of a classification algorithm based on the ANN model. We can better understand the categorization model's achievements by calculating a confusion matrix. Figure 15 presents the confusion matrix as a predicted label chart. The dark blue box values represent correct predictions, while

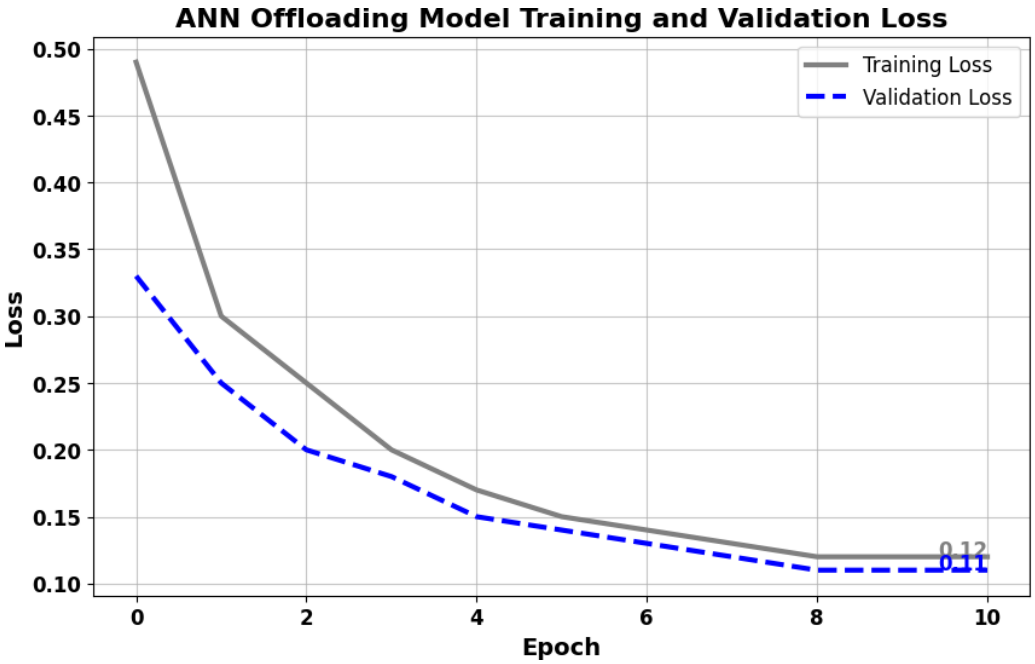

Figure 14 **ANN offloading model training and validation loss.**
Full-size DOI: 10.7717/peerjcs.2211/fig-14

the white box values represent incorrect predictions. From a 70/30 data partition, we have achieved an accuracy of 92%.

In smart home systems, the decision to offload data is a critical aspect that significantly influences resource utilization and overall system performance. Employing ANN is a pivotal strategy for intelligent offloading decisions and weighing factors such as computational complexity, latency requirements, and resource availability. The process involves carefully selecting features, training the ANN with historical data to learn patterns, and deploying it for real-time decision-making based on current environmental conditions and system states.

Figure 16 presents innovative offloading strategies based on ANN that can be categorized into Edge Offloading (Decision-1), Fog Offloading (Decision-2), and Cloud Offloading (Decision-3). Edge Offloading is suitable for tasks with low latency requirements and moderate complexity, allowing quick processing on nearby edge devices. Fog Offloading extends this by accommodating tasks that require more computational power but can still benefit from low-latency processing at fog nodes closer to the edge. Finally, Cloud Offloading is reserved for complex tasks that are not time-sensitive, utilizing the extensive processing capabilities of the cloud.

The core function of our ANN model is to assess different scenarios by considering multiple factors such as QoS requirements, energy constraints, and available bandwidth. By integrating these parameters into the decision-making process, our model strives to identify the most suitable offloading strategy for each computational task within the smart home environment. The output generated by our ANN model provides comprehensive

