EVENT RECONSTRUCTION IN A DIGITAL FORENSICS INVESTIGATION MODEL FOR A SMART HOME

SATIA ELVINE SAIKWA

A Thesis Submitted to the Institute of Postgraduate Studies in Partial Fulfillment of the Requirements of the Master of Science in Information Technology Security and Audit, Kabarak University

KABARAK UNIVERSITY

NOVEMBER, 2025.

DECLARATION

1. I do hereby declare that:
   i. This thesis is my own work, and to the best of my knowledge, it has not been presented for the award of a degree in any university or college.
   ii. That the work has not incorporated material from other works or a paraphrase of such material without due and appropriate acknowledgement
   iii. The work has been subjected to anti-plagiarism checks and meets Kabarak University's 15% similarity threshold.
2. I understand that issues of academic integrity are paramount, and therefore, I may be suspended or expelled from the University, or my degree may be recalled for academic dishonesty or any other related academic malpractices.

Signed:  Date: 14/11/2025

Elvine Saikwa Satia GMIA/NE/0225/01/18

RECOMMENDATION

To the Institute of Postgraduate Studies:

The research thesis entitled "***Event Reconstruction in a Digital Forensics Investigation Model for a Smart Home,***" written by Elvine Saikwa, is presented to the Institute of Postgraduate Studies of Kabarak University. We have reviewed the research proposal and recommend its acceptance in partial fulfillment of the requirements for the award of the degree of Master of Science in IT Security Audit.

Signed: _____ Date: 14/11/2025

Prof. Simon Karume,

Department of Computer Science and Information Technology,

Kabarak University.

Signed: _____ Date: _____: 14/11/2025

Dr. Nelson Masese,

Department of Computer Science and Information Technology,

Kabarak University.

ACKNOWLEDGEMENTS

I wish to thank Almighty God for the gift of life and good health He has granted me throughout my studies. His grace has been sufficient. I want to thank my supervisors, Prof. Simon M. Karume and Dr. Nelson Masese, who are exceptional scholars and great mentors. Their scholarly insights were both invaluable and inspirational. I would also like to acknowledge my classmate, Irvin Kilot, for the constant encouragement and wise words that helped me shape parts of this study. In addition, I would like to acknowledge my parents and siblings for their prayers and support throughout my studies. I would also like to thank the entire Kabarak University, School of Science, Engineering, and Technology, for the opportunity to study at Kabarak and for the support they accorded me during my studies.

DEDICATION

To my parents, Dr. Emmanuel Satia and Edith Satia. Thank you for the gift of education.

ABSTRACT

The use of Internet of Things technology has made life more comfortable for human beings. This is evidenced by the various applications of this technology, such as smart homes. A typical smart home comprises multiple devices that communicate seamlessly with one another. As a result, physical things can share and collect data. The outcome of the communication is that a lot of data is generated and shared, but not stored. This makes it very difficult to conduct a digital forensic investigation when a cybercriminal commits an attack. Moreover, reconstructing events that occurred poses a security challenge. Therefore, this study sought to address this challenge by developing a digital forensic investigation model for a smart home that incorporates event reconstruction. Accordingly, various methodologies, such as systematic literature review, design science, and rapid prototyping, were used to conduct the study within the digital forensics model. The model includes key components such as user and device registration, real-time log management, and an event timeline. The study established that the model successfully captured real-time logs and generated an event timeline, thereby improving the level of certainty during the forensic investigation of a smart home. Despite the model's success, challenges arose with data synchronization, sensor accuracy, calibration, and hardware and software integration. The study concludes that incorporating event reconstruction in a digital forensic investigation model for a smart home can significantly reduce the time taken to conduct forensic investigations. This study contributes to the field by providing a practical solution for event reconstruction. It is recommended that IoT device manufacturers adopt the model to enhance the digital forensic capabilities of their devices, enforce adapted legal frameworks, expand research with advanced techniques, and build collaborative ecosystems to ensure smart homes produce reliable forensic evidence and strengthen digital investigations.

Keywords: *Internet of Things; Event Reconstruction; Digital Forensics; Digital Forensic Investigation, Smart-home*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

OPERATIONAL TERMS

Cyber security	The practice of protecting computer systems, networks, and digital assets from threats such as unauthorized access, data breaches, cyberattacks, and other malicious activities.
Digital forensics	The process used to acquire, preserve, analyze, and report on evidence using scientific methods that are demonstrably reliable, accurate, and repeatable, such that it may be used in judicial proceedings (SWGDE, 2014).
Internet of Things	A network of sensors, actuators, and smart objects designed to interconnect a wide range of items, encompassing both everyday and industrial objects. Its goal is to enhance their intelligence, programmability, and capacity to interact with both humans and each other.
IoT Forensics	The process of applying digital forensics investigation procedures in the IoT system.
Security incident	An occurrence that actually or potentially jeopardizes the confidentiality, integrity, or availability of an information system or the information the system processes, stores, or transmits, or that constitutes a violation or imminent threat of a breach of security policies, security procedures, or acceptable use policies.
Smart home	A residence equipped with various internet-connected devices and technologies that enable homeowners to automate and control different aspects of their living environment remotely.

LIST OF ABBREVIATIONS

API	Application Programming Interface
BLE	Bluetooth Low Energy
COAP	Constrained Application Protocol
DB	Database
DF	Digital Forensics
DFI	Digital Forensic Investigation
DSR	Design Science Research
DSRM	Design Science Research Methodology
DTLS	Datagram Transport Layer Security
FS	Forensic Science
HMAC	Hash-based Message Authentication Code
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol
IPV6	Internet Protocol Version 6
KUREC	Kabarak University Research Ethics Committee
LEA	Law Enforcement Agency
MQTT	Message Queuing Telemetry Transport
NACOSTI	National Commission for Science Innovation and Technology
SWGDE	Scientific Working Group on Digital Evidence

CHAPTER ONE

INTRODUCTION

1.1 Introduction

This chapter provides the background of the study and defines the main concepts used. It then states the statement of the problem, objectives, and research questions. Furthermore, the study's scope, significance, and assumptions are presented.

1.2 Background Information

Cybercrimes are on the rise, mainly because cybercriminals have become increasingly relentless in their pursuit of profit and disruption. (Cybersecurity Ventures, 2022), had projected that cybercrime was to cost the world $10.5 trillion by 2025. Additionally, newer technologies, such as the Internet of Things (IoT), have provided alternative avenues for cybercriminals to commit digital crimes. These crimes need to be investigated. Investigations of these crimes help the aggrieved party to know when the crime occurred, how it happened, and what was lost. Furthermore, they can put measures in place to mitigate against such attacks. Investigations of cybercrime are guided by Digital Forensics (DF), a subset of Forensic Science (FS) applied to digital information.

