

Article

AI and Blockchain-Assisted Secure Data-Exchange Framework for Smart Home Systems

Khush Shah ¹, Nilesh Kumar Jadav ¹, Sudeep Tanwar ^{1,*}, Anupam Singh ², Costel Pleșcan ³,
Fayez Alqahtani ⁴ and Amr Tolba ⁵

¹ Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad 382481, Gujarat, India; 19bec120@nirmauni.ac.in (K.S.); 21ftphde53@nirmauni.ac.in (N.K.J.)

² Department of Computer Science and Engineering, Graphic Era Hill University, Dehradun 248002, Uttarakhand, India; anupamsingh@gehu.ac.in

³ Department of Civil Engineering, Transilvania University of Brașov, 00036 Brașov, Romania; plescan.costel@unitbv.ro

⁴ Software Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh 11437, Saudi Arabia; fhalqahtani@ksu.edu.sa

⁵ Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia; atolba@ksu.edu.sa

* Correspondence: sudeep.tanwar@nirmauni.ac.in

Abstract: The rapid expansion of the Internet of Things (IoT) on a global scale has facilitated the convergence of revolutionary technologies such as artificial intelligence (AI), blockchain, and cloud computing. The integration of these technologies has paved the way for the development of intricate infrastructures, such as smart homes, smart cities, and smart industries, that are capable of delivering advanced solutions and enhancing human living standards. Nevertheless, IoT devices, while providing effective connectivity and convenience, often rely on traditional network interfaces that can be vulnerable to exploitation by adversaries. If not properly secured and updated, these legacy communication protocols and interfaces can expose potential vulnerabilities that attackers may exploit to gain unauthorized access, disrupt operations, or compromise sensitive data. To overcome the security challenges associated with smart home systems, we have devised a robust framework that leverages the capabilities of both AI and blockchain technology. The proposed framework employs a standard dataset for smart home systems, from which we first eliminated the anomalies using an isolation forest (IF) algorithm using random partitioning, path length, anomaly score calculation, and thresholding stages. Next, the dataset is utilized for training classification algorithms, such as K-nearest neighbors (KNN), support vector machine (SVM), linear discriminate analysis (LDA), and quadratic discriminant analysis (QDA) to classify the attack and non-attack data of the smart home system. Further, an interplanetary file system (IPFS) is utilized to store classified data (non-attack data) from classification algorithms to confront data-manipulation attacks. The IPFS acts as an onsite storage system, securely storing non-attack data, and its computed hash is forwarded to the blockchain's immutable ledger. We evaluated the proposed framework with different performance parameters. These include training accuracy (99.53%) by the KNN classification algorithm and 99.27% by IF for anomaly detection. Further, we used the validation curve, lift curve, execution cost of blockchain transactions, and scalability (86.23%) to showcase the effectiveness of the proposed framework.

Keywords: blockchain; smart contract; smart home systems; Internet of Things; artificial intelligence; anomaly detection

MSC: 68T01



Citation: Shah, K.; Jadav, N.K.; Tanwar, S.; Singh, A.; Pleșcan, C.; Alqahtani, F.; Tolba, A. AI and Blockchain-Assisted Secure Data-Exchange Framework for Smart Home Systems. *Mathematics* **2023**, *11*, 4062. <https://doi.org/10.3390/math11194062>

Academic Editor: Jan Lansky

Received: 26 August 2023

Revised: 19 September 2023

Accepted: 21 September 2023

Published: 25 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The involvement of modern technology, such as the Internet of Things (IoT), blockchain, and artificial intelligence (AI), transforms the legacy Internet into the next-generation Internet, where everything is interconnected. The unprecedented proliferation of IoT recently has made every technology come closer and connect to improve the quality of life. Applications such as smart cities, smart grids, smart vehicles, and smart industries have potentially impacted the nation's economy. Through IoT applications, the devices connected to the Internet can be monitored and controlled from remote locations, enabling users to access the systems from any location. It enables system automation with better connectivity and communication, which results in efficiency in the system operations and better productivity. However, IoT has a severe security flaw because it utilizes lightweight protocols (e.g., message queue telemetry transport (MQTT), wireless fidelity (Wi-Fi), and constrained application protocol (CoAP)) and error-prone communication channels to relay sensitive data [1–3]. For example, IoT sensors deployed in the smart home system can be maneuvered by attackers, resulting in significant damage to the home and the living beings residing in it. Therefore, there is a stringent requirement for some ideal solution that can confront the security and privacy issues of IoT applications [4–6].

Security enthusiasts and researchers across the globe have proposed several security solutions to overcome the aforementioned security issues of IoT applications (e.g., smart home systems). For example, the authors of [7] proposed lightweight cryptography, which includes features like robustness, long-range data transfer, and an acceptable level of security. The lightweight algorithms are applied on intelligent IoT devices and analyze their performance on an open standard system called a long-range wide area network (LoRaWAN), which defines the communication protocol for low-power wide area network (LPWAN) technology. Similarly, the authors of [8] proposed another lightweight algorithm called “elliptic curve cryptography” for securing the data communication between the nodes in an IoT infrastructure. The proposed technique has been analyzed with the conventional lightweight algorithms to determine which algorithm has the most efficient technique to secure data. In [9], the authors proposed a Java-based encryption system to provide a more efficient security framework for the data stored on the cloud storage. The proposed approach combines the Rivest–Shamir–Adleman (RSA) and the data encryption standard (DES) algorithm to develop a synergized combination of the mentioned algorithms, thus strengthening the security of the data before storing them on the cloud. However, the cryptography algorithms do not provide the scope of automation and data immutability; moreover, with recent computing power, it is easier to break off the crypto cipher, thus degrading the performance of the smart home system. Further, the work proposed by [10–12] used a conventional signature-based intrusion-detection system that minimizes the security issues of smart home systems. Nevertheless, the intrusion-detection system has to rely on a high volume of data, which is a significant challenge as the system has to handle data efficiently without introducing latency. Moreover, it has to strike a balance between the detection of correct intrusions and minimizing false alarms [13,14].

The sensors associated with the IoT application show data readings from the surrounding environment (e.g., the sensor of a water treatment plant collects data readings of chlorine levels), which is essential for each sensor to accomplish a shared task [15,16]. However, it has been observed from the literature that these readings are manipulated by adversaries or are mistakenly errored by the legitimate personnel of the IoT application. Such data are formally known as anomalies, and it is essential to detect and remove them to enhance the performance of the IoT application. Recently, the advent of AI algorithms has shown a remarkable improvement in detecting anomalies and enhancing the security issues of IoT applications. For example, Emmanuel et al. in [17] present an AI-based solution comprising extreme machine learning techniques for classification tasks. In addition, a regression-based solution is also examined for anomaly detection in smart home systems. The authors have primarily focused on intrusion and anomaly detection on the Mozilla Gateway installed in their sensor network infrastructure. With modifications in the afore-

mentioned hybrid model, the authors achieved significant accuracy in anomaly detection. Similarly, Sihai et al. in [18] used ensemble techniques for anomaly detection in the smart home infrastructure. To overcome the issue of AI model overfitting, the authors combined the synthetic minority over-sampling technique (SMOTE) with ensemble machine learning models for better efficiency when the model was working with an unbalanced dataset.

Nevertheless, the AI algorithms are not secured from data integrity issues, where the attacker can target the IoT data to jeopardize AI learning. To resolve this issue, blockchain is a prominent solution that offers secure data storage. Researchers have explored multiple techniques to integrate blockchain technology into the smart home infrastructure to preserve data privacy [19]. For example, to preserve the privacy of traditional smart home systems, the authors of [20] proposed a homomorphic consortium blockchain framework to strengthen the security of the sensitive data in the infrastructure; the framework comprises an algorithm where the verification nodes are required to verify working nodes and transactions occurring in the network. The authors also introduce a new block data structure based on homomorphic encryption. They evaluated their proposed work using data availability, security, and robustness, in which it outperforms other existing state-of-the-art works. Similarly, the authors of [21] introduced a private blockchain network using the received signal strength's indicator-based trilateration to secure data privacy in the smart home infrastructure. The authors have proposed a three-layer intrusion detection system (IDS) to detect cyber-attacks in IoT networks. To track the sources of attack, the Kalman filtering method has been incorporated into the trilateration. The proposed system was tested on a physical setup to evaluate it with the existing systems.

However, the aforementioned approaches are lacking in terms of showing the amalgamation between AI and blockchain to strengthen the security of smart home systems. Motivated by the above-mentioned papers on anomaly detection and blockchain implementation in smart home systems, we propose a robust solution and framework where both the important components required for the security of the smart home systems from external threats have been incorporated and synergized. The proposed framework, in the case of any abnormal activity, generates an alert and simultaneously examines the threshold levels of the nominal data to prevent the system from any kind of failure. Further, the data are stored in a blockchain network for immutability, which records data in a method safe from any further attack.

1.1. Research Contributions

We proposed an AI- and BC-enabled secure framework to tackle network-related attacks on smart home systems. Since the IoT sensors deployed in the smart home systems use weak network interfaces and protocols, the attackers leverage this situation and exploit the sensor data exchange. Consequently, the susceptibility of these systems to cyber threats and unauthorized access is significantly heightened, posing serious security risks and underscoring the need for robust protective measures and advanced security solutions. To approach this challenge, the proposed work utilizes the standard smart home system dataset to train AI classifiers (such as K-nearest neighbor (KNN), support vector machine (SVM), linear discriminant analysis (LDA), and quadratic discriminate analysis (QDA)) to classify attack and non-attack data. Nevertheless, prior to classification, we employ anomaly-detection algorithms, such as local outlier factor (LOF) and isolation forest (IF), to remove falsified data from the original smart home system dataset. The rationale behind this is that if the AI classifiers are trained on falsified data, it deteriorates the AI training, which jeopardizes the operational performance of the smart home systems.