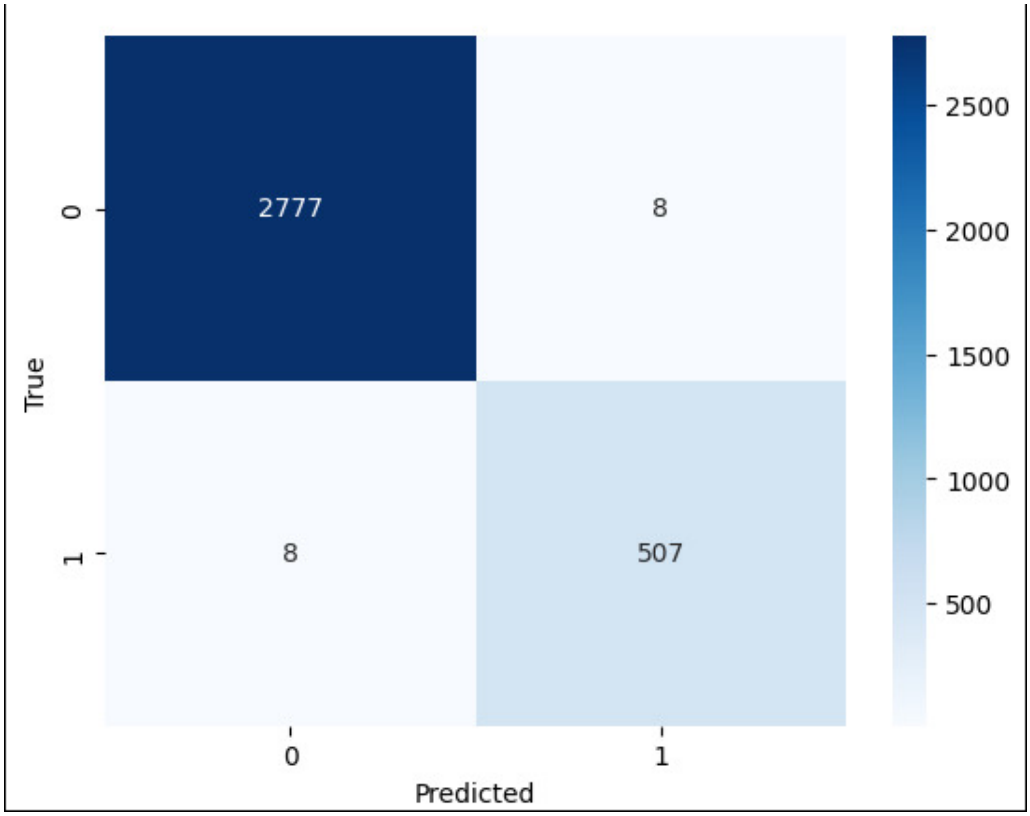

Figure 15 **ANN offloading confusion matrix.**

Full-size DOI: 10.7717/peerjcs.2211/fig-15

and conclusive results based on the evaluated metrics and decision scenarios. These results facilitate optimal resource utilization and performance enhancement within the smart home ecosystem. By dynamically determining the most appropriate offloading destination for each task, our model maximizes efficiency, minimizes energy consumption, and ensures seamless operation of smart home devices and services.

Our performance analysis delves into the intricate dynamics of smart home environments, leveraging advanced ANN techniques to enable informed offloading decisions. Through meticulous evaluation and optimization, our approach aims to enhance overall resource utilization and elevate the performance of smart home systems, ultimately delivering a superior user experience.

Challenges and considerations for real-world implementation

While the integration of IoT-based home automation systems offers significant benefits, several challenges must be addressed to ensure successful implementation in real-world environments. These challenges include resource management, security and privacy concerns, network reliability, and user acceptance and adaptability.

- Resource management:

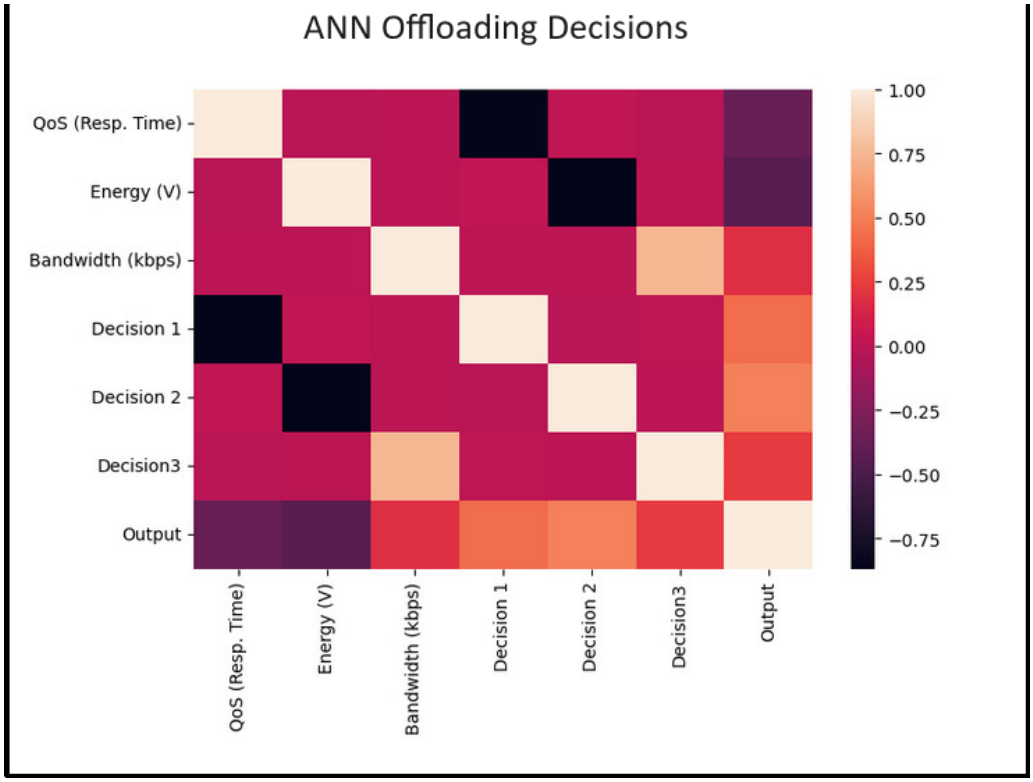

Figure 16 **ANN offloading strategies results presentation.**
Full-size DOI: 10.7717/peerjcs.2211/fig-16

One of the primary challenges is the high demand for resources. Home automation applications require substantial processing power, memory, and energy. This can lead to rapid battery depletion and reduced performance of other applications on the device. Effective resource management strategies are needed to optimize the use of these resources. Techniques such as efficient coding, adaptive resource allocation, and offloading computational tasks to cloud or edge servers can mitigate these issues.