FS has been in existence for many decades. It concerns applying scientific methods to establish factual answers to legal problems (Årnes, 2018). Over the years, it has evolved to meet the needs of forensic investigators across various forensic science disciplines, including DF. DF can be described as the use of scientifically derived and validated methods for storing, gathering, validating, recognizing, evaluating, interpreting, recording, and presenting digital evidence from digital media to encourage or promote the reconstruction of criminal events or to help anticipate destructive behavior or operations scheduled (Palmer, 2001). Under it, other subcategories are specific to the areas they apply to. Examples include internet, device, and network forensics, among others. The use of digital forensics helps ensure that a court can consider the digital artifacts collected by forensic investigators.

Digital forensic investigation (DFI) enables the investigation process within DF. DFI is a particular form of investigation that allows the findings (digital evidence) to be admissible in a court of law by the empirical methods and techniques used (Michael Donovan, 2012). Therefore, DFI helps a qualified forensic investigator unearth probative information from a digital crime scene.

“IoT is a system consisting of networks of sensors, actuators, and smart objects whose purpose is to interconnect ‘all’ things, including everyday and industrial objects, in such a way as to make them intelligent, programmable, and more capable of interacting with humans and each other” (IEEE, 2015). The use of this technology has made life more comfortable. Various applications of IoT can be found across sectors of the economy, ranging from health care to the transport industry. Some examples include smart refrigerators, smart homes, smart meters, and smart irrigation, among others. The advent of these technologies, unfortunately, has led to newer ways of committing cybercrime. This is because, during the design of some IoT devices, “security is not the primary concern” (Kebande et al., 2017). This has led to cyber attackers exploiting the weaknesses of these IoT devices. A case in point is when the Mirai botnet facilitated a DDoS attack on a service provider (Dunlap, 2017). These kinds of attacks need to be investigated. However, many existing investigation models remain high-level (that is, conceptual without empirical validation or prototypes), with limited follow-up development after the initial proposal.

The world's population is growing. Consequently, the use of IoT technology has experienced exponential growth as more people have harnessed its power to improve their lives. According to the Gartner (2017) report, there were to be over 20 billion connected IoT devices by the year 2020. Also, IoT consumers were expected to average 2.70 devices per person. Nevertheless, the COVID-19 pandemic negatively impacted IoT. There was decreased interest in IoT consumer devices (Lueth, 2020), and spending on IoT in 2020 decreased. On the flip side, its growth was expected to rise by double digits in 2021 (International Data Corporation (IDC), 2020).

Digital event reconstruction is of critical importance in digital forensics. It assists a forensic expert in explaining why an object possesses specific characteristics (Carrier & Spafford, 2004). According to Jeyaraman & Atallah (2006), event reconstruction is the process of identifying the underlying conditions and reconstructing the sequence of events leading to a security incident. However, determining how an event occurred in the digital world is not easy. Because the digital world is highly dynamic, with technologies and protocols constantly evolving, many researchers (e.g., Gladyshev, 2004; Soltani & Seno, 2019) have proposed formal methods for reconstructing events.

Since the emergence of the Internet of Things (IoT), various benefits have been realized. For example, improved service efficiency and effectiveness, as well as greater flexibility for real-time monitoring (Brous & Janssen, 2015). For example, an organization transporting oil through a pipeline can harness the power of IoT to perform predictive preventive maintenance using sensors, helping avert issues before they escalate. Despite these benefits, players in the IoT field have also faced challenges in enforcing security procedures and implementing mitigation strategies.

As the use of IoT skyrockets, conducting digital forensics has become a challenge. In addition, IoT can be in the cloud, smart devices, among others. This has made the IoT ecosystem very dynamic, and various scholars have proposed digital forensic models to aid forensic investigations. This is evidenced by the following works: Babun et al. (2018); Goudbeek et al. (2018); Oriwoh et al. (2013); Perumal et al. (2015); Zia et al. (2017). This clearly shows that progress is being made in forensic investigation.

1.3 Problem Statement

The use of the Internet of Things is on the rise, as evidenced by its various applications. However, during the design of some of these products, like smart homes, security is not a significant priority. Therefore, devices that facilitate this technology can be exploited by cybercriminals to commit digital crimes without the owner's knowledge. This, in turn, makes a digital forensic investigation of these devices difficult because there will be no way to know who initiated which commands at any given time. Also, these IoT devices only pass data, and they do not store it. Thus, it is crucial to develop ways to mitigate cyber-attack risks on IoT devices, thereby reducing the cost and time required to conduct forensic investigations on those devices. Therefore, this study investigates the lack of event reconstruction in a digital forensic investigative model for a smart home. Hence, this study proposed to include event reconstruction in a digital forensic model for a smart home. The aim would be to help digital forensic investigators conduct a digital forensic investigation of a smart home with a degree of certainty.

1.4 Study Objectives

The general objective of this study was to include event reconstruction in a digital forensic investigation model for a smart home.

Specific Objectives

i. To investigate existing models of digital forensics for Internet of Things devices.

ii. To design a digital forensic investigation model for a smart home with event reconstruction.

iii. To implement a digital forensic investigation model for a smart home with event reconstruction.

iv. To evaluate the digital forensic investigation model for a smart home with event reconstruction.

1.5 Research Questions

i. What models of digital forensic investigation for Internet of Things devices exist?

ii. How can a digital forensic investigation model for a smart home with event reconstruction be designed?

iii. How can a digital forensic investigation model for a smart home with event reconstruction be implemented?

iv. How can a digital forensic investigation model for a smart home with event reconstruction be evaluated?

1.6 Justification of the study

The growth of IoT has been facilitated by its simplicity and low deployment cost (Baho & Abawajy, 2023). Accordingly, a study conducted by HP (2014) found that over 70% of IoT devices have security flaws that are vulnerable to a wide range of attacks. Moreover, given that the number of devices is expected to grow exponentially in the coming years, we must address security challenges, including privacy.

According to Gladyshev (2004), the reasons for event reconstruction are: reducing reasoning errors, automating analysis, and meeting legal requirements. Consequently, when event reconstruction is incorporated, one can obtain reliable evidence that withstands scrutiny in a court of law. As a result, no innocent person will be incarcerated in a court of law.

The primary beneficiary of this research is the digital forensic community. The reason is that they are the only people with the technical ability to conduct a digital forensic investigation when needed. Therefore, they can act as expert witnesses in court, provided that due process is followed during investigations. Furthermore, the model can serve as a quick reference tool for aspiring forensic investigators and law enforcement agencies.

1.7 Scope of the Study

This research limits itself to adding event reconstruction to a smart home environment. It sought to determine whether digital forensic investigations could be conducted within the given smart home environment in the event of an attack. Furthermore, the model developed only serves as a proof-of-concept product.