Further, we adopted the interplanetary file system (IPFS)-based Ethereum blockchain to confront data integrity issues. Here, the non-attack data from AI classifiers are allowed for secure data storage. For that purpose, a smart contract is designed, where different user-defined functions are utilized to validate the non-attack data. Incorporating IPFS improves the response time and the scalability of the blockchain network. The proposed framework is evaluated using different performance metrics, such as accuracy, lift curve, validation curve, and the blockchain's transaction and execution cost. A training accuracy

of 99.27% is achieved while finding the anomalies and 99.53% while classifying the attack and non-attack data. Further, due to the incorporation of IPFS, we achieved a scalability of 86.23% compared to the conventional blockchain.

1.2. Organization

The article is divided into sections, where Section 2 showcases the literature review; Section 3 introduces the main aim of the proposed work; Section 4 presents the proposed framework comprised of cognitive, AI, blockchain, and application layers to achieve the aim specified in Section 3; and the results and discussion are presented in Section 5. Finally, in Section 6, we conclude the article by providing the main insights of the proposed work.

2. Related Works

Various researchers worldwide have published the application of IoT in several domains, including smart home systems. However, these studies do not explore the possibilities of integrating AI and blockchain in resolving the security issues of smart home systems. For example, Cultice et al. in [22] proposed an autoencoder-based system to detect anomalies in smart home systems. They primarily focus on applying neural network algorithms in smart home systems and implementing them to prevent hazards in the environment where they are installed. Further, Lee et al. [23] present a blockchain-enabled secure solution to overcome security threats, such as device vulnerabilities and data integrity for the home gateway network. Their solution offers decentralization, immutability, and transparency to overcome the challenges associated with centralized systems. The framework proposed implements the developed blockchain network on the Ethereum platform. They assess their smart contracts using security response time and accuracy, where the results revealed that their designed smart contracts are more effective than the existing works.

Further, Hamed et al. in [24] proposed a detailed, layered system architecture for the IoT infrastructure called “AI4SAFE-IoT”. The developed architecture comprises security protocols and machine learning in its different layers to confront various IoT-related attacks. Their proposed system successfully detects attributes and also identifies the stage of an attack life cycle based on the “Cyber Kill Chain” model. The authors evaluated the proposed architecture based on the “IoT service management” score, where they achieved considerable results. Then, Prarthi et al. in [25] developed an anomaly-detection algorithm called “PiForest”. They first surveyed the implementation of various anomaly-detection algorithms in several cases and calculated the accuracy of the implemented algorithms. The authors further proposed their own anomaly-detection algorithm and implemented it in a real-time scenario. The accuracy of the “PiForest” algorithm was also compared with the accuracy of other algorithms to determine the performance of the developed algorithm.

Similarly, Subhi et al. [26] proposed an AI- and blockchain-based architecture to secure various IoT applications in the smart city. Their solution can automate certain tasks, such as environment monitoring, data aggregation, and data analysis. The analyzed data are forwarded to the AI expert engine for offering predictive services. Their experimental results show that the employed AI models achieved 95% accuracy. Further, they utilized blockchain to store the actual data after the classification task. Further, in [27], blockchain was adopted for a secure natural gas transaction framework. In their framework, buyers and sellers interact with each other to purchase the gas contract and maximize their profit. However, the solution did not utilize intelligence and automation to classify the attack and non-attack data. Next, the authors of [28] use the amalgamation of AI and blockchain to promote sustainable IoT by enhancing the security and privacy issues of the smart city. Alternatively, the work proposed by [29] uses a learning engine for a smart home communication network that uses blockchain and cloud-based data evaluation to improve security. The proposed algorithm outperforms existing methods in terms of computation complexity, false authentication rate, and qualitative parameters.

It is also observed from the literature that most existing solutions do not amalgamate both AI and blockchain to strengthen the performance of smart home systems [30].

For instance, they did not consider anomaly detection with classification. Further, their blockchain-based solutions are computationally expensive because blockchain has to process both attack and non-attack data. Further, the work proposed by [31] uses differential privacy and the indispensable properties of blockchain to enhance security in smart home systems. Their results show outperformance regarding scalability, confidentiality, and resilience against data tampering. To analyze the feasibility of blockchain technology in the smart home system, Arif, Yiyang et al. in [32,33] examine the adaptability of blockchain by developing a consortium blockchain-based testbed. Only a few papers have shown an amalgamation of blockchain and AI specifically for smart home systems. For example, Ref. [34] proposed a private-blockchain-based smart home network architecture that integrates an AI model for intrusion detection. Similarly, the authors in [35] use AI and blockchain to propose a secure monitoring system for the COVID-19 outbreak. The aforementioned papers incorporate AI and blockchain for different applications and are not considered smart home systems. Moreover, most researchers have shown their significance in terms of survey or review papers [36]. From that viewpoint, we propose a secure and intelligent framework for secure data exchange in the smart home environment by incorporating AI and blockchain technology.

In this context, Table 1 displays the comparative analysis between the state-of-the-art works and the proposed work. Therefore, the proposed work offers a secure pipeline where the first anomalous data points are detected and eradicated from the smart home system dataset. Further, the employed AI models bifurcate attack and non-attack data, and only non-attack data are forwarded to the blockchain network for secure storage.

Table 1. Comparison between the existing works and the proposed work.

Author	Year	Objective	Pros	Cons
The proposed work	2023	Detects fraudulent behavior in the smart home system through the amalgamation of blockchain and AI for improved security with automation.	Utilizes state-of-the-art AI algorithms for anomaly detection and blockchain, assuring security.	Due to high mining costs, the blockchain network gets computationally expensive.
Nilupulee A. et al. [7]	2019	The research proposes a framework to secure long-range data communication for IoT systems.	Employed LoRaWAN communication for effective communication.	Amalgamation of AI and blockchain is missing.
M. Ayub et al. [8]	2020	The framework comprises a developed security algorithm to secure the internal communication between the IoT nodes.	Utilized lightweight cryptography—elliptic curve.	Does not explore the scope of integrating automation in the framework to detect malicious activities.
A. Kumar et al. [9]	2020	A Java-based cryptography system has been proposed to secure cloud computing systems from external attacks.	Integration of DES and RSA algorithms to secure the data before it gets deployed on cloud systems.	Cryptography can be deciphered using modern computing processors.
E.D. Alalade et al. [17]	2020	Proposes an ML framework to detect anomalies in smart systems through classification and regression-based methods.	Detects anomalies in a Mozilla Gateway installation and uses AI algorithms.	Anomaly detection is not performed.
S. Tang et al. [18]	2019	Proposes an ensemble-based ML approach to identify the anomalous behavior of the data present in the smart systems.	Integrates SMOTE with ensemble ML algorithms to detect anomalies in IoT networks.	Does not integrate AI and blockchain technology.
W. She et al. [20]	2019	Proposes a homomorphic consortium blockchain framework to strengthen the data security in smart systems.	Uses lightweight encryption—homomorphic to verify the data transaction in the blockchain network.	The scope of ML integration for breaches or data manipulation is not explored.

Table 1. *Cont.*

Author	Year	Objective	Pros	Cons
M. Jayson et al. [21]	2021	The framework proposes a multi-layer IDS to detect cyber-attacks in IoT networks.	Private-blockchain network based on IDS to secure smart home systems through Kalman filtering.	Did not employ the advantage of automation and intelligence.
T. Cultice et al.	2020	Proposes real-time anomaly detection system in smart home and smart grid infrastructures using autoencoders.	Implements autoencoders that detect the sensor drift quicker and more accurately.	Did not explore the data security after anomaly detection.
Y. Lee et al. [23]	2020	Proposes an Ethereum blockchain-based smart home gateway, a network that secures the gateway of smart homes.	The proposed Ethereum-based framework outperforms the standard models in the comparative analysis.	Did not explore the scope of machine learning for data classification.
H. Haddadjpajouh et al. [24]	2020	Proposes an ML-based framework to secure the smart systems called “AI4SAFE-IoT”, which is based on the “Cyber-Kill Chain” model.	High accuracy in the detection of security threats	The framework is not secured from data manipulation attacks.
P. Jain et al. [25]	2022	Developed an anomaly-detection algorithm called “Pi-Forest” to detect the anomalies in the data collected from the smart systems.	Detects anomalies with higher accuracies compared to the standard algorithms.	Did not integrate AI and blockchain technology.
S. M. Alrubei et al. [26]	2022	Provides a framework that integrates both ML and BT attributes to secure and automate COVID-19 prediction with considerable accuracy.	Implemented on low-power and low-cost RPi systems to detect and secure the COVID-19 data using AI algorithms.	Did not examine the need for IPFS in the blockchain network.
W. Xiao et al. [27]	2021	The framework proposes a blockchain network to secure gas transactions to maximize profits.	Efficient blockchain-based framework for securing gas transactions.	Computational cost is high.

3. System Model and Problem Formulation

This section elaborates the system model for the proposed framework, which consists of different homes represented by $\{h_1, h_2, h_3, \dots, h_n\} \in H$, and each h_i is equipped with various smart sensors, represented as $\{s_1, s_2, s_3, \dots, s_m\} \in S$. Each smart home (h_i) has at least one (s_i) or a group of multiple sensors $\{s_2, s_4, \dots, s_l\}$ deployed at various locations to offer smart home services.

$$s_i \text{ or } \{s_2, s_4, \dots, s_l\} \in h_i \tag{1}$$

Each (s_i) has a sensing capability, such as tracking the air quality, controlling the temperature of the freezer pipe, and sensing the motion. These sensing capabilities are the data readings of sensors denoted as D , such that $\{d_1, d_2, \dots, d_l\} \in D$. A source sensor s_i sends the aforementioned data reading (d_i) to the receiver sensor s_j to take essential action \mathcal{A} . For example, if the freezer temperature rises to a certain threshold, an immediate alert is generated to lower the temperature. Moreover, to offer such services, each sensor has to exchange data with other sensors, wherein lowering the temperature is the specific action taken (\mathcal{A}).

$$s_i \xrightarrow{\sum_{i=1}^l (d_i) \text{ sends}} s_j \xrightarrow{\text{take}} \mathcal{A} \tag{2}$$

$$\text{if, } d_i \leq \text{or } \geq \mathcal{T} \tag{3}$$

where \mathcal{T} is some specific threshold that the sensor’s data reading has to maintain; otherwise, a necessary action \mathcal{A} is triggered in the smart home system. Moreover, the (s_i) uses a

network interface (e.g., public internet, Wi-Fi, etc.) to relay d_i to the receiver sensor (s_j). The Wi-Fi network is open to various network-related attacks, such as session hijacking, data integrity attacks, malware, and DDoS, that can deteriorate the performance of smart home systems. An attacker (Ψ) can exploit the communication channel (C) and manipulate the data exchange (d_i) between the source sensor (s_i) and the receiver sensor (s_j). In addition, Ψ can also deploy a rogue sensor node (s_k) in the smart home system that acts as a man in the middle to maneuver the data exchange (d_i) of smart home systems.

$$\Psi \xrightarrow{\text{exploit}} C \xrightarrow[\text{manipulates}]{d_i} d'_i \tag{4}$$

$$\Psi(s_k) \xrightarrow{\text{manipulates}} d_i \rightarrow d'_i \tag{5}$$

where d'_i is the manipulated data exchange between s_i and s_j . Therefore, there is a need for an automated and intelligent mechanism that can detect such malicious activity and resolve the security and privacy issues of the smart home system.