- Security and privacy:

IoT devices are inherently vulnerable to security breaches due to their limited computational resources and the open nature of their communication protocols. Ensuring data security and user privacy is paramount. This involves implementing robust encryption methods, secure authentication protocols, and regular software updates to protect against emerging threats. Privacy-by-design principles should be integrated into the development process, ensuring that data is anonymized and users have control over their personal information.

- Network reliability:

The seamless operation of home automation systems depends on reliable network connectivity. Network outages, latency, and bandwidth limitations can disrupt the communication between devices, leading to potential system failures. To address this, the implementation of redundant communication pathways and the use of local storage for critical data can enhance system resilience. Employing edge and fog computing can reduce dependency on centralized cloud services, thus improving responsiveness and reliability.

- Interoperability:

Real-world environments often consist of diverse IoT devices from various manufacturers, each with communication protocols and standards. Ensuring interoperability among these devices is a significant challenge. Developing and adhering to universal standards and protocols can facilitate seamless integration and communication among different devices, enhancing the home automation system's overall functionality and user experience.

- Cost and scalability:

The initial cost of setting up a comprehensive IoT-based home automation system can be high, potentially limiting its accessibility to a broader audience. Additionally, as the number of connected devices increases, the system must be scalable to handle the growing data and processing demands. Cost-effective solutions and scalable architectures are essential for widespread adoption and long-term viability.

- Environmental impact:

Finally, the environmental impact of deploying numerous IoT devices must be considered. These devices contribute to electronic waste and consume energy. Developing energy-efficient devices and promoting sustainable practices in manufacturing and disposal can mitigate the environmental footprint of home automation systems.

CONCLUSION & FUTURE WORK

This study presents an innovative, Trusted IoT Big Data Analytics (TIBDA) framework for securing IoT data in smart homes, utilizing cutting-edge cryptographic methods to protect the privacy and confidentiality of user information. By integrating ECC, PQC, and BCT, our framework embeds trust mechanisms, guaranteeing secure processing and user information confidentiality. Additionally, we comprehensively compared prominent anomaly detection and machine learning classification algorithms to detect anomalies and malicious activities effectively. Moreover, our framework introduces the ANN dynamic offloading algorithm. It optimizes secure task distribution among Edge, Fog, and Cloud resources based on real-time conditions, enhancing operational efficiency and overall system performance. The comprehensive analysis reveals that TIBDA consistently outperforms competing systems such as SAS-Cloud, SHCEF, AILBSM, MHE-IS-CPM, and PA. Specifically, TIBDA's average response times are 10–20% lower, demonstrating its superior efficiency in handling varying loads. TIBDA maintains a 5–15% higher AUC value

in security performance, indicating robust threat mitigation capabilities. Additionally, TIBDA's trustworthiness, measured through uptime percentages, surpasses other systems by 10–12%, ensuring higher reliability and system availability. In anomaly detection, TIBDA's Isolation Forest algorithm excels with 99.30% and random forest algorithm 94.70% accuracy, significantly higher than other methods. Our ANN-based offloading decision-making model for smart home systems achieved a validation accuracy of 99% and minimized loss to 0.11, indicating effective optimization of resource utilization and system performance. These findings confirm TIBDA as a highly effective and reliable system for diverse real-world applications, offering significant improvements in performance, security, anomaly detection, reliability, and smart home system efficiency over other existing methods.

Despite the significant findings of our study, there are certain limitations and opportunities for future research. The dataset diversity in our experiments, although broad, may still be limited, and incorporating additional datasets representing a wider array of scenarios could enhance the generalizability and robustness of our findings. Moreover, while we employed various evaluation metrics, exploring alternative metrics and benchmarking criteria could provide a more comprehensive understanding of our model's performance. Scalability and performance are also crucial considerations; future studies should evaluate the model's scalability under different workload conditions and in real-world environments with more IoT devices and users. Further research could also focus on enhancing security mechanisms, such as integrating advanced cryptographic techniques and anomaly detection algorithms to protect against emerging threats. Additionally, developing privacy-preserving techniques, such as differential privacy and federated learning, will be important to ensure data confidentiality and compliance with regulations. As IoT ecosystems evolve, ensuring interoperability and standardization across heterogeneous devices will be essential, and future research should explore interoperability standards and protocols. Finally, real-world deployment and validation of our proposed security model in collaboration with industry partners will be critical to assessing its practical viability, effectiveness, usability, and scalability in diverse real-world scenarios.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This research received funding from King Saud University through Researchers Supporting Project Number (RSPD2024R1060), King Saud University, Riyadh, Saudi Arabia. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
King Saud University through Researchers Supporting Project Number, King Saud University, Riyadh, Saudi Arabia: RSPD2024R1060.