1.8 Assumptions of the Study

The research assumed that the investigator was unbiased during the digital forensic investigation in the smart home. In addition, a proper chain of custody for the potential digital evidence was maintained so that the evidence could be used in a court of law.

1.9 Summary

This chapter began by providing the study's background; it then stated the problem statement, objectives, research questions, justification, scope, and, finally, the study's assumptions.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter begins by expounding on existing models of digital forensics for the Internet of Things. Thereafter, it discusses the implementation of the system, the theoretical, and ends with the conceptual framework.

2.2 Existing digital forensic investigation models on IoT.

Many researchers have conducted studies on digital forensics in the Internet of Things. The studies have offered excellent insights into what is expected before and after digital forensic investigations.

The first model is "1-2-3 zone and next best thing triage" by Oriwoh et al. (2013). (Oriwoh et al., 2013) proposed increasing the effectiveness of investigations and reducing the time required to conduct them in the IoT environment. Furthermore, their two models work together. The decision on which model to choose first depends on the investigator. Accordingly, the 1-2-3 zone model assists investigators in answering the questions of *who, what, when*, and *how* an incident occurred. In contrast, the next-best-thing triage will specify where to look for evidence.

The 1-2-3 zone model has three zones: Zone 1 (internal network), Zone 2 (middle), and Zone 3 (external network). These zones help an investigator segment a particular IoT environment they are investigating. According to the authors, the internal network comprises the hardware, software, and networks where a security incident occurs. Therefore, decisions are made on the evidence to be collected and the devices to be seized. After that, the middle layer, which provides communication between zones one and three, is investigated. The probable origin of digital artefacts in this layer may include network intrusion devices and network records, among others. On the other hand, the external network is outside the crime scene, but evidence may also be acquired. The sources of evidence may include cloud service providers, mobile network providers' data (Oriwoh et al., 2013), among others.

The next best thing is triage, which specifies the areas where evidence can be sought. It provides alternatives to an investigator when obtaining proof is difficult. A good example is when a malicious intruder launches an attack using a virtual machine towards a target and then

powers off the virtual machine. Investigations on such an attack would be extremely tedious. However, triage would enable an investigator to identify other areas where traces of evidence may be found. While this approach improved the effectiveness of investigations, it has not been tested in a real-world setting (Oriwoh et al., 2013).

The second model is "Internet of Things Digital forensic investigation model: Top-down forensic approach methodology by Perumal et al., 2015". This model has seven major parts. The parts consist of planning, which co-occurs with obtaining authorization and a warrant. Thereafter, a suspected IoT platform used to commit a digital crime is identified. Within the IoT platform, when forensic investigators determine the communication medium and devices used to perpetrate a digital crime, they conduct a triage examination of the server cluster. Thereafter, they revert to the standard digital forensic process: maintaining a list of people who handle the evidence, conducting a workroom investigation, documenting the outcome, corroborating and protecting the evidence, and, lastly, archiving and storing the results. While this model provides a means to conduct forensic investigations on IoT platforms, it is high-level and untested.

The third model is "Forensic-aware IoT" (Zawoad & Hasan, 2015). In this model, the authors proposed to address IoT from two viewpoints. The first is implementing digital forensic processes when IoT is the subject or cause of an attack, while the second is determining new or unidentified truths via IoT infrastructure. Accordingly, the researchers focused on "specific research sub-problems in the IoT domain". The forensic domain of IoT is divided into three parts: devices, networks, and the cloud. An investigator should therefore look for evidence in these parts.

The proposed model comprises three parts. The first part is the secure evidence preservation module. In this module, the forensic evidence is stored. The confidentiality of the evidence is ensured through public-key encryption. In addition, evidence is stored in a central repository to facilitate its collection and examination. Data of forensic interest is grouped by device and owner. The second module is secure provenance. This module helps to ensure that the evidence obtained cannot be disputed. This is enabled by protecting the list of people who have had access to the evidence. Lastly, the integrity of evidence is maintained using secure provenance chaining (Zawoad & Hasan, 2015).

The last module in the model is the access to evidence via an API. The API is web-based, and API calls will be made to access evidence and provenance information. Only law enforcement agencies will have access to the evidence. Court prosecutors and the court will be the only parties with access to these APIs. Thus, when all components work together, conducting a forensic investigation in IoT becomes easier. The shortcoming of this model, however, is that it was developed at a high level (conceptual) and has not been tested (Zawoad & Hasan, 2015).

(Mahmood et al., 2024) Conducted a comparative study of several IoT forensic frameworks, including the Forensic-aware IoT model. He found that, although this model is conceptual and multi-phased, covering identification, preservation, and analysis, it does not address practical challenges such as heterogeneity and scalability. Mahmood stresses the need to develop these high-level models into tested and comprehensive frameworks.

The fourth framework is "A generic digital forensic investigation framework for Internet of Things (DFIF-IoT) by Kebande & Ray (2016). In this model, the authors leveraged the ISO/IEC 27043:2015 standard to develop their prototype. The framework comprises four divisions: proactive, IoT forensics, reactive, and concurrent processes. Concurrent processes include activities carried out throughout the investigative process. They involve obtaining authorization, in which an investigator or LEA seeks a court's permission to commence an investigation. Further, the evidence is documented, and a record of those who handle it is maintained. This helps increase the acceptability of potential digital artefacts in a court of law. Additionally, physical investigation and interaction with the devices are conducted.

The proactive process occurs before a security incident. The DFIF-IoT framework has six phases. It begins with an IoT scenario definition in which a specific source of evidence is identified through a risk assessment. Moreover, IoT crime scenes are identified, and incident planning is then done. This will assist an investigator in outlining the actions to take in the event of a security incident. Based on the risk assessment of a given IoT scenario, potential digital evidence is gathered according to a scientific process. Thereafter, the evidence is digitally preserved before any analysis is performed. To preserve evidence, ISO standards such as ISO/IEC 27037:2012 on handling potential digital evidence and ISO/IEC 10118-2:2010 are considered. Finally, the evidence may be stored for future analysis (Kebande & Ray, 2016).

(Rekha & Sudha, 2025) proposed a blockchain-assisted IoT forensic framework that enhances this preservation process by ensuring the authenticity and secure transmission of forensic data

within IoT environments. Their approach addresses volatility and integrity challenges in evidence, enabling the safe handling of evidence from devices to the cloud and the fog.

The next phase is IoT forensics. The forensics aspects involve cloud, network, and devices. When evidence is sought from the three, one can establish how an attack was propagated. In most cases, information such as network logs can be extracted from the network, cloud access logs from the cloud environment, and physical device information from the devices.