The main aim of the proposed work is to secure the data exchange between the source and the receiver sensor. For that purpose, an objective function O_f is formulated, which is defined as

$$O_f = \max_{\text{secure}} \sum_{i=0}^l (d_i) \tag{6}$$

where d_i is the data reading relayed between s_i and s_j .

4. Proposed Framework

This section presents the proposed framework for the IoT-based smart home system. The proposed framework has multiple layers, i.e., cognitive, AI, blockchain, and application layers, that provide a sequential flow, i.e., data acquisition from sensors, classifying the data (malicious or non-malicious), and securing them in the blockchain. Figure 1 depicts the proposed framework with its associated entities. A summarized explanation of Figure 1 is as follows.

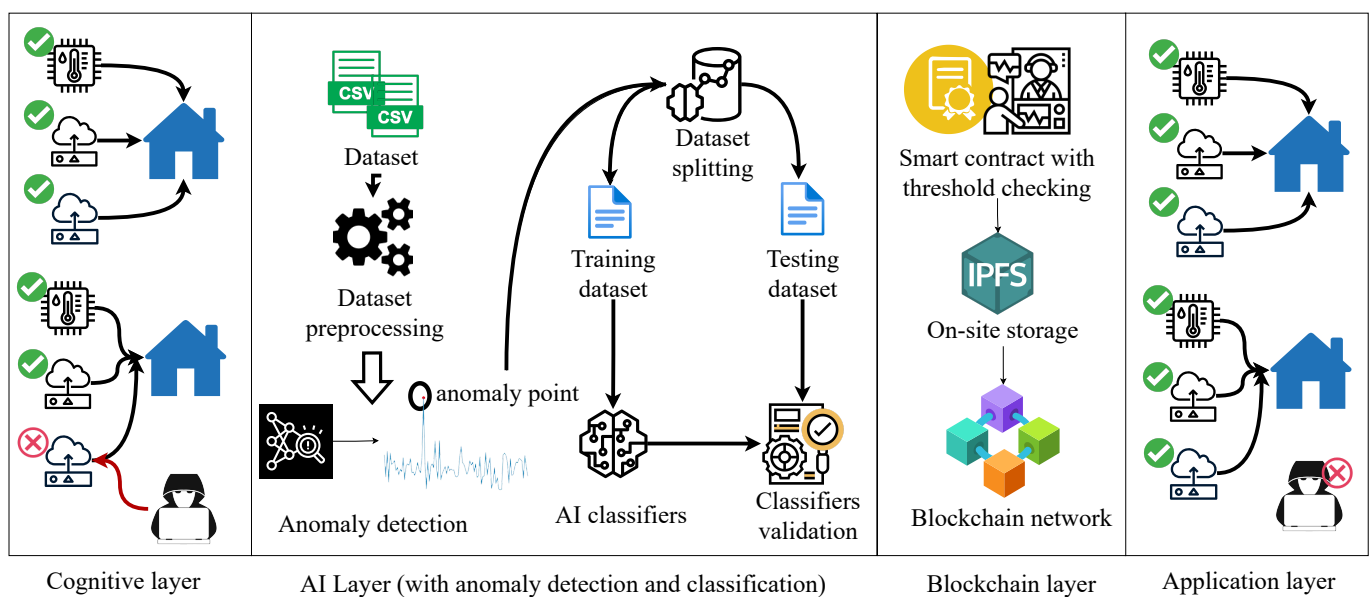


Figure 1. The proposed framework.

4.1. Cognitive Layer

The cognitive layer consists of an IoT-based smart home system that comprises several smart sensors, such as thermostats, motion sensors, light, water leak, and smoke sensors. These sensors are capable of capturing the surrounding data (d) to trigger a specific action associated with the event. For instance, if a water leak sensor detects any water leak in a sewage pipe, it triggers an alarm system.

$$S = \{s_1, s_2, \dots, s_m\} \tag{7}$$

where $s_1, s_2,$ and s_m are the temperature, humidity, carbon monoxide, and butane sensors, respectively belonging to S installed in the smart home system. The sensor (s_i) has a multitude of data points that pertain to its operational aspects, including sensor readings, updates, and maintenance records, which are expressed as $\{d_1, d_2, \dots, d_l\}$, where each d_i represents a specific piece of information in smart home systems. Further, s_i transmits d_i to another sensor s_j to accomplish a collaborative task (e.g., turn on the light, detect an open window, and many more).

$$\{d_1, d_2, \dots, d_l\} \in s_i \tag{8}$$

$$\text{Each } s_i \xrightarrow[d_i]{\text{sends}} s_j \tag{9}$$

The data collected by these sensors are vital and need to be secure from adversaries that try to manipulate it to degrade the performance of the smart home system. Moreover, an adversary k can use a malicious sensor s_k that impersonates a legitimate sensor and jeopardizes the efficiency of different sensors deployed on the smart home system.

$$\Psi(s_k) \xrightarrow[d'_i]{\text{sends}} s_j \tag{10}$$

where $\Psi(s_k)$ is the adversary that uses a malicious sensor s_k that sends the malicious data (payload d'_i) to s_j . Moreover, the smart home system utilizes the public network, which is open to several attacks, such as sniffing the network traffic, session hijacking, and data integrity attacks [37,38]. The attacker can easily lure such open networks to thwart the data dissemination of the sensors attached to the smart home system.

$$s_i \xrightarrow[\text{Public network}]{d'_i} s_j \tag{11}$$

Moreover, conventional solutions are not automated or intelligent enough to detect such malicious activities in the smart home system. Therefore, there is a need for a proactive mechanism that efficiently detects malicious activities in the smart home system.

4.2. AI Layer

In this section, we present the working mechanism of the AI layer by adopting different AI algorithms, such as KNN, SVM, LDA, and QDA. This subsection is divided into two parts, i.e., Dataset Description and Adoption of AI algorithms. A detailed explanation of each subsection is as follows.

4.2.1. Dataset Description

The cognitive layer collects malicious and non-malicious data from the smart home system. For this purpose, we used a standard smart home system dataset, i.e., the TON IoT dataset [39], which comprises different IoT sensors, such as garage doors, refrigeration, weather, and motion sensors. The entire dataset is bifurcated into different service profiles, i.e., IoT fridge activity, IoT garage door activity, location tracker activity, thermostat activity, and many others. The dataset of the services describes the features of the activity, such as “fridge temperature” in the fridge activity, “latitude”, and “longitude” in the location

tracker activity, and many other relevant features in the other service profiles. From [39], we acquired multiple datasets of smart home systems. For instance, a dataset of garage door (D_1), fridge activity (D_2), GPS tracker (D_3), motion activity (D_4), and weather (D_5). Therefore, a smart home system dataset is represented as $D \in \{D_1, D_2, \dots, D_5\}$. Each dataset ($D_i \in D$) comprises the number of rows (w) and columns (q), as represented in Equation (12).

$$D_i^{w \times q} = D_i^{6401 \times 5} \tag{12}$$

4.2.2. Dataset Preprocessing

In this phase, the dataset ($D_i \in D$) is preprocessed using data preprocessing steps [40,41]. In D_i , there are inconsistencies, such as missing values, not a number (NaN), infinity values, not a normalized column, and datatype casting. Consider the dimension of D_i , expressed as

$$D_i^{w \times q} = \begin{pmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,q} \\ d_{2,1} & d_{2,2} & \dots & d_{2,q} \\ \vdots & \vdots & \ddots & \vdots \\ d_{w,1} & d_{w,2} & \dots & d_{w,q} \end{pmatrix} \xrightarrow{\text{contains}} \{ " - ", \inf(\infty), NaN \} \tag{13}$$

where $\{ " - ", \inf(\infty), NaN \}$ are the missing values, ∞ is infinity values and NaN is the value that is filled using the central tendency value, i.e., mean (v).

$$\begin{pmatrix} d_{1,1} & \text{--} & \dots & d_{1,q} \\ d_{2,1} & d_{2,2} & \dots & NaN \\ \vdots & \vdots & \ddots & \vdots \\ \text{--inf} & d_{w,2} & \dots & d_{w,q} \end{pmatrix} \xrightarrow[\text{with } v]{\text{filled}} \begin{pmatrix} d_{1,1} & v & \dots & d_{1,q} \\ d_{2,1} & d_{2,2} & \dots & v \\ \vdots & \vdots & \ddots & \vdots \\ v & d_{w,2} & \dots & d_{w,q} \end{pmatrix} \tag{14}$$

Further, we analyzed the normalization of the dataset D_i , where the values of the i th column of D_i are not scaled up properly, for example, the value of $d_{1,1} \gg d_{2,1}$ or $d_{1,1} \ll d_{2,1}$. Therefore, normalization has to be performed on all columns of the dataset D_i . From that viewpoint, we utilized the min–max scalar, which is expressed as

$$\vartheta = \frac{d_i - d_i^{min}}{d_i^{max} - d_i^{min}} \tag{15}$$

where ϑ is the rescaled output for D_i , which is in the range $[0,1]$. d_i is the input value, and d_i^{min} and d_i^{max} are the minimum and maximum values of the i th column of D_i . Further, the D_i has columns that are incompatible with AI models due to their datatype. For example, a conditional probability-based AI algorithm cannot adopt the column with an object datatype. Hence, a suitable datatype conversion has to be performed on D_i .

$$\underbrace{\text{int } d_i}_{\text{same datatype}} = \underbrace{(\text{int}) d_i}_{\text{same datatype}} \tag{16}$$

Here, in Equation (16), an explicit datatype casting has been performed so that the AI algorithms can train on the dataset D_i . The final preprocessed dataset is represented as D'_i .