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Sheharyar Khan conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Zheng Jiangbin conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, authored or reviewed drafts of the article, and approved the final draft.
- Farhan Ullah conceived and designed the experiments, performed the experiments, analyzed the data, authored or reviewed drafts of the article, and approved the final draft.
- Muhammad Pervez Akhter conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Sohrab Khan conceived and designed the experiments, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Fuad A. Awwad conceived and designed the experiments, analyzed the data, prepared figures and/or tables, and approved the final draft.
- Emad A.A. Ismail performed the experiments, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The raw measurements are available in the Supplemental Files.

The data is available at Zenodo:

- Sheharyar, K. (2024). smart home dataset [Data set]. Zenodo. https://doi.org/10.5281/zenodo.10520983

- Subutai Ahmad, Alexander Lavin, Scott Purdy, Zuha Agha, Ian Danforth, Marcus Lewis, Luiz Scheinkman, Matthew Taylor, Jay Gokhale, Tom Silver, Mikhail Smirnov, Austin Marshall, Marion Le Borgne, Yuya Takashina, Sergey Kibish, ish-vlad, Celeste Baranski, Richard Crowder, m.fab, …breznak. (2019). numenta/NAB: v1.1 (v1.1). Zenodo. https://doi.org/10.5281/zenodo.3571294

The code is available at Colab and Zenodo:

- https://colab.research.google.com/drive/1VBlRtBuVXmfFKT2sE20KTkZNuN_Nwjw0?usp=sharing

- Sheharyar Khan. (2024). shksherry/Smat-home-data-AI-ML-algorithms-Analysis: Smat-home-data-AI-ML-algorithms-Analysis (Smat-home-data-AI-ML-algorithms-Analysis). Zenodo. https://doi.org/10.5281/zenodo.12671357.

- https://colab.research.google.com/drive/1jnm95PnRKGMB3IK0Qr7b-Q1ZF9W0yvGn?usp=sharing

- Sheharyar Khan. (2024). shksherry/Dynamic-Offloading-of-ANN-Model-Algorithm-Analysis: Dynamic-Offloading-of-ANN-Model-Algorithm-Analysis (Offloading). Zenodo. https://doi.org/10.5281/zenodo.12671340

- https://colab.research.google.com/drive/1wUezSCARGsUhO2etx1KOWZTY3r2N1rUB?usp=sharing

- Sheharyar Khan. (2024). shksherry/TIBDA-Numenta-Anomaly-Benchmark-NAB-dataset-Analysis-: TIBDA-Numenta-Anomaly-Benchmark-NAB-dataset-Analysis (TIBDA-Numenta-Anomaly-Benchmark-NAB-dataset-Analysis). Zenodo. https://doi.org/10.5281/zenodo.12671363.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.2211#supplemental-information.

REFERENCES

Abbas K, Tawalbeh LA, Rafiq A, Muthanna A, Elgendy IA, Abd El-Latif AA. 2021. Convergence of blockchain and IoT for secure transportation systems in smart cities. *Security and Communication Networks* **2021**:1–13 DOI 10.1155/2021/5597679.

Abu-Tair M, Djahel S, Perry P, Scotney B, Zia U, Carracedo JM, Sajjad A. 2020. Towards secure and privacy-preserving IoT enabled smart home: architecture and experimental study. *Sensors* **20**(**21**):6131 DOI 10.3390/s20216131.

Achar S. 2022. Cloud computing security for multi-cloud service providers: controls and techniques in our modern threat landscape. *International Journal of Computer and Systems Engineering* **16**(**9**):379–384.

Ahanger TA, Tariq U, Ibrahim A, Ullah I, Bouteraa Y. 2020. Iot-inspired framework of intruder detection for smart home security systems. *Electronics* **9**(**9**):1361 DOI 10.3390/electronics9091361.

Ahmad S, Mehfuz S, Beg J. 2023. Hybrid cryptographic approach to enhance the mode of key management system in cloud environment. *The Journal of Supercomputing* **79**(**7**):7377–7413 DOI 10.1007/s11227-022-04964-9.

Alalade ED. 2020. Intrusion detection system in smart home network using artificial immune system and extreme learning machine hybrid approach. In: *2020 IEEE 6th world forum on internet of things (WF-IoT)*. Piscataway: IEEE, 1–2.

Ammi M, Alarabi S, Benkhelifa E. 2021. Customized blockchain-based architecture for secure smart home for lightweight IoT. *Information Processing & Management* **58**(**3**):102482 DOI 10.1016/j.ipm.2020.102482.