The last phase is the reactive process. This phase begins after an incident is detected and it involves three processes: initialization, acquisitive, and investigative. The initialization step begins immediately after incident detection and includes activities such as incident detection and planning. Additionally, the acquisitive step involves obtaining potential digital evidence from a given IoT environment. Furthermore, the collection, protection, and carriage of forensic artefacts are carried out. The final step is the investigative step, which involves analyzing evidence to unearth security incidents. Additionally, the evidence is interpreted and reported. Although the DFIF-IoT provides valuable insights into investigations, it is still in its preliminary stages, is high-level, and lacks a prototype demonstrating that the framework actually works (Kebande & Ray, 2016).

(Silva et al., 2025) Performed a systematic review of IoT forensic process models and reinforced the identification, preservation, and analysis phases as integral components. Their study underscored ongoing challenges, including device heterogeneity, volatile data, and legal/jurisdiction complexities, that existing frameworks, including DFIF-IoT, have yet to address fully.

The fifth model is “Application-Specific Digital Forensics Investigative Model in Internet of Things (IoT) by Zia et al. (2017). In this investigative model, the authors proposed examining industry-established guidelines and practices for IoT in digital forensics. The researchers highlighted the various types of artefacts that will be of importance during an investigation. Furthermore, they argue that different IoT applications will require different ways of investigation. However, the underlying principles should be the same. Hence, an investigation should look beyond the traditional investigation techniques. They, therefore, proposed a three-part investigation model.

The proposed investigation model has three independent parts: application-specific forensics, digital forensics, and the forensic process. Information flux among the elements is not a

concern. In most cases, information will move from the application forensics to the digital forensics to the forensic procedure. In the application forensics, the IoT application under investigation is identified. Then, to continue the investigation, forensic issues specific to the application are considered. This enables the investigator to select the appropriate method for extracting information (Zia et al., 2017).

The digital forensics phase involves network, device, and cloud forensics. Various artifacts of importance can be recovered from the three layers. Elements such as timeline logs to network logs can be identified. This significantly eases the forensic process. However, the complexity that arises is cloud forensics. This is due to concerns such as cross-jurisdictional laws. Lastly, the forensic process uses information from the two components to produce reliable evidence. It follows a systematic scientific process. However, the model did not focus on pre-incident detection (Zia et al., 2017).

The sixth framework is “A forensic investigation framework for a smart home environment " (Goudbeek et al., 2018). In this framework, researchers set out to investigate how forensic investigation could be done in a smart home environment. Their target was devices with more processing and computational power. Furthermore, the authors explained the challenges encountered in forensic investigations of a smart home. The main issue was identifying a smart home environment for a forensic investigator upon arrival at a potential crime scene. Accordingly, they proposed a seven-phase framework to address the investigation issue.

The proposed framework has seven phases. In each phase, different activities are carried out. The first phase is preparation. Usually, this step is done before an investigation begins. An investigator must familiarize themselves with existing technologies and procedures to conduct an investigation. Moreover, they should validate and verify the tools they will use. The second phase involves searching for the home automation system at the crime scene. All activities necessary to identify a home automation system, such as verifying the presence of sensors and actuators, should be carried out. The third phase concerns preserving the crime scene as it is. This enables the admissibility of digital evidence, as it would not be interfered with. Also, devices should be seized (Goudbeek et al., 2018).

The fourth phase focuses on understanding the workflow of the identified home automation system. A detective should be able to generate a global map of that system and its information flow. This is achieved using the information collected in earlier phases. In the fifth phase, the

system's security level is checked. Factors such as known vulnerabilities and access controls should be considered for the given system. The sixth phase involves locating and acquiring evidence. Evidence acquisition depends on information acquired in earlier stages. The forensic investigator should use the relevant tools and techniques to gather evidence. In the process, the reliability of the evidence should be checked. Therefore, (Goudbeek et al., 2018). propose that the investigator test whether the home automation system is working correctly. Lastly, the evidence should be analyzed to draw a logical conclusion from the data obtained.

The mentioned structure provided good insights on where to obtain digital artefacts. The artefacts are primarily log data from the devices in the smart home. One major challenge is finding potential digital evidence (log information) in the proprietary file system of a given smart home. While this model provides valuable insights into the forensic investigation of a smart home, it focuses on the aftermath of the incident.

The seventh framework is “IOT Forensic: A digital investigation framework for IoT systems (Sathwara et al., 2019). In this framework, the authors’ objective was to “study components of IoT devices to support digital investigations and to tackle emerging challenges in digital forensics”. In accordance, they alluded that an IoT forensic ecosystem did not exist. Thus, they used three stacks in parallel to explain their forensic ecosystem of IoT. Layer one consists of the architecture of IoT; layer two represents the elements that make up IoT; and layer three represents the likely forensic alternatives. In addition, the authors proceeded to expound on the various evidence that could be excavated from the multiple layers of the ecosystem. The evidence may range from network logs and memory analysis to service-level agreements and fingerprint collection.

Furthermore, the authors propose a three-step forensic approach for conducting an IoT investigation. The steps comprise identification, preservation, and analysis. It is important to note that they focused on end devices. During the identification stage, a detective must determine whether IoT is present in a given environment. The challenge posed here is that IoT devices are usually small, making their identification difficult. Thus, the problem that arises at this step is the lack of adequate logs compared to those in other computer networks. The protection step focuses on preserving the crime scene. However, given the dynamic and heterogeneous nature of IoT, maintaining the digital crime scene is much more challenging than in a traditional computer system. Consequently, the challenge that arises in this step is the

lack of tools to preserve information from sensing devices. Lastly, the third step of analysis entails gathering evidence to identify how an attack was propagated. Nevertheless, identification of an attacker becomes difficult because limited information is stored in logs and caches. However, this model has not been tested (Sathwara et al., 2019).

(Silva et al., 2025) Reinforced these findings in their systematic review by emphasizing the critical role of identification, preservation, and analysis phases while outlining persisting challenges such as device heterogeneity, volatile evidence, and legal complexities in IoT forensic investigations.

The eighth model is "Internet of Things Digital Forensic Investigation Using Open-Source Gears" by Al-Sadi et al. (2018). The authors proposed investigating how open-source digital forensics tools could be used to conduct IoT investigations. They also identified the open-source tools, as shown in Table 1, that can aid in conducting investigations. Accordingly, they proposed a layered architecture to guide the inquiry of IoT devices. The architecture has three layers: top, middle, and bottom. The top layer of the model comprises the infrastructure, which is the cloud or resembles the IoT cloud infrastructure. It represents the command, control, and monitoring center of IoT. The researchers identify application servers, IoT operational data, and other components as part of the infrastructure.