4.2.3. Anomaly Detection and Classification Task

Once the dataset is preprocessed, it is forwarded to the AI layer, where different AI models are employed for anomaly detection and classification purposes [42,43]. Here, the preprocessed dataset D'_i is split into the training and the testing datasets to validate the parameters of the trained model.

$$\forall D'_i = \begin{cases} D'_{train} \\ D'_{test} \end{cases} \tag{17}$$

The terms (D'_{train} and D'_{test}) represent the training and testing parts of the preprocessed dataset (D'_i). The dataset is split into a fraction of 0.8 (80%) and 0.2 (20%) for the training and testing, respectively, using the `train_test_split()` method. The validation of the model includes the multiple parameters through which its performance is analyzed. The model accuracy has been verified by re-iterating the model on the test data. Before classification on the (D'_i) dataset is performed, it is verified for anomaly detection, i.e., whether the attacker has manipulated the dataset or not. If an attacker has forged the dataset values, the AI models are trained on manipulated data and provide false results. As a result, it jeopardizes the performance of the entire smart home system.

In the AI layer, first, the anomaly-detection algorithms are iterated on the dataset (D'_i) to detect the behavior of the data, i.e., whether the data are anomalous or not. The algorithm detects the outliers or anomalies in the data and classifies them in the categories of anomaly and nominal data. Through model performance analysis, we found that IF is the best algorithm amongst other anomaly-detection algorithms that can efficiently detect outliers as an anomaly.

The algorithmic flow of IF is similar to the algorithmic flow of the random forest algorithm. The point of the tuple that is processed at the given point of time of model iteration will be segregated to find its behavior (anomaly or nominal). The number of divisions required to determine the location of that particular point or tuple is called an estimator. IF operates by constructing an ensemble of isolation trees. Each isolation tree is built by randomly selecting a feature and a random split value within the range of that feature. The feature and split value are used to partition the data into two subsets, which is known as random partitioning. This process is repeated recursively until each data point is isolated in its own leaf node. Once the tree is formed, as discussed above, the anomaly score of the feature value is calculated to determine the nature of that instance. The anomaly scores \mathcal{Z} can be formulated as

$$\mathcal{Z}(o) = 2^{\frac{-E(h(o))}{c(s)}} \tag{18}$$

where o represents the data point for which the anomaly score is being calculated. The term ($E(h(o))$) is the average path length of the data point o across all trees in the ensemble. Further, $c(s)$ is the normalization factor, i.e., the average path length along the isolation trees, where s represents the total number of data points in the dataset. The term $c(s)$ is defined through the formula.

$$c(s) = \begin{cases} 2h(s-1) - 2^{\frac{s-1}{n}}, & \text{for } s > 2 \\ 1, & \text{for } s = 2 \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

The structure of an isolation tree is the same as that of a binary tree. Thus, the $c(s)$ has been defined similarly to that of a binary tree, where each parent node has exactly two child nodes. The value obtained of the anomaly score \mathcal{Z} determines the behavior of the point. If the score is found near 1, it is classified as anomalous. If it is near 0.5, it is classified as a nominal point. The updated dataset D_a is the anomaly-free dataset, with only nominal data. However, it is to be noted that the D_a still has attack and non-attack data, where the attackers

have performed various network-related attacks to maneuver the performance of the smart home system. Therefore, classification algorithms are needed to classify the data (attack or non-attack) in D_a . Supervised learning algorithms are implemented and tested using various performance metrics to classify the data. From the result analysis, we can know that the KNN algorithm performs well compared to other existing AI models. The performance metrics of the iterated models are briefly discussed in Section 5. The algorithm classifies the data point through the distance metric and the number of neighbors defined in the algorithm. There are multiple available distance metrics, including Euclidean, Manhattan, and Minkowski. The Euclidean distance metric is implemented here for the classification in the iterated KNN model. The Euclidean distance between two data points can be formulated as

$$d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (20)$$

where d is the calculated distance between two data points p and q in the dataset. The Euclidean distance of the data point selected and the number of its neighbors is determined. The nearest neighbor n is selected, and on the basis of the highest behavior found of the selected number of neighbors, the behavior of the selected data point or tuple is determined. The algorithmic flow of the KNN model is shown in Algorithm 1.

Algorithm 1 Working mechanism of the KNN algorithm

Input: D_a

Output: Classification of anomalous and nominal data

```

1: procedure CLASSIFICATION(C)
2:   Dataset  $D_a \leftarrow$  KNN
3:   Select number of neighbors  $n$ .
4:   Select the distance metric.
5:   Calculate the distance through distance metric.
6:   Find the nearest neighbors.
7:   if  $\mathcal{P}_a > \mathcal{P}_n$  then
8:     Classify as an attack.
9:   else
10:    Classify as non-attack.
11:   end if
12: end procedure

```

The terms \mathcal{P}_a and \mathcal{P}_n play a crucial role in our analysis, as they define the probability of encountering attack and non-attack data, respectively, in the vicinity of the selected point. These probabilities, denoted as \mathcal{P}_a and \mathcal{P}_n , are determined through the KNN model. \mathcal{P}_a represents the likelihood of encountering attack-related data points near the selected location, while \mathcal{P}_n signifies the probability of finding non-attack data points in the same vicinity. In essence, these probabilities are derived from the KNN model, which, based on its training data and distance metrics, estimates the chances of a given point being associated with either an attack or non-attack scenario. By utilizing the KNN model's predictive capabilities, we can assess the risk associated with a specific location or data point, helping us make informed decisions in the context of security or anomaly detection. In post-classification, the behavior of the model is inspected, where if it is found to be an attack, the proposed system generates an alert. Otherwise, if the behavior is found to be non-attack, the data is stored in the blockchain network described in the blockchain layer.

4.3. Blockchain Layer

In this layer, the non-attack data from the AI layer is forwarded for secure storage. Formally, the non-attack data of a smart home system will be stored in a buffer space or web storage, where an attacker can perform several security attacks, such as data manipulation and data injection. For that reason, secure storage is required, which is transparent and can tackle data integrity issues. Blockchain technology is a prominent solution to this issue, where we designed a smart contract that validates the incoming non-attack data. For that purpose, we designed a smart contract in the Remix development environment, comprising functions such as `addauthorized()`, `changedevicestate()`, `removeauthorization()`, and `currentdevicestate()`. The incoming non-attack data from the AI classifier are validated using these smart contract functions. The smart contract is attached with an on-site file system storage, i.e., IPFS, which allows the data to be stored in their secure storage systems. For that purpose, a Firebase application programming interface (API) is used that programmatically interacts with IPFS. The aforementioned smart contract functions take the data as a parameter and forward them to the IPFS. Once the validated data from the smart contract are uploaded to IPFS via Firebase, a unique content identifier (CID) is received to retrieve the content later.

Additionally, the IPFS computes the hash of the original data and forwards the hash to an immutable blockchain ledger. Here, we used an Ethereum-based public blockchain to obtain benefits such as transparency, decentralization, and immutability. As all entities of the smart home system have to register with the blockchain network, it makes the blockchain network transparent. Due to the blockchain's transparency property, one can find the entity that has performed the data manipulation, hence improving the security and privacy of the smart home system. Further, the data can be fetched from the IPFS node by computing its hash. If the computed and stored hash are the same, we can infer that the data are not manipulated; otherwise, we can simply discard that data and find the adversary behind this act. The entire smart contract and IPFS are deployed in a Sepolia-based test network to analyze the performance of the blockchain network.

4.4. Application Layer

The application layer receives the data from the blockchain layer, which is given as input to the other sensors available in the smart home system of that particular home in the cluster. If the nominal value stored in the network is found to be close to the predefined threshold values, the actuators present in the smart sensors will perform necessary actions to control the environment, preventing it from any possible hazardous scenarios in the system. Through the seamless coordination of sensor data and responsive actuation, the smart home sensors act as intelligent custodians, ensuring the safety and stability of the smart home system while minimizing risks and promoting operational efficiency. Figure 2 shows a sequential flow of the proposed framework.