Anđelić N, Baressi Šegota S, Car Z. 2023. Improvement of malicious software detection accuracy through genetic programming symbolic classifier with application of dataset oversampling techniques. *Computers* **12**(**12**):242 DOI 10.3390/computers12120242.

Anthi E. 2022. Detecting and defending against cyber attacks in a smart home Internet of Things ecosystem. PhD thesis, Cardiff University.

Azzaoui AE, Sharma PK, Park JH. 2022. Blockchain-based delegated Quantum Cloud architecture for medical big data security. *Journal of Network and Computer Applications* **198**:103304 DOI 10.1016/j.jnca.2021.103304.

Bajaj K, Sharma B, Singh R. 2022. Implementation analysis of IoT-based offloading frameworks on cloud/edge computing for sensor generated big data. *Complex & Intelligent Systems* **8(5)**:3641–3658 DOI 10.1007/s40747-021-00434-6.

Benchmark NA. 2020. Numenta anomaly benchmark. *Available at https://Github.Com/Numenta/NAB* (accessed on March 15 2024).

Bilgin ME, Kilinc HH, Zaim AH. 2022. An anomaly detection study for the smart home environment. In: *2022 7th international conference on computer science and engineering (UBMK)*. Piscataway: IEEE, 31–36.

Bin Aftab MU. 2017. *Building bluetooth low energy systems*. Packt Publishing Ltd.

Buil-Gil D, Kemp S, Kuenzel S, Coventry L, Zakhary S, Tilley D, Nicholson J. 2023. The digital harms of smart home devices: a systematic literature review. *Computers in Human Behavior* **145**:107770 DOI 10.1016/j.chb.2023.107770.

Bulgurcu B, Cavusoglu H, Benbasat I. 2010. Information security policy compliance: an empirical study of rationality-based beliefs and information security awareness. *MIS Quarterly* 523–548.

Chifor B-C, Bica I, Patriciu V-V, Pop F. 2018. A security authorization scheme for smart home Internet of Things devices. *Future Generation Computer Systems* **86**:740–749 DOI 10.1016/j.future.2017.05.048.

Chithaluru P, Al-Turjman F, Dugyala R, Stephan T, Kumar M, Dhatterwal JS. 2024. An enhanced consortium blockchain diversity mining technique for IoT metadata aggregation. *Future Generation Computer Systems* **152**:239–253 DOI 10.1016/j.future.2023.10.020.

Díaz M, Martín C, Rubio B. 2016. State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. *Journal of Network and Computer Applications* **67**:99–117 DOI 10.1016/j.jnca.2016.01.010.

Dilraj M, Nimmy K, Sankaran S. 2019. Towards behavioral profiling based anomaly detection for smart homes. In: *TENCON 2019–2019 IEEE region 10 conference (TENCON)*. Piscataway: IEEE, 1258–1263.

Edu JS, Such JM, Suarez-Tangil G. 2020. Smart home personal assistants: a security and privacy review. *ACM Computing Surveys (CSUR)* **53(6)**:1–36 DOI 10.1145/341238.

El-Sayed H, Sankar S, Prasad M, Puthal D, Gupta A, Mohanty M, Lin C-T. 2017. Edge of things: the big picture on the integration of edge, IoT and the cloud in a distributed computing environment. *Ieee Access* **6**:1706–1717 DOI 10.1109/ACCESS.2017.2780087.

Froiz-Míguez I, Fernández-Caramés TM, Fraga-Lamas P, Castedo L. 2018. Design, implementation and practical evaluation of an IoT home automation system for fog computing applications based on MQTT and ZigBee-WiFi sensor nodes. *Sensors* **18(8)**:2660 DOI 10.3390/s18082660.

Geneiatakis D, Kounelis I, Neisse R, Nai-Fovino I, Steri G, Baldini G. 2017. Security and privacy issues for an IoT based smart home. In: *2017 40th international convention on*

information and communication technology, electronics and microelectronics (MIPRO). Piscataway: IEEE, 1292–1297.

Haney J, Acar Y, Furman S. 2021. "It's the Company, the Government, You and I": user perceptions of responsibility for smart home privacy and security. In: *30th USENIX security symposium (USENIX Security 21)*. 411–428.

Haney JM, Furman SM, Acar Y. 2020. Smart home security and privacy mitigations: consumer perceptions, practices, and challenges. In: *HCI for cybersecurity, privacy and trust: second international conference, HCI-CPT 2020, held as part of the 22nd HCI international conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings 22*. Cham: Springer, 393–411.

Hoa NT, Van Huy L, Son BD, Luong NC, Niyato D. 2023. Dynamic offloading for edge computing-assisted metaverse systems. *IEEE Communications Letters* **27(7)**:1749–1753 DOI 10.1109/LCOMM.2023.3274649.

Hossain MD, Sultana T, Hossain MA, Layek MA, Hossain MI, Sone PP, Lee G-W, Huh E-N. 2022. Dynamic task offloading for cloud-assisted vehicular edge computing networks: a non-cooperative game theoretic approach. *Sensors* **22(10)**:3678 DOI 10.3390/s22103678.