The second layer is the network, which handles all communication in the IoT environment. Communication can occur between the devices themselves and the application server. Additionally, the authors include the IoT pillars as defined by Cisco Systems. The pillars are application enablement, cyber and physical security, data analytics, network connectivity, IoT and fog applications, management, and automation. The said pillars represent the different aspects of digital forensics. Thus, an investigator, bearing the pillars in mind, should be able to join all the dots during an investigation to unearth potential crimes (Al-Sadi et al., 2018).

The third layer comprises the devices. In some cases, devices send data to the cloud or store it locally. Therefore, the investigations should be tailored towards the working of the specific IoT environment. Although the model provided valuable insights into open-source tools that can aid forensic investigations of IoT, it was not tested. Further, they presented an open-source toolset for conducting digital forensics, as shown in Table 1.

Table 1
Open-source digital forensics tool Source (Al-Sadi et. al, 2018)

No	Name of Open-Source Tools	Description	Website	Specialty
1	Autopsy	Suitable for the bottom layer (IoT devices).	www.sleuthkit.org/autopsy	Windows and Unix Disks and file systems
2	Wireshark	Network packet analysis. Applicable for the network layer.	www.wireshark.org	Packet Analysis
3	NMAP	Security scanner for the network layer. Applicable for the network layer.	www.nmap.org	Device discovery, service discovery, vulnerability discovery, and operating system discovery
4	SIFT	Applicable for all layers.	Digital-forensics.sans.org/community/downloads/#overview	File system analysis, memory image analysis, among others
5	XPLICO	Specially built for network analysis. Applicable for the network layer.	www.xplico.org	Network traffic analysis
6	Bulk Extractor	Scans files, directories, and disk images. Applicable for the top layer.	Tools.kali.org/forensics/Bulk-extractor	Extraction of disk images, files, and directories
7	Digital Forensics Framework (DFF)	Suitable for both top and bottom layers.	www.arxsys.fr	Investigation of hard drives and volatile memory
8	Foremost	Data recovery tools (file carving). Applicable for the top layer.	www.foremost.sourceforge.net	Data carving
9	The Open Computer Forensics Architecture	Suitable for both top and bottom layers.	www.ocfa.sourceforge.net	Digital forensic analysis
10	Guymager	Forensic Imager. Applicable for all three layers.	www.forensciswiki.org/wiki/Guymager	Disk imaging

(Mahmood et al., 2024) Further confirm that although several open-source tools are available for digital forensics in IoT, testing and validation remain incomplete, underscoring the need for rigorous evaluation in real-world settings.

The ninth framework is “IoTDots: A Digital Forensics Framework for Smart Environments by Babun et al. (2018). The researchers proposed to overcome the challenge of “smart application programming platforms not having digital forensics capabilities”. As a result, they developed a model targeted at *Samsung SmartThings*. The model incorporates policies that determine how a given IoT environment should operate. Consequently, the scholars created a two-tier model. The first tier is the IoTDots modifier, which collects forensic-relevant data and sends it to a database. The forensic data is obtained at compile-time and forwarded to the IoT Dots logs database at runtime. It is important to note that the smart app's source code will be modified during compile time. The logs are stored as forensic logs. In the second tier, an IoTDots analyzer extracts data from the IoTDots logs and analyzes them to obtain crucial forensic information. Analysis of the logs is based on a threat model that categorizes user activities as malicious, regular, or anomalous. Furthermore, machine learning has been incorporated into the framework to associate each device with its logs in the innovative environment. The authors' opinion is that the association would enable an investigator to identify all user activities and the state of a particular environment at the time of forensic analysis (Babun et al., 2018).

The proposed approach is novel because it includes machine learning techniques. It boasts 96% accuracy in identifying all the user's activity. However, the approach is platform-specific, and accuracy decreased as the number of users and compromised devices increased (Babun et al., 2018).

Building on this, Rudrakar et al. (2025) incorporated machine learning and AI-driven forensic readiness into IoT forensic models, thereby improving the accuracy and scalability of forensic analysis in complex real-world environments. (Mohammed et al., 2023) further proposed federated learning-based methods for enhanced detection of cyber-attacks in IoT, contributing to the evolving analysis phase in digital forensic investigations.

2.3 Designing a Model

How are IoT-based models for forensic investigation designed? This section will discuss the approaches that are used.

2.3.1 Modeling

"Systems modeling is the process of developing abstract models of a system, with each model presenting a different view of that system" (Sommerville, 2016). The different aspects of the system determine the requirements that can be used to develop a given model. Modeling usually takes a formal, informal, or a mixture of both approaches. "During the specifications engineering process, models are used to help obtain comprehensive system requirements, explain the system to engineers implementing the system during the design process, and record the layout and function of the system after implementation" (Sommerville, 2016). The formal approach uses mathematical functions to determine the requirements for a given model, whereas the informal approach uses graphical notation to determine the criteria for a given system.

2.3.2 Formal Methods

As stated earlier, official ways utilize mathematical concepts to determine system requirements. According to Pressman & Maxim (2020), "models developed using formal methods are described using a formal syntax and semantics that specify system function and behavior. The specification is mathematical in form; for example, predicate calculus can be used as the basis for a formal specification language. Usually, "set theory and logic notation are used to create a clear statement of requirements that can be analyzed to improve the correctness and consistency of a given model". It enjoys the advantage of developing software with an extremely low failure rate (Pressman & Maxim, 2020).

The ideas that are fundamental to formal methods include:

i. Data invariant, an accurate state in the entire implementation of the system that comprises aggregated data;
ii. The condition represents a system's externally detectable behavior, or the cache data the system accesses and modifies.
iii. Functioning is an activity that occurs in a system that reads or writes data to a particular condition. A function is associated with two states: a precondition and a post-condition.

2.3.3 Informal methods

Informal methods use graphical notation to determine the requirements for developing a given system. (Sommerville, 2016) states that graphical models are utilized in three methods. First, they can be used to stimulate and focus discussion about an existing or proposed system.

Secondly, they can be used to document the existing system, and, lastly, to facilitate a comprehensive system description that leads to its implementation. The notation uses Unified Modeling Language (UML) diagrams to elicit requirements.

A model can be represented from different perspectives. The perspectives can be external, interaction, structural, and behavioral (Sommerville, 2016). These perspectives give different views of the model in question. Diagrams such as use cases, activities, sequences, and classes are used to show the models. After generating the models, the system can be implemented, as the model requirements have been determined.

To implement a system successfully, every step of the requirements engineering process should be followed. "Requirement engineering deals with a broad spectrum of tasks and techniques that lead to an understanding of requirements" (Pressman & Maxim, 2020). According to Pressman & Maxim (2020), requirements engineering comprises seven distinct tasks: initiation, induction, explanation, conciliation, requirement, authentication, and administration. These tasks are typically tailored to the project, and some can run in parallel.