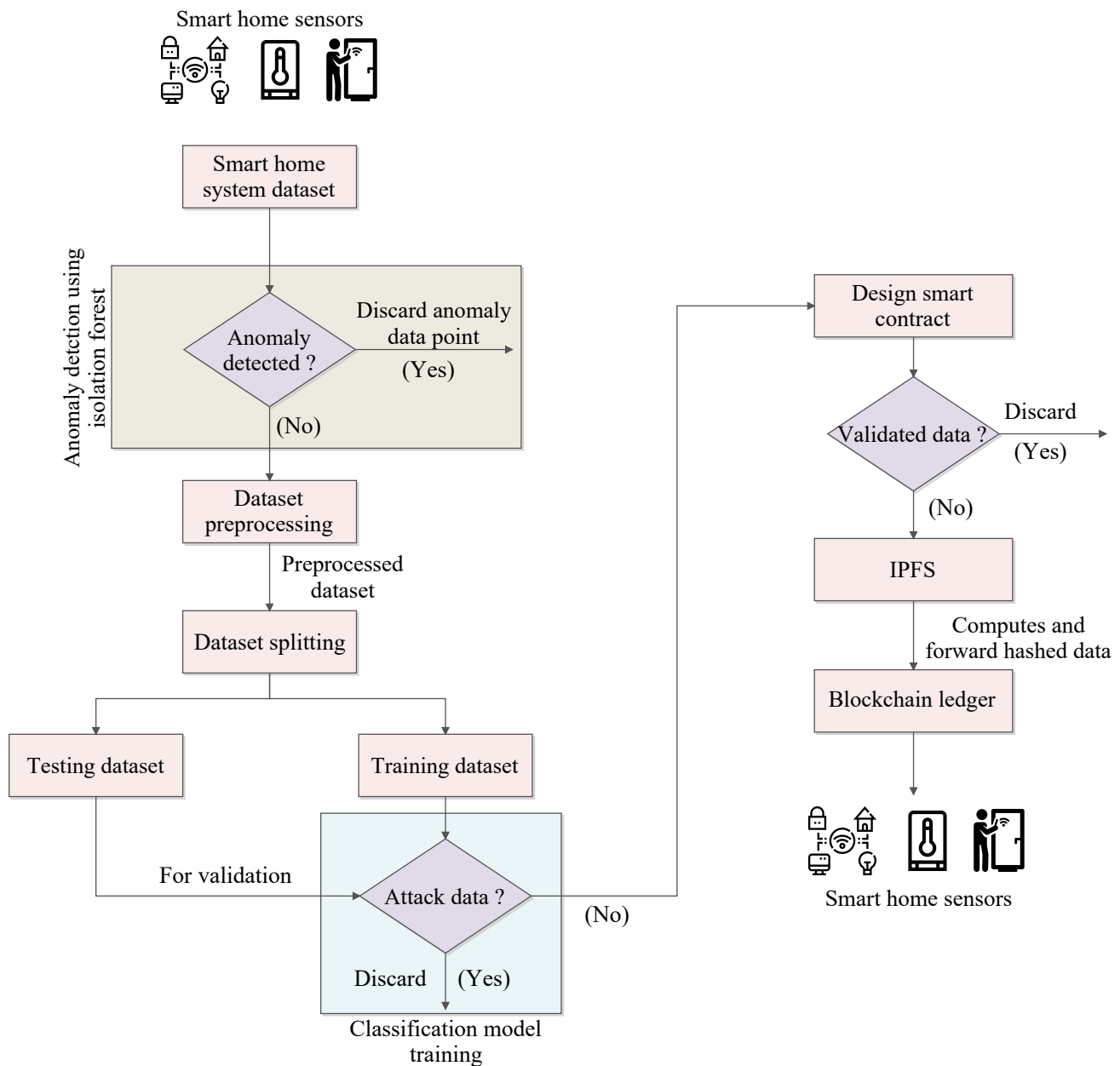


Figure 2. Sequential flow of the proposed framework.

5. Analysis of Results

This section discusses the analysis of the results of the proposed architecture using different performance parameters, such as statistical measures (e.g., accuracy, precision, recall, lift curve, and validation curve). Additionally, we present the experimental setup and tools showing the tools, libraries, and software platforms used to develop the proposed architecture.

5.1. Experimental Setup and Tools

The proposed architecture is developed using sophisticated tools, recent AI libraries, and open-source development platforms to write source code, train the AI algorithms, and visualize its performance. For that purpose, the anaconda distribution of version 6.3.0 is utilized, wherein the Jupyter Notebook is used to write the source code for data preprocessing, data modeling and training, and visualization. Further, different AI-based

libraries, such as Python 3.8.8, Pandas, Numpy, Matplotlib, Plotly, and Pycaret, are utilized in the proposed work. The Pandas library is used for data manipulation and preprocessing using functions such as `readcsv()`, `isna()`, and `value – counts()`. Next, the Numpy library is used for data computation, where the dataset is transformed into arrays for easy computation. We used the Pycaret library with user-defined functions for data modeling and training. Further, Plotly and Matplotlib were used for data visualization. For creating smart contracts, we utilize the Remix development environment with version 0.33.2. In Remix, we used solidity language with version 0.8.0 to design the smart contract. The smart contract comprises different user-defined functions—`addAuthorizedDevice()`, `removeAuthorizedDevice()`, `changeDeviceState()`, `deviceState()`, and `authorizedDevices`—that validate the non-attack data of the smart home system. These functions are compiled using a solidity compiler with version 0.8.18+commit.87f61d96. The proposed architecture is implemented on a system comprising 11th generation Intel(R) Core(TM) (i5-1135G7), 12 GB of random access memory (RAM), and an Intel Iris Xe graphic card. The system specification helps other readers to boost the training time and minimizes the processing time.

5.2. Discussion of Anomaly-Based Results

This section presents the results obtained for anomaly detection in smart home systems. Algorithms like LOF and IF are quite effective in detecting anomalies from real-world applications. For instance, LOF is a density and distance-based algorithm similar to the KNN algorithm, while IF is an ensemble method similar to random forests. The advantage of tree algorithms is that they offer essential benefits in finding anomalies in smart home systems.

Figure 3 illustrates the performance of the proposed framework in terms of the accuracy of detecting anomalies from the smart home system. The x-axis and y-axis represent the detection accuracy and the adopted anomaly-detection algorithms (i.e., IF and LOF) for the proposed framework. We used two different libraries to evaluate the performance of the detection algorithm: the IF algorithm from the SKlearn library (IF_SKL) and the IF algorithm from the Pycaret library (IF_Pycaret). From the graph, it is clear that IF (from IF_SKL) transcends the LOF, whereby IF (from IF_SKL) and LOF achieve 99.95% and 74.34%, respectively. Furthermore, the IF_Pycaret achieves 92.12% accuracy, which is better than the LOF. The hyperparameters play an essential role in lifting the model's performance; in that view, LOF uses the "number of neighbors" parameter to achieve 74.34% accuracy. However, as the number of neighbors increases, the computational complexity of the model increases. LOF has a high computational complexity, i.e., $\mathcal{O}(n)$, where n depends on the number of data sizes. Moreover, we used hyperparameters (e.g., number of neighbors) that were to be used in each iteration, resulting in increasing the computation complexity from $\mathcal{O}(n)$ to $\mathcal{O}(k \times n)$, where k is the number of neighbors. Contrary, the computational complexity of IF is $\mathcal{O}(n \log n)$ (without any hyperparameters), which is less than the LOF.

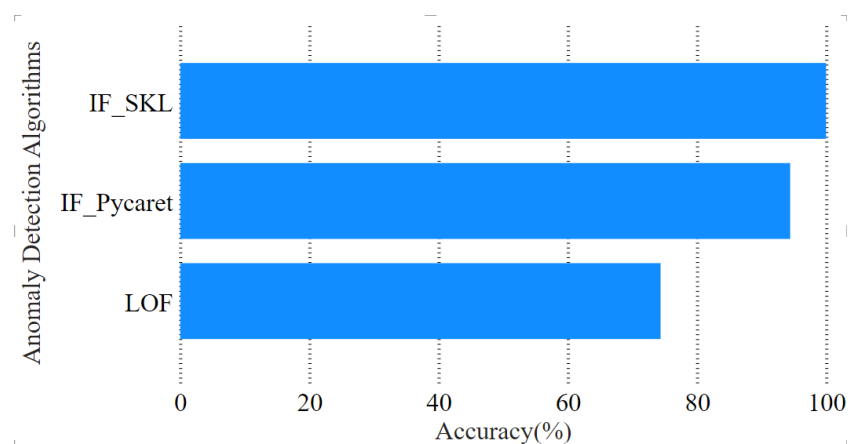


Figure 3. Accuracy of anomaly-detection algorithms.

Figure 4 illustrates the number of anomalous and non-anomalous points detected by the anomaly-detection algorithms. The blue-colored bars in Figure 4 represent the nominal points in the dataset. In contrast, the orange-colored bars in the graph represent the anomalous points in the dataset detected by the anomaly-detection algorithm. The IF algorithm gives the outcomes that are most accurate with respect to the results obtained from the dataset. The algorithm “IF” differed by three tuples in terms of anomalies from the original dataset, which resulted in a high accuracy. For the comparison of the tuples to determine the accuracy, the *where()* function of numpy is used, which displays the number of tuples between the outcomes of two algorithms that differ with the selected attribute. The attribute chosen for the comparison is the additional column of behavior 7 of the instance or the “Anomaly” column.

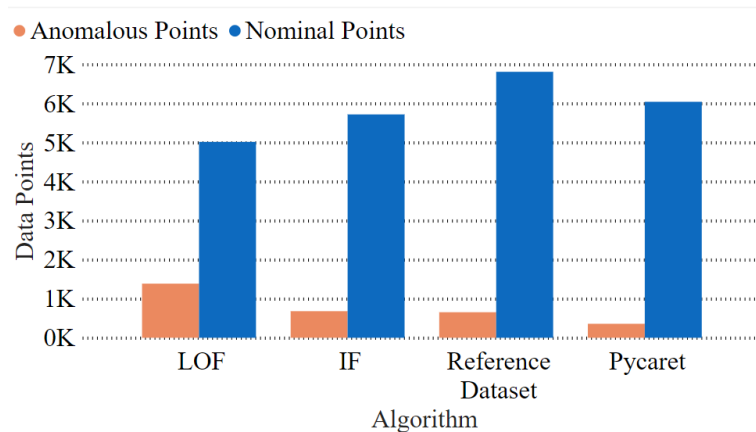


Figure 4. Nominal and Anomalous points.

5.3. Classification-Based Result Discussion

This subsection presents the results obtained through implementing different AI classifiers, such as NB, KNN, SVM, LDA, and QDA. Here, the classification algorithms are used to detect the behavior of the input value on every instance. We want to remark that the anomaly-detection algorithms are semi-supervised learning algorithms that are not preferred for the classification task. Thus, in such cases, AI-based classifiers prove to be more efficient for classification tasks as they lie in the supervised learning category. Figure 5 illustrates the comparison of the accuracy obtained for the classification algorithms implemented on the dataset, which includes the class labels, i.e., Anomaly and nominal, which are depicted as “1” and “0”. The accuracy of an AI classifier is formulated as follows.

$$Accuracy = \frac{\mu + \gamma}{\mu + \gamma + \theta + \varrho} \tag{21}$$

where μ , γ , θ , and ϱ represent the true positive, true negative, false positive, and false negative, respectively. The x-axis of the graph represents the algorithm applied, and the y-axis represents the accuracy of the applied AI algorithms. The train–test split function is applied to the dataset to evaluate and enhance detection performance. The model is trained on the training dataset and is tested on the remaining section of the dataset (i.e., the testing dataset). From the graph, it can be seen that the KNN algorithm gives the highest accuracy, which is 99.53%, compared to the other AI algorithms. This is because KNN is simple and relatively easy to implement; moreover, it does not need an explicit training phase, so the prediction for new data points is adjusted without retraining the model.