Hsu Y-L, Chou P-H, Chang H-C, Lin S-L, Yang S-C, Su H-Y, Chang C-C, Cheng Y-S, Kuo Y-C. 2017. Design and implementation of a smart home system using multisensor data fusion technology. *Sensors* **17(7)**:1631 DOI 10.3390/s17071631.

Huang Q, Chen H, Zhang Q. 2020. Joint design of sensing and communication systems for smart homes. *IEEE Network* **34(6)**:191–197 DOI 10.1109/MNET.011.2000107.

Irshad A, Chaudhry SA. 2021. Comment on 'ElGamal cryptosystem-based secure authentication system for cloud-based IoT applications'. *IET Networks* **10(5)**:244–245 DOI 10.1049/ntw2.12014.

Irshad RR, Hussain S, Hussain I, Nasir JA, Zeb A, Alalayah KM, Alattab AA, Yousif A, Alwayle IM. 2023. IoT-enabled secure and scalable cloud architecture for multi-user systems: a hybrid post-quantum cryptographic and blockchain based approach towards a trustworthy cloud computing. *IEEE Access* **11**:105479–105498 DOI 10.1109/ACCESS.2023.3318755.

Jain P, Jain S, Zaïane OR, Srivastava A. 2021. Anomaly detection in resource constrained environments with streaming data. *IEEE Transactions on Emerging Topics in Computational Intelligence* **6(3)**:649–659 DOI 10.1109/TETCI.2021.3070660.

Jalasri M, Lakshmanan L. 2023. Managing data security in fog computing in IoT devices using noise framework encryption with power probabilistic clustering algorithm. *Cluster Computing* **26(1)**:823–836 DOI 10.1007/s10586-022-03606-2.

Joseph JV, Kwak J, Iosifidis G. 2019. Dynamic computation offloading in mobile-edge-cloud computing systems. In: *2019 IEEE wireless communications and networking conference (WCNC)*. Piscataway: IEEE, 1–6.

Kandhoul N, Dhurandher SK, Woungang I. 2021. Random forest classifier-based safe and reliable routing for opportunistic IoT networks. *International Journal of Communication Systems* **34(1)**:e4646 DOI 10.1002/dac.4646.

Kang K, Xu L, Wang W, Wu G, Wei J, Shi W, Li J. 2020. A hierarchical automata based approach for anomaly detection in smart home devices. In: *2020 international conferences on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData) and IEEE congress on cybermatics (Cybermatics)*. Piscataway: IEEE, 1–8.

Kashyap M, Sharma V, Gupta N. 2018. Taking MQTT and NodeMcu to IOT: communication in Internet of Things. *Procedia Computer Science* **132**:1611–1618 DOI 10.1016/j.procs.2018.05.126.

Katuk N, Ku-Mahamud KR, Zakaria NH, Maarof MA. 2018. Implementation and recent progress in cloud-based smart home automation systems. In: *2018 IEEE symposium on computer applications & industrial electronics (ISCAIE)*. Piscataway: IEEE, 71–77.

Khan MA, Ahmad I, Nordin AN, Ahmed AE-S, Mewada H, Daradkeh YI, Rasheed S, Eldin ET, Shafiq M. 2022. Smart android based home automation system using internet of things (IoT). *Sustainability* **14(17)**:10717 DOI 10.3390/su141710717.

Khan S, Jiangbin Z, Irfan M, Ullah F, Khan S. 2024. An expert system for hybrid edge to cloud computational offloading in heterogeneous MEC-MCC environments. *Journal of Network and Computer Applications* **225**:103867 DOI 10.1016/j.jnca.2024.103867.

Khan S, Zheng J, Khan S, Masood Z, Akhter MP. 2023. Dynamic offloading technique for real-time edge-to-cloud computing in heterogeneous MEC-MCC and IoT devices. *Internet of Things* **24**:100996 DOI 10.1016/j.iot.2023.100996.

Krishna BH, Kiran S, Murali G, Reddy RPK. 2016. Security issues in service model of cloud computing environment. *Procedia Computer Science* **87**:246–251 DOI 10.1016/j.procs.2016.05.156.

Kuldeep G, Zhang Q. 2022. Multi-class privacy-preserving cloud computing based on compressive sensing for IoT. *Journal of Information Security and Applications* **66**:103139 DOI 10.1016/j.jisa.2022.103139.

Kumari S, Singh M, Singh R, Tewari H. 2022. To secure the communication in powerful internet of things using innovative post-quantum cryptographic method. *Arabian Journal for Science and Engineering* **47(2)**:2419–2434 DOI 10.1007/s13369-021-06166-6.

Lara-Nino CA, Morales-Sandoval M, Diaz-Perez A. 2021. Post-quantum cryptography on wireless sensor networks: challenges and opportunities. *Integration of WSNs into Internet of Things* 81–99 DOI 10.1201/9781003107521-5.