The requirements engineering process leads to the discovery of requirements. The specifications can be tangible or non-tangible for a software system. Tangible specifications address what a given system should be able to do, whereas non-tangible specifications concern the constraints on the functions of a given model as a whole. The requirements are documented after collection to facilitate the implementation of the model in question.

Despite the absence of standards in developing IoT, there are de facto standards for the digital forensic investigation process. A study conducted by Valjarevic et al. (2017) concluded that ISO/IEC 27043:2015 is an umbrella standard that covers all aspects of the digital forensic investigation process. Therefore, to accept digital evidence in court, it is paramount that the standard serves as a baseline for developing a digital forensic investigation model.

2.3.4 Internet of Things Ecosystem

The term IoT was coined by Kevin Ashton in 1999, during the fourth industrial revolution. Since that time, the field of IoT has undergone numerous changes, and as a result, we have seen myriad applications of the technology. The applications arise from the areas where the technology is used. IoT can be broadly categorized into two categories: consumer and enterprise (Leverege, 2018; Yeo et al., 2015). Consumer IoT includes applications such as smart home devices and wearables, which are typically sold to end users (Leverege, 2018). In

contrast, enterprise IoT has applications such as smart irrigation and fleet tracking that aim to "improve an organization's existing systems and processes and enable organizations to increase operational efficiency or unlock entirely new value" (Leverege, 2018).

According to the Internet Architecture Board (IAB) in 2015, bright objects in IoT use various communication mechanisms. The mechanisms are as follows:

i. Device-to-Device communication pattern

   In this design, the devices communicate directly through a wireless network. An example of this application is when a person uses a heart rate monitor paired with a smartwatch or a light bulb paired with its switch, among others. This model uses protocols such as Bluetooth Smart, Zigbee, and IP for its communication.

ii. Device-to-Cloud communication pattern

   This design uses a cloud service. The IoT 'things' are directly connected to the cloud. Given that 'things' have a connection to the cloud, a person can access the contents that are being relayed to the cloud from any location. This pattern uses cellular radio, a wireless local area network (WLAN), and wired Ethernet. In addition, protocols such as the Internet Protocol (IP) can be utilized.

iii. Device-to-Gateway communication pattern

   In this model, IoT devices communicate with a cloud service through an intermediary. The intermediary is a gateway device. For example, a smartphone that connects to IoT devices and the internet. It is always assumed that the gateway is always connected to the Internet. The model uses radio technologies such as BLE, IEEE 802.11, and IEEE 802.15.4, and protocols such as DTLS, IPv6, and CoAP, among others, for communication.

iv. Back-End data sharing pattern

   This model extends the device-to-cloud model, which often leads to silos. Moreover, users at times demand the ability to export and analyze sensor data alongside data from other sources, such as an external analytics platform. Therefore, authorized third parties need access to the data.

2.3.5 Digital Forensic Investigation

As earlier stated, digital forensics is a subset of forensic science. To unearth potential digital artefacts in a digital crime scene, a forensic investigator would have to use a digital forensic process. The processes involved are: collection, preservation, analysis, reporting, identification, validation, and interpretation. All the processes deal with digital evidence.

According to Donovan (2012), the terms "digital forensic investigations, digital investigations, forensic investigations, and forensic examination have all been used to describe investigations in which a digital device forms part of the incident. Many authors have sought to define digital forensic investigation (DFI). Therefore, we will examine their various perceptions of DFI and what it entails.

(Ieong, 2006) defined DFI "as a process to determine and relate extracted information and digital evidence to establish factual information for judicial review". He added that the principles of reliability, reconnaissance, and relevance guide DFI. For example, if an investigator focuses on the relevant evidence, he would be in a position to have control over the price and period (Ieong, 2006). Moreover, the sentiment of controlling costs was echoed by Tan (2001).

"A DFI is concerned with the retrieval, acquisition, analysis, and examination of potential digital evidence (PDE) in such a way that the evidence will be accepted in a court of law"(Kebande, 2018). This definition encompasses the processes involved in the investigation.

DFI covers two things from the definitions. One is the acquisition of reliable evidence that explains how a digital crime was perpetrated. Secondly, it is a demonstration of information of probative value before a court, in a manner acceptable to the court. Therefore, a forensic investigator must use an investigative process that leads to the discovery of evidence that withstands scrutiny by a court.

2.3.6 Challenges Facing Internet of Things Forensics

Given the dynamic nature of IoT, various challenges have emerged. These challenges range from standards governing IoT to the hardware that runs it.

i. Security and Privacy

Various authors have noted that security and privacy are primary concerns for IoT devices. Hewlett-Packard (HP) in 2014 investigated the widely used IoT 'things. They discovered

vulnerabilities that existed in over 70% of ‘things. They include weak password mechanisms and weak encoding, among others. (Koley & Ghosal, 2016) acknowledged that security and privacy preservation are a problem in IoT, both for the hardware and software that support it. Software and hardware vulnerabilities are often left unpatched (Kaul & Goudar, 2017; Koley & Ghosal, 2016; Nzabahimana, 2018), thereby increasing the attack surface of the IoT ecosystem (Nzabahimana, 2018). As a result, hacking incidents occur because the prevention mechanisms are not adequate across all devices (R. Gupta & Gupta, 2016). Recent studies reinforce this concern, highlighting the need for advanced forensic tools and AI-driven threat detection to address emerging vulnerabilities (Ahmed et al., 2024; Ahmed et al., 2025; Garcia Avila et al., 2024).

ii. Standards

According to (Andress & Leary, 2015), “an information security standard describes the implementation and management of information security controls. A standard provides an information security control to meet required specifications, including those for meeting specific industries or regulatory compliance objectives”. Furthermore, standards can be de facto or de jure (Whitman & Mattord, 2021). In the IoT environment, there are no universal standards for how IoT devices should operate (Al-Shargabi & Sabri, 2018; Singh & Singh, 2016; Yadav et al., 2018). This can cause interoperability issues due to the various protocols and the number of devices involved in the IoT environment. Thus, there is a need to develop common standards to mitigate the problem. In addition, current studies emphasize the importance of harmonizing standards to enable effective forensic investigations and secure communication protocols (A. A. Ahmed et al., 2024; Garcia Avila et al., 2024).