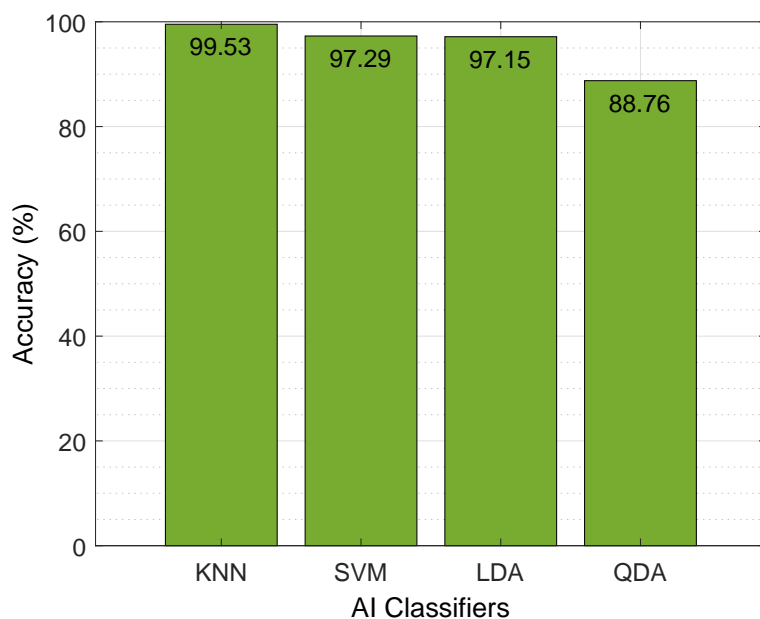


Figure 5. Accuracy of AI classifiers.

Figures 6–8 illustrate other result parameters used to analyze the performance of the KNN algorithm. Figure 6 shows the confusion matrix of the KNN algorithm. A confusion matrix is a statistical, matrix-based performance parameter used to summarize the performance of a classifier. The confusion matrix is built up using four features. These features are the evaluation parameters used to evaluate a classifier. The parameters integrated into the confusion matrix are as follows.

1. True Positive (μ): The true positive parameter represents the total number of positive outcomes that are correctly classified as positive with reference to the data.
2. False Positive (θ): The total number of negative outcomes that are incorrectly iterated as positive outcomes by the algorithm is the false positive category.
3. True Negative (γ): The number of correct negative classified outcomes falls under the true negative category.
4. False Negative (ϱ): The false negative parameter refers to the total number of incorrectly predicted negative outcomes that are supposedly positive outcomes.

		Predicted Value	
		Anomalous	Nominal
True Value	Anomalous	203	13
	Nominal	0	1705

Figure 6. Confusion matrix.

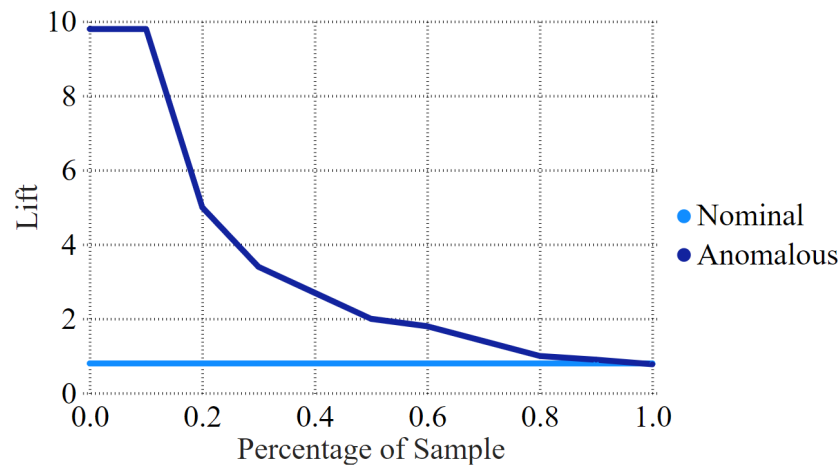


Figure 7. Lift Curve.

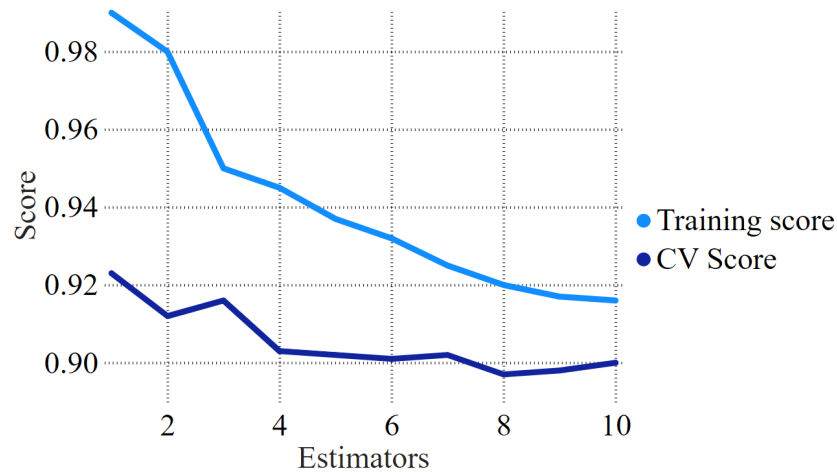


Figure 8. Validation Curve.

The true positive, false positive, true negative, and false negative values were found to be 203, 13, 0, and 1705, respectively. Through these features, the accuracy, precision, and recall of the model can be calculated. Figure 7 illustrates the lift curve of the KNN algorithm. It is observed that the x-axis represents the fraction of the sample of the data iterated data that corresponds to the lift, represented on the y-axis of the plot. The lift is calculated as the ratio of positive outcomes on the selected sample point, divided by the ratio of positive outcomes present in the whole iterated dataset. When the data available are ordered, the algorithm with the highest probability will appear on the left of the graph, along with the highest lift scores. A lift curve for a model can be defined as ideal when there are several real positive labels in a fraction of the number that has a very high probability of being positive. The model with the maximum lift is preferably considered as the better-iterated model. The lift curve for the model represented in Figure 7 is near the ideal condition given the parameters for the analysis of the lift curve.

Figure 8 illustrates the validation curve for the KNN algorithm. A validation curve is a graphical performance metric to evaluate an iterated model based on the hyperparameters defined in the model. The validation curve and the training curve look similar to each other in an ideal condition. If both scores of the curves are established to be low, the iterated model is determined to be underfitting for the situation. The underfitting condition arises when too much regularization occurs or the model is informed by a few features in the condition. When the training curve reaches a higher score quickly in comparison to the validation curve, the model is established as overfitting for the condition. Further, the model can be evaluated for the overfitting conditions, i.e., if the lift curve shows a significant lift for anomalous instances compared to the baseline, it suggests that the model incorrectly

captures or classifies anomalous data as nominal. This indicates overfitting and a lack of generalization to unseen anomalous patterns. The closer the nominal and anomalous curves are, the better the model's performance.

The curve in Figure 8 can be said to be the ideal condition, as both the curves are near to each other and present no overfitting or underfitting conditions. The non-attack data of the smart home systems are stored on different storage platforms for offering varied services to the smart home systems. However, it can be tampered with via data injection and manipulation attacks. Therefore, the proposed work adopts the indispensable characteristics of blockchain technology, where a smart contract is deployed on an Ethereum-based public blockchain.

5.4. Discussion of Blockchain-Based Results

The designed smart contract has various user-defined functions, such as `addauthorizeddevice()`, `changedevicestate()`, `removeauthorizeddevice()`, and `devicestate()` that act as a data validator that validates the non-attack data. In particular, the `authorizeddevice()` function includes a threshold-checking parameter. This parameter allows the smart contract to enforce a predefined threshold for authorized devices. The threshold could be a numerical value or a specific condition that needs to be met before a device is considered authorized. For example, the temperature sensor reading must lie within the threshold set by the regulatory bodies. If the sensor reading is out of the range, we invoke the `removeauthorizeddevice()` function to generate an alert and eliminate that particular device from the smart home system.

The purpose of this function is to ensure that only devices meeting the specified criteria can perform certain actions or access specific resources within the smart home system. By incorporating this threshold checking parameter into the `authorizeddevice()` function, the smart contract can enforce a more robust and secure authorization mechanism. This mechanism prevents unauthorized or potentially malicious devices from accessing sensitive data or performing unauthorized actions within the blockchain network. On successful validation, the non-attack data are forwarded to the IPFS-based secure storage. Figure 9 shows the deployed smart contract and its different user-defined functions. For deployment, we used an injected provider–metamask environment that offers different test networks for smart contract deployment. We utilized the Sepolia test network to deploy the smart contract shown in Figure 9.

When deploying a smart contract, there are two main costs to consider, i.e., transaction costs and execution costs. Transaction costs refer to the fees associated with interacting with the blockchain network to deploy a smart contract. These costs can vary depending on the blockchain platform being used and are typically paid in the native cryptocurrency of that platform, for example, Ethers in the Ethereum blockchain. Moreover, execution costs pertain to the computational resources required to execute the smart contract code once it is deployed on the blockchain. Here, we used an event log and struct to store the IPFS hash that has significant advantages in terms of gas consumption. Event logs are used to see the logged data that are not frequently retrieved. On the contrary, struct is used to enhance data retrieval; it organizes and stores data directly within the contract's state, making it accessible for on-chain operations. Figure 10 shows the transaction and execution cost incurred while deploying the smart contract in the Ethereum blockchain.