Lee W-K, Chen L, Chang C-C, Yao Z. 2021. Post-quantum blockchain for secure communication in IoT-based smart home services. *International Journal of Embedded Systems* **14(5)**:509–524 DOI 10.1504/IJES.2021.120260.

Lin C, He D, Kumar N, Huang X, Vijayakumar P, Choo K-KR. 2019. HomeChain: a blockchain-based secure mutual authentication system for smart homes. *IEEE Internet of Things Journal* **7(2)**:818–829 DOI 10.1109/JIOT.2019.2944400.

Liu FT, Ting KM, Zhou Z-H. 2008. Isolation forest. In: *2008 eighth IEEE international conference on data mining*. Piscataway: IEEE, 413–422.

Majumder S, Ray S, Sadhukhan D, Khan MK, Dasgupta M. 2021. ECC-CoAP: elliptic curve cryptography based constraint application protocol for internet of things. *Wireless Personal Communications* **116(3)**:1867–1896 DOI 10.1007/s11277-020-07769-2.

Mocrii D, Chen Y, Musilek P. 2018. IoT-based smart homes: a review of system architecture, software, communications, privacy and security. *Internet of Things* **1**:81–98 DOI 10.1016/j.iot.2018.08.009.

Mohammad ZN, Farha F, Abuassba AO, Yang S, Zhou F. 2021. Access control and authorization in smart homes: a survey. *Tsinghua Science and Technology* **26(6)**:906–917 DOI 10.26599/TST.2021.9010001.

Mu S, Zhong Z. 2020. Computation offloading to edge cloud and dynamically resource-sharing collaborators in Internet of Things. *EURASIP Journal on Wireless Communications and Networking* **2020**:1–21 DOI 10.1186/s13638-019-1618-7.

Nguyen DC, Pathirana PN, Ding M, Seneviratne A. 2021. Secure computation offloading in blockchain based IoT networks with deep reinforcement learning. *IEEE Transactions on Network Science and Engineering* **8(4)**:3192–3208 DOI 10.1109/TNSE.2021.3106956.

Nouioua T, Belbachir AH. 2023. The quantum computer for accelerating image processing and strengthening the security of information systems. *Chinese Journal of Physics* **81**:104–124 DOI 10.1016/j.cjph.2022.11.006.

Padhy RP, Patra MR, Satapathy SC. 2011. Cloud computing: security issues and research challenges. *International Journal of Computer Science and Information Technology & Security (IJCSITS)* **1(2)**:136–146.

Raghunath KK, Rengarajan N. 2019. Response time optimization with enhanced fault-tolerant wireless sensor network design for on-board rapid transit applications. *Cluster Computing* **22(Suppl 4)**:9737–9753 DOI 10.1007/s10586-017-1473-4.

Sanaa E, Bajit A, Barodi A, Chaoui H, Tamtaoui A. 2020. An optimized security vehicular Internet of Things-IoT-application layer protocols MQTT and COAP based on cryptographic elliptic-curve. In: *2020 IEEE 2nd international conference on electronics, control, optimization and computer science (ICECOCS)*. Piscataway: IEEE, 1–6.

Saxena U, Sodhi J, Singh Y. 2017. Analysis of security attacks in a smart home networks. In: *2017 7th international conference on cloud computing, data science & engineering-confluence*. Piscataway: IEEE, 431–436.

Selvarajan S, Srivastava G, Khadidos AO, Khadidos AO, Baza M, Alshehri A, Lin JC-W. 2023. An artificial intelligence lightweight blockchain security model for security and privacy in IIoT systems. *Journal of Cloud Computing* **12(1)**:38 DOI 10.1186/s13677-023-00412-y.

Shah K, Jadav NK, Tanwar S, Singh A, Pleşcan C, Alqahtani F, Tolba A. 2023. AI and blockchain-assisted secure data-exchange framework for smart home systems. *Mathematics* **11(19)**:4062 DOI 10.3390/math11194062.

Sharma A, Goyal T, Pilli ES, Mazumdar AP, Govil MC, Joshi RC. 2015. A secure hybrid cloud enabled architecture for internet of things. In: *2015 IEEE 2nd world forum on Internet of Things (WF-IoT)*. Piscataway: IEEE, 274–279.

Sharma P, Namasudra S, Crespo RG, Parra-Fuente J, Trivedi MC. 2023a. EHDHE: enhancing security of healthcare documents in IoT-enabled digital healthcare ecosystems using blockchain. *Information Sciences* **629**:703–718 DOI 10.1016/j.ins.2023.01.148.

Sharma S, Ramkumar K, Kaur A, Hasija T, Mittal S, Singh B. 2023b. Post-quantum cryptography: a solution to the challenges of classical encryption algorithms. In: *Modern electronics devices and communication systems: select proceedings of MEDCOM 2021*. 23–38.

Shouran Z, Ashari A, Priyambodo T. 2019. Internet of things (IoT) of smart home: privacy and security. *International Journal of Computer Applications* **182(39)**:3–8 DOI 10.5120/ijca2019918450.