iii. Power Consumption

IoT devices require an uninterrupted power supply to function effectively. (Tongay, 2016) Indicated that sensor networks need a reliable power supply to function optimally. However, K. Gupta & Shukla (2016) stated that one of the main challenges IoT faces is reducing power consumption. This is due to the number of devices that need to be connected for the technology to work. Moreover, Kaul & Goudar (2017) postulated that there is a need to prolong the battery life of IoT devices and to harvest energy. This will help mitigate the challenge of high energy consumption. Emerging research suggests that forensic readiness must consider energy-efficient logging and evidence preservation mechanisms (Ahmed et al., 2025)

iv. Heterogeneity

The IoT environment is heterogeneous. This is because different devices are interconnected so that technology can work. The interconnections bring about the issue of interoperability. According to Reddy et al. (2018), the heterogeneous nature of IoT protocols is a significant hurdle. As a result, the standardized implementation of protection frameworks for IoT devices is complex. (Anthi et al., 2018).(Garcia Avila et al., 2024) emphasizes that heterogeneity complicates forensic data collection and necessitates adaptive frameworks for evidence analysis.

v. Computational Power

IoT devices vary in size, but most are extremely small. The small size makes it impossible to add more computational power, resulting in low computational capability in the devices (Anthi et al., 2018; Koley & Ghosal, 2016). Therefore, the present complex security algorithms are not suitable for IoT devices. (K. Gupta & Shukla, 2016) added that there are no lightweight cryptosystems or security protocols, leading to insecure web interfaces and a lack of transport encryption (Anthi et al., 2018). Recent studies propose lightweight forensic and security protocols tailored for constrained devices (Ahmed et al., 2024; Ahmed et al., 2025).

vi. Big Data

According to IBM, 90 percent of data generated by devices such as PDAs, smartphones, and others is not examined. In addition, the data loses value immediately, within milliseconds. The data is around 60 percent. In the context of IoT, many devices act as enablers of technology. This leads to the generation of large amounts of data (Kaul & Goudar, 2017), raising the question of how to handle large volumes of data (Tongay, 2016). Furthermore, Tongay (2016) posited that sensor data quality is problematic.

vii. Authentication and Authorization

Authentication and authorization are other significant challenges in the IoT ecosystem. Considering that the devices are small and may lack user interfaces, Singh & Singh (2016) theorize that they lack “measures to confirm identity of entities requesting access to any data”. This is worsened by the fact that the IoT environment is very complex and heterogeneous. Therefore, it becomes challenging to identify rogue nodes in an IoT ecosystem if they appear. Additionally, there is a lack of authentication and authorization mechanisms.

viii. Identity and Location

Identifying and locating devices in an IoT ecosystem poses another security challenge. Given that the current IPv4 address space is insufficient (R. Gupta & Gupta, 2016), connecting a large number of network devices becomes problematic.

2.3.7 Relevance of IoT Forensic Challenges to Event Reconstruction in Smart Homes

The challenges inherent in IoT forensics are directly relevant to event reconstruction in smart home environments, as they determine the reliability, accuracy, and admissibility of forensic evidence. Security and privacy vulnerabilities remain a critical concern. Weak authentication, poor encryption, and unpatched firmware create opportunities for attackers to delete logs, fabricate false events, or conceal their activities. Such vulnerabilities compromise the reliability of event reconstruction (Koley & Ghosal, 2016; Kaul & Goudar, 2017; Nzabahimana, 2018). More recent studies confirm that smart home devices continue to suffer from exploitable weaknesses, making forensic reliability dependent on robust integrity verification mechanisms (Kaushik & Bhardwaj, 2023; Ahmed et al., 2024). This research, therefore, integrates Hash-based Message Authentication Codes (HMACs), encrypted transmission, and secure storage to protect forensic evidence.

The absence of universal IoT standards further complicates event correlation. Devices from different manufacturers often use proprietary protocols and heterogeneous data formats, leading to inconsistent timestamps and incompatible event sequences (Al-Shargabi & Sabri, 2018; Singh & Singh, 2016). Recent work emphasizes that interoperability remains a barrier to forensic readiness in smart homes (Kaushik & Bhardwaj, 2023; Ahmed et al., 2024). To address this, the proposed model adopts a standardized log format, unique device identifiers, correlation IDs, and ISO/IEC 27043:2015-compliant processes to ensure consistency.

Power consumption constraints also affect evidence volatility. IoT devices with limited energy resources may unexpectedly shut down, losing volatile logs or disabling continuous monitoring (Tongay, 2016; Gupta & Shukla, 2016). Contemporary studies highlight that forensic readiness requires proactive solutions to mitigate data loss during outages (Rizal et al., 2025). This research employs local caching on Raspberry Pi, battery backup systems, and efficient logging mechanisms to minimize power-related evidence gaps.

The heterogeneity of IoT environments introduces complexity in timeline reconstruction. Devices differ in their synchronization protocols, timestamp precision, and data structures, making it difficult to establish unified timelines (Reddy et al., 2018; Anthi et al., 2018). Recent frameworks propose centralized timestamp normalization and correlation algorithms to overcome these inconsistencies (Breitinger et al., 2025). This study adopts similar strategies to generate coherent timelines across diverse devices.

Limited computational power in IoT devices restricts real-time forensic processing. Resource-constrained sensors cannot perform complex analysis, necessitating lightweight protocols and efficient algorithms (Anthi et al., 2018; Koley & Ghosal, 2016). Current research confirms that forensic processing should be offloaded to intermediary hubs with greater computational capacity (Ahmed et al., 2024). Accordingly, this model performs forensic processing on a Raspberry Pi hub, while lightweight logging is maintained on ESP32 devices.

The volume of data generated by IoT devices presents challenges in evidence management. Massive datasets complicate storage, retrieval, and reconstruction (Kaul & Goudar, 2017). Recent studies emphasize selective logging and efficient indexing as critical for forensic scalability (Kaushik & Bhardwaj, 2023). This research implements severity-based logging, database indexing, and advanced filtering in the forensic dashboard.

Weaknesses in authentication and authorization undermine event attribution. Without robust mechanisms, investigators struggle to determine who initiated specific actions or to distinguish legitimate from malicious activity (Singh & Singh, 2016). Updated approaches recommend mandatory authentication, role-based access control, and comprehensive user action logging (Ahmed et al., 2024). These measures are integrated into the proposed model.

Finally, identity and location issues complicate device attribution. Ambiguity in device origins and physical locations hinders correlation of events to specific smart home zones (Gupta & Gupta, 2016). Recent work stresses the importance of metadata-driven device identification (Kaushik & Bhardwaj, 2023). This study resolves such issues through unique device IDs, IP address logging, and contextual metadata storage.

Collectively, these challenges demonstrate that traditional forensic approaches are insufficient for smart homes. Event reconstruction emerges as essential, enabling causality analysis,

preservation of volatile evidence, unified representation of heterogeneous environments, legally admissible timelines, and tamper detection through integrity verification. Recent scholarship affirms that event reconstruction is the cornerstone of forensic readiness in smart homes, bridging the gap between conceptual frameworks and practical implementation (Ahmed et al., 2024; Breitinger et al., 2025; Rizal et al., 2025).