Further, we evaluated the performance of IPFS using scalability parameters. Since IPFS computes the hash of the legitimate non-attack smart home system data and forwards them to the Ethereum-based public blockchain, it improves the response time of the blockchain network. Response time is inversely proportional to the scalability parameter. Therefore, the lower the response time, the higher the scalability. Figure 11 shows the scalability improvement when IPFS is employed in the blockchain network. As the response time improves, more transactions can be granted, increasing the blockchain network's scalability.

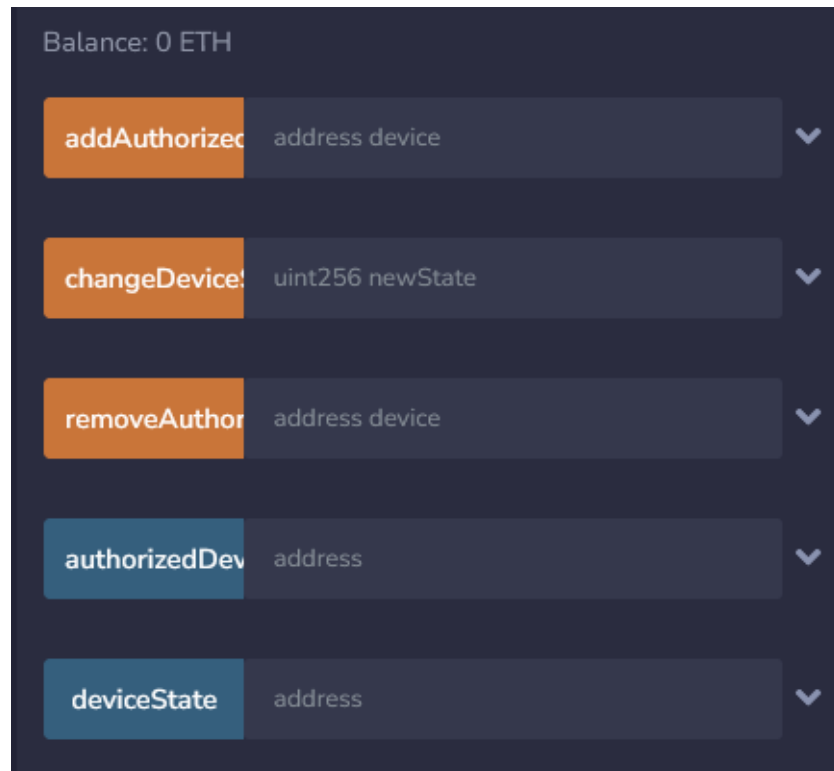


Figure 9. Blockchain’s smart contract functions.

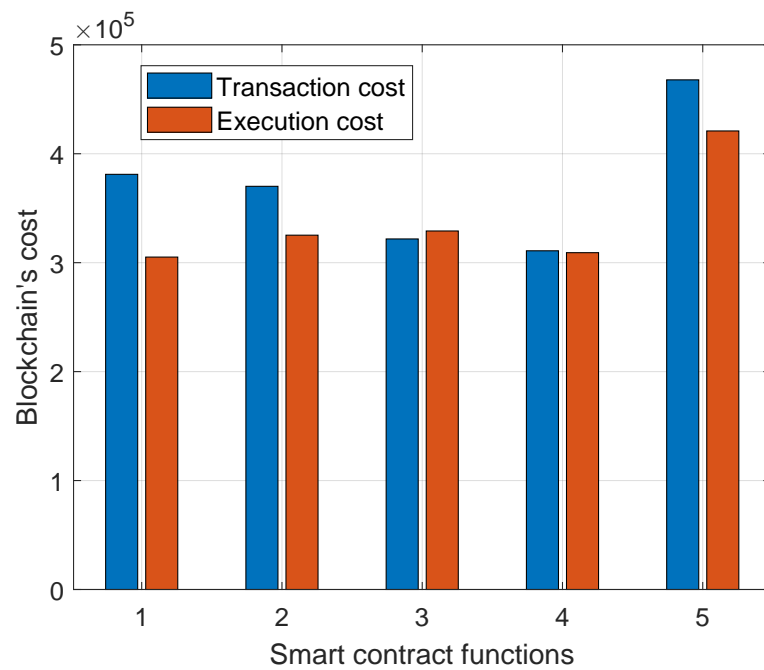


Figure 10. Blockchain’s transaction and execution cost.

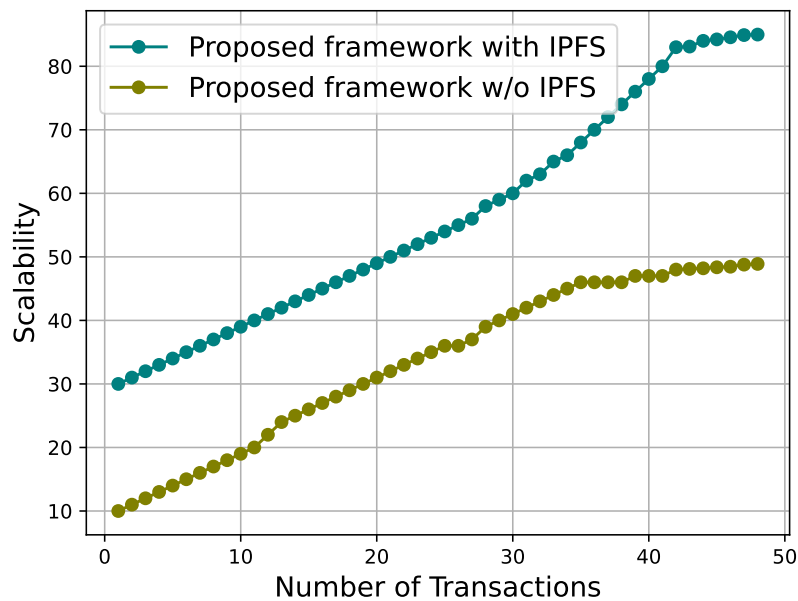


Figure 11. Scalability comparison.

6. Conclusions

The paper proposed a secure and intelligent framework to handle security threats associated with smart home systems. It is observed from the literature that due to the openness of the network interface, adopting weak protocols and lightweight encryption, the attacker can leverage these benefits and maneuver the performance of the smart home system. The proposed framework employs the automation and intelligent property of the AI algorithms, which are first trained to eradicate anomalous data from smart home systems. This is accomplished by training an IF algorithm that uses an ensemble approach to pinpoint and eliminate anomalous data points within the dataset with exceptional accuracy (99.27%). Once the anomalous data are eliminated, the AI classifiers are trained to classify attack and non-attack data. The proposed framework discards the attack data and only allows non-attack data to assist in enhancing the performance of the smart home system. Furthermore, to strengthen the security of the smart home system, the non-attack data are forwarded to the immutable blockchain nodes. For that purpose, we designed a smart contract in the Remix development environment that validates the non-attack smart home system data and deploys them on the Ethereum-based public blockchain. The smart contract is connected to the IPFS that stores the non-attack data. The IPFS computes the hash of the original non-attack data and forwards them to the blockchain's immutable ledger. Storing the non-attack data of smart homes in blockchain nodes reduces the chance of data manipulation. The results show that the performance of the proposed framework is better than the existing state-of-the-art work. Here, the IF and KNN algorithms offer 99.53% and 99.27% accuracy in detecting anomalous and attack data, respectively. Moreover, the incorporation of the IPFS with the blockchain network improves the response time and scalability of smart home systems.

Adopting blockchain technology can degrade the latency and increase the mining cost. To respond to this challenge, we utilized IPFS and event logs to minimize the mining cost. However, we want to remark that the mining cost is still a persistent challenge and a significant limitation that necessitates careful consideration. In future work, we will utilize the proof-of-stake (PoS) and hybrid approaches, which aim to reduce energy consumption and lower the barriers to entry for participants. These innovations seek to strike a balance between security, decentralization, and cost-effectiveness, thereby making proposed work

more sustainable and accessible. In addition, we will also incorporate the essential benefits of a 5G network interface to enhance the latency of the proposed framework.