Signoretti G, Silva M, Andrade P, Silva I, Sisinni E, Ferrari P. 2021. An evolving tinyml compression algorithm for iot environments based on data eccentricity. *Sensors* **21(12)**:4153 DOI 10.3390/s21124153.

Sikder AK, Babun L, Celik ZB, Acar A, Aksu H, McDaniel P, Kirda E, Uluagac AS. 2020. Kratos: multi-user multi-device-aware access control system for the smart home. In: *Proceedings of the 13th ACM conference on security and privacy in wireless and mobile networks*. 1–12.

Tang S, Gu Z, Yang Q, Fu S. 2019. Smart home iot anomaly detection based on ensemble model learning from heterogeneous data. In: *2019 IEEE international conference on big data (Big Data)*. Piscataway: IEEE, 4185–4190.

Tanwar S, Ramani T, Tyagi S. 2018. Dimensionality reduction using PCA and SVD in big data: a comparative case study. In: *Future internet technologies and trends: first international conference, ICFITT 2017, Surat, India, August 31–September 2, 2017, Proceedings 1*. Cham: Springer, 116–125.

Tchagna Kouanou A, Tchito Tchapga C, Sone Ekonde M, Monthe V, Mezatio BA, Manga J, Simo GR, Muhozam Y. 2022. Securing data in an internet of things network using blockchain technology: smart home case. *SN Computer Science* **3(2)**:167 DOI 10.1007/s42979-022-01065-5.

Ul Hassan I, Ali RH, Ul Abideen Z, Khan TA, Kouatly R. 2022. Significance of machine learning for detection of malicious websites on an unbalanced dataset. *Digital* **2(4)**:501–519 DOI 10.3390/digital2040027.

Ullah S, Zheng J, Din N, Hussain MT, Ullah F, Yousaf M. 2023. Elliptic curve cryptography; applications, challenges, recent advances, and future trends: a comprehensive survey. *Computer Science Review* **47**:100530 DOI 10.1016/j.cosrev.2022.100530.

Unal D, Al-Ali A, Catak FO, Hammoudeh M. 2021. A secure and efficient Internet of Things cloud encryption scheme with forensics investigation compatibility based on identity-based encryption. *Future Generation Computer Systems* **125**:433–445 DOI 10.1016/j.future.2021.06.050.

Uppuluri S, Lakshmeeswari G. 2023. Secure user authentication and key agreement scheme for IoT device access control based smart home communications. *Wireless Networks* **29(3)**:1333–1354 DOI 10.1007/s11276-022-03197-1.

Wang M, Zhang G, Zhang C, Zhang J, Li C. 2013. An IoT-based appliance control system for smart homes. In: *2013 fourth international conference on intelligent control and information processing (ICICIP)*. Piscataway: IEEE, 744–747.

Xu X, Liu Q, Luo Y, Peng K, Zhang X, Meng S, Qi L. 2019. A computation offloading method over big data for IoT-enabled cloud-edge computing. *Future Generation Computer Systems* **95**:522–533 DOI 10.1016/j.future.2018.12.055.

Yalcinkaya F, Aydilek H, Erten MY, İnanç N. 2020. IoT based smart home testbed using MQTT communication protocol. *International Journal of Engineering Research and Development* **12(1)**:317–324.

Yang S, Lee G, Huang L. 2022. Deep learning-based dynamic computation task offloading for mobile edge computing networks. *Sensors* **22(11)**:4088 DOI 10.3390/s22114088.

Yang J, Sun L. 2022. A comprehensive survey of security issues of smart home system:"Spear" and "Shields," theory and practice. *IEEE Access* **10**:124167–124192 DOI 10.1109/ACCESS.2022.3224806.

Yar H, Imran AS, Khan ZA, Sajjad M, Kastrati Z. 2021. Towards smart home automation using IoT-enabled edge-computing paradigm. *Sensors* **21(14)**:4932 DOI 10.3390/s21144932.

Yusoff ZYM, Ishak MK, Rahim LA, Ali O. 2022. Elliptic curve cryptography based security on mqtt system for smart home application. In: *2022 19th International conference on electrical engineering/electronics, computer, telecommunications and information technology (ECTI-CON)*. Piscataway: IEEE, 1–4.

Zaidan AA, Zaidan BB, Qahtan M, Albahri OS, Albahri AS, Alaa M, Jumaah FM, Talal M, Tan KL, Shir W. 2018. A survey on communication components for IoT-based technologies in smart homes. *Telecommunication Systems* **69**:1–25 DOI 10.1007/s11235-018-0430-8.

Zhang F, Wang H, Zhou L, Xu D, Liu L. 2023. A blockchain-based security and trust mechanism for AI-enabled IIoT systems. *Future Generation Computer Systems* **146**:78–85 DOI 10.1016/j.future.2023.03.011.

Zhang J, Zhou Z, Li S, Gan L, Zhang X, Qi L, Xu X, Dou W. 2018. Hybrid computation offloading for smart home automation in mobile cloud computing. *Personal and Ubiquitous Computing* **22**:121–134 DOI 10.1007/s00779-017-1095-0.