Author Contributions: Conceptualization: S.T., K.S., C.P., F.A. and A.T.; Writing—Original draft preparation: A.T., N.K.J., S.T. and A.S.; methodology: S.T., C.P., F.A., K.S. and N.K.J.; Writing—Review and editing: A.T., A.S., C.P., K.S., C.P. and N.K.J.; Investigation: K.S., F.A., N.K.J. and S.T.; Visualization: S.T., C.P., K.S., F.A., A.S. and A.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Researchers Supporting Project Number (RSP2023R509), King Saud University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No data are associated with this research work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khan, M.N.; Rao, A.; Camtepe, S. Lightweight Cryptographic Protocols for IoT-Constrained Devices: A Survey. *IEEE Internet Things J.* **2021**, *8*, 4132–4156. [\[CrossRef\]](#)
2. Ramirez, J.; Pedraza, C. Performance analysis of communication protocols for Internet of things platforms. In Proceedings of the 2017 IEEE Colombian Conference on Communications and Computing (COLCOM), Cartagena, Colombia, 16–18 August 2017; pp. 1–7. [\[CrossRef\]](#)
3. Tawalbeh, L.; Muheidat, F.; Tawalbeh, M.; Quwaider, M. IoT Privacy and Security: Challenges and Solutions. *Appl. Sci.* **2020**, *10*, 4102. [\[CrossRef\]](#)
4. Aldahmani, A.; Ouni, B.; Lestable, T.; Debbah, M. Cyber-Security of Embedded IoTs in Smart Homes: Challenges, Requirements, Countermeasures, and Trends. *IEEE Open J. Veh. Technol.* **2023**, *4*, 281–292. [\[CrossRef\]](#)
5. Khalid, M.H.; Murtaza, M.; Habbal, M. Study of Security and Privacy Issues in Internet of Things. In Proceedings of the 2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA), Sydney, Australia, 25–27 November 2020; pp. 1–5. [\[CrossRef\]](#)
6. Chanal, P.M.; Kakkasageri, M.S. Security and privacy in IoT: A survey. *Wirel. Pers. Commun.* **2020**, *115*, 1667–1693. [\[CrossRef\]](#)
7. Gunathilake, N.A.; Buchanan, W.J.; Asif, R. Next Generation Lightweight Cryptography for Smart IoT Devices: Implementation, Challenges and Applications. In Proceedings of the 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 15–18 April 2019; pp. 707–710. [\[CrossRef\]](#)
8. Latif, M.A.; Ahmad, M.B.; Khan, M.K. A Review on Key Management and Lightweight Cryptography for IoT. In Proceedings of the 2020 Global Conference on Wireless and Optical Technologies (GCWOT), Malaga, Spain, 6–8 October 2020; pp. 1–7. [\[CrossRef\]](#)
9. Kumar, A.; Jain, V.; Yadav, A. A New Approach for Security in Cloud Data Storage for IOT Applications Using Hybrid Cryptography Technique. In Proceedings of the 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Mathura, India, 28–29 February 2020; pp. 514–517. [\[CrossRef\]](#)
10. Anthi, E.; Williams, L.; Słowińska, M.; Theodorakopoulos, G.; Burnap, P. A Supervised Intrusion Detection System for Smart Home IoT Devices. *IEEE Internet Things J.* **2019**, *6*, 9042–9053. [\[CrossRef\]](#)
11. Chowdhry, D.; Paranjape, R.; Laforge, P. Smart home automation system for intrusion detection. In Proceedings of the 2015 IEEE 14th Canadian Workshop on Information Theory (CWIT), St. John's, NL, Canada, 6–9 July 2015; pp. 75–78. [\[CrossRef\]](#)
12. Alghayadh, F.; Debnath, D. A Hybrid Intrusion Detection System for Smart Home Security. In Proceedings of the 2020 IEEE International Conference on Electro Information Technology (EIT), Chicago, IL, USA, 31 July–1 August 2020; pp. 319–323. [\[CrossRef\]](#)
13. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 1–22. [\[CrossRef\]](#)
14. Bashir, U.; Chachoo, M. Intrusion detection and prevention system: Challenges & opportunities. In Proceedings of the 2014 International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 5–7 March 2014; pp. 806–809. [\[CrossRef\]](#)
15. Mohammad, Z.N.; Farha, F.; Abuassba, A.O.M.; Yang, S.; Zhou, F. Access control and authorization in smart homes: A survey. *Tsinghua Sci. Technol.* **2021**, *26*, 906–917. [\[CrossRef\]](#)
16. Alam, M.R.; Reaz, M.B.I.; Ali, M.A.M. A Review of Smart Homes—Past, Present, and Future. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 1190–1203. [\[CrossRef\]](#)
17. Alalade, E.D. Intrusion Detection System in Smart Home Network Using Artificial Immune System and Extreme Learning Machine Hybrid Approach. In Proceedings of the 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2–16 June 2020; pp. 1–2. [\[CrossRef\]](#)

18. Tang, S.; Gu, Z.; Yang, Q.; Fu, S. Smart Home IoT Anomaly Detection based on Ensemble Model Learning From Heterogeneous Data. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 4185–4190. [CrossRef]
19. Bodkhe, U.; Tanwar, S.; Bhattacharya, P.; Kumar, N. Blockchain for precision irrigation: Opportunities and challenges. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4059. [CrossRef]
20. She, W.; Gu, Z.H.; Lyu, X.K.; Liu, Q.; Tian, Z.; Liu, W. Homomorphic Consortium Blockchain for Smart Home System Sensitive Data Privacy Preserving. *IEEE Access* **2019**, *7*, 62058–62070. [CrossRef]
21. Baucas, M.J.; Gadsden, S.A.; Spachos, P. IoT-Based Smart Home Device Monitor Using Private Blockchain Technology and Localization. *IEEE Netw. Lett.* **2021**, *3*, 52–55. [CrossRef]
22. Cultice, T.; Ionel, D.; Thapliyal, H. Smart Home Sensor Anomaly Detection Using Convolutional Autoencoder Neural Network. In Proceedings of the 2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), Chennai, India, 14–16 December 2020; pp. 67–70. [CrossRef]
23. Lee, Y.; Rathore, S.; Park, J.; Park, J. A blockchain-based smart home gateway architecture for preventing data forgery. *Hum.-Centric Comput. Inf. Sci.* **2020**, *10*, 9. [CrossRef]
24. Haddadpajouh, H.; Khayami, R.; Dehghantanha, A.; Choo, K.K.R.; Parizi, R. AI4SAFE-IoT: An AI-powered secure architecture for edge layer of Internet of things. *Neural Comput. Appl.* **2020**, *32*, 16119–16133. [CrossRef]
25. Jain, P.; Jain, S.; Zaïane, O.R.; Srivastava, A. Anomaly Detection in Resource Constrained Environments with Streaming Data. *IEEE Trans. Emerg. Top. Comput. Intell.* **2022**, *6*, 649–659. [CrossRef]
26. Alrubei, S.M.; Ball, E.; Rigelsford, J.M. A Secure Blockchain Platform for Supporting AI-Enabled IoT Applications at the Edge Layer. *IEEE Access* **2022**, *10*, 18583–18595. [CrossRef]
27. Xiao, W.; Liu, C.; Wang, H.; Zhou, M.; Hossain, M.S.; Alrashoud, M.; Muhammad, G. Blockchain for Secure-GaS: Blockchain-Powered Secure Natural Gas IoT System with AI-Enabled Gas Prediction and Transaction in Smart City. *IEEE Internet Things J.* **2021**, *8*, 6305–6312. [CrossRef]
28. Ahmed, I.; Zhang, Y.; Jeon, G.; Lin, W.; Khosravi, M.R.; Qi, L. A blockchain- and artificial intelligence-enabled smart IoT framework for sustainable city. *Int. J. Intell. Syst.* **2022**, *37*, 6493–6507. [CrossRef]
29. Menon, S.; Anand, D.; Kavita.; Verma, S.; Kaur, M.; Jhanjhi, N.Z.; Ghoniem, R.M.; Ray, S.K. Blockchain and Machine Learning Inspired Secure Smart Home Communication Network. *Sensors* **2023**, *23*, 6132. [CrossRef]
30. Chauhan, K.; Jani, S.; Thakkar, D.; Dave, R.; Bhatia, J.; Tanwar, S.; Obaidat, M.S. Automated Machine Learning: The New Wave of Machine Learning. In Proceedings of the 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 5–7 March 2020; pp. 205–212. [CrossRef]
31. Qashlan, A.; Nanda, P.; He, X.; Mohanty, M. Privacy-Preserving Mechanism in Smart Home Using Blockchain. *IEEE Access* **2021**, *9*, 103651–103669. [CrossRef]
32. Arif, S.; Khan, M.A.; Rehman, S.U.; Kabir, M.A.; Imran, M. Investigating Smart Home Security: Is Blockchain the Answer? *IEEE Access* **2020**, *8*, 117802–117816. [CrossRef]
33. Yiyang, C.; Takashio, K. A Preliminary Study for the Ethereum Blockchain-Based Smart Home Systems. In Proceedings of the 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 19–21 October 2022; pp. 71–76. [CrossRef]
34. Farooq, M.S.; Khan, S.; Rehman, A.; Abbas, S.; Khan, M.A.; Hwang, S.O. Blockchain-Based Smart Home Networks Security Empowered with Fused Machine Learning. *Sensors* **2022**, *22*, 4522. [CrossRef] [PubMed]
35. Bera, B.; Mitra, A.; Das, A.K.; Puthal, D.; Park, Y. Private Blockchain-Based AI-Envisioned Home Monitoring Framework in IoMT-Enabled COVID-19 Environment. *IEEE Consum. Electron. Mag.* **2023**, *12*, 62–71. [CrossRef]
36. Păvăloaia, V.D.; Necula, S.C. Artificial Intelligence as a Disruptive Technology: A Systematic Literature Review. *Electronics* **2023**, *12*, 1102. [CrossRef]
37. Yang, J.; Sun, L. A Comprehensive Survey of Security Issues of Smart Home System: “Spear” and “Shields,” Theory and Practice. *IEEE Access* **2022**, *10*, 124167–124192. [CrossRef]
38. Buil-Gil, D.; Kemp, S.; Kuenzel, S.; Coventry, L.; Zakhary, S.; Tilley, D.; Nicholson, J. The digital harms of smart home devices: A systematic literature review. *Comput. Hum. Behav.* **2023**, *145*, 107770. [CrossRef]
39. Moustafa, N. ToN IoT Datasets. 2019. Available online: <https://iee-dataport.org/documents/toniot-datasets> (accessed on 19 September 2022). [CrossRef]
40. Tanwar, S.; Ramani, T.; Tyagi, S. Dimensionality reduction using PCA and SVD in big data: A comparative case study. In Proceedings of the Future Internet Technologies and Trends: First International Conference (ICFIT 2017), Surat, India, 31 August–2 September 2017; Springer: Berlin/Heidelberg, Germany, 2018; pp. 116–125.
41. Jadav, N.K.; Gupta, R.; Alshehri, M.D.; Mankodiya, H.; Tanwar, S.; Kumar, N. Deep Learning and Onion Routing-Based Collaborative Intelligence Framework for Smart Homes Underlying 6G Networks. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 3401–3412. [CrossRef]

42. Dilraj, M.; Nimmy, K.; Sankaran, S. Towards Behavioral Profiling Based Anomaly Detection for Smart Homes. In Proceedings of the TENCON 2019—2019 IEEE Region 10 Conference (TENCON), Kochi, India, 17–20 October 2019; pp. 1258–1263. [[CrossRef](#)]
43. Kang, K.; Xu, L.; Wang, W.; Wu, G.; Wei, J.; Shi, W.; Li, J. A Hierarchical Automata Based Approach for Anomaly Detection in Smart Home Devices. In Proceedings of the 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), Rhodes, Greece, 2–6 November 2020; pp. 1–8. . [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.