Event reconstruction, as implemented in this research, is the comprehensive solution that addresses all these challenges simultaneously. It provides a forensically sound method for capturing, preserving, correlating, and analyzing events in smart home environments despite the numerous technical and architectural limitations of IoT devices.

Without addressing these challenges through systematic event reconstruction, digital forensic investigations in smart homes would remain unreliable, incomplete, and potentially inadmissible in legal proceedings. This research bridges the gap between the identified challenges and a practical, implementable solution.

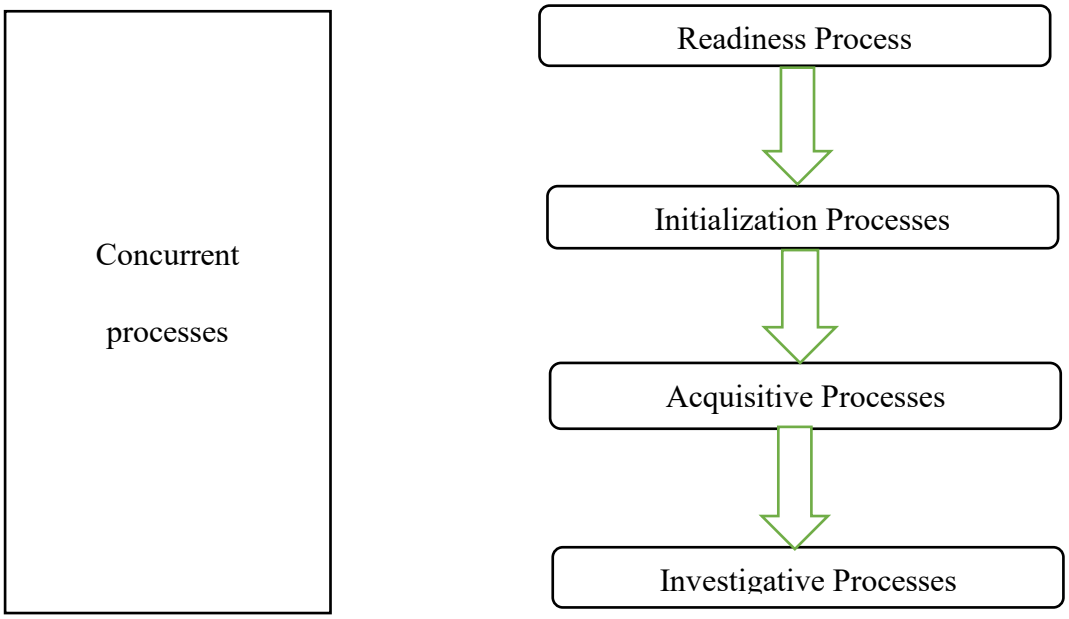
2.4 Requirements for Designing a Model

2.4.1 Overview of ISO/IEC 27043:2015

This standard is titled: Information technology — Security techniques — Incident investigation principles and processes. According to (ISO/IEC 27043:2015, 2015), “it presents an idealized model for digital forensic investigation process (DFIP)”, covering all aspects of digital investigation, including forensic preparedness. It covers all kinds of investigations, that is, civil, criminal, and corporate; thus, it can be used for both live and dead forensics. The standard is a high-level guide; hence, it provides a broad overview of the whole event exploration exercise. The standard also sets out fundamental propositions that help ensure instruments and approaches are correctly chosen and shown to suit the motive when needed.

Figure 1

Classes of digital investigation process (ISOIEC 27043:2015)



i. Concurrent processes

Concurrent processes were initially developed by Valjarevic & Venter (2015). It entails examination procedures that run concurrently with other operations (Valjarevic & Venter, 2015). It has processes, such as obtaining authorization, that help ensure evidence can be legally acquired and hence admissible in a court of law. Furthermore, it has documentation outlining the steps to be taken during the investigation. Additionally, it has information flow management, protection of PDE, protection of how PDE is handled, and interaction with physical investigation. In the authors' opinion, simultaneous procedures aim to achieve the desired outcome in an inquiry.

ii. Readiness process

"Forensic readiness is defined as the ability of an organization to maximize its potential to use digital evidence while minimizing the costs of an investigation" (Rowlingson, 2004). It is worth noting that activities during this phase are always proactive. Proactive activities help an organization put the necessary measures in place before a security incident occurs. The measures entail planning for incidents and handling potential digital evidence when incidents

are detected (ISO/IEC 27043:2015, 2015). (Tan, 2001) identified that logging is the central element of forensic readiness. Therefore, an organization should put mechanisms in place to help it be forensically ready.

iii. Initialization process

These are activities that begin after an event has been detected. According to ISO/IEC 27043, it includes activities such as observing an incident, responding to the initial incident, organizing, and preparing for a real digital forensic investigation. This step requires the forensic investigator to familiarize themselves with the apparatus and techniques that will aid in conducting the investigation. Typically, this procedure is reactive in an investigation.

iv. Acquisitive process

This deals with the physical acquisition of evidence. All the processes in this group contribute towards acquiring information of probative value. (ISO/IEC 27043:2015, 2015) incorporates the procedures of recognition, assemblage, gaining, storage, and protection of the information of probative value.

v. Investigative process

According to ISO/IEC 27043, the investigative process includes activities aimed at the incident that led to the commencement of a digital forensic investigation. It involves analysis of evidence to draw reasonable conclusions. This phase helps answer the forensic questions of who, what, when, how, where, and why a given security incident occurred.

2.5 Research Gap

After reviewing the literature, it is evident that a research gap exists. Table 2 shows the weaknesses of the various models discussed. Among the models reviewed and those not mentioned, none included event reconstruction. Thus, the author seeks to fill this gap by developing a model that enables event reconstruction.

Table 2

Research Gap

Model	Weaknesses

1-2-3 zone and next best thing triage by (Oriwoh et al., 2013)	The proposed framework may not be applicable in real-world scenarios due to the diverse and ever-evolving nature of IoT devices. The proposed structured approach to the framework may underestimate the complexities of actual investigations. The framework does not consider the legal implications of data collection in IoT forensics. Various issues, such as data ownership, user privacy, and the admissibility of evidence in court, have not been explored despite their critical nature. The framework does not provide for data integrity during the collection and analysis of evidence, thereby allowing evidence to be altered. Also, the model does not offer a robust solution. This is critical because IoT devices lack adequate security measures. The framework does not provide mitigating strategies for potential data loss arising from the limited storage capacity of IoT devices.
Internet of things Digital forensic investigation model: Top-down forensic approach methodology by Perumal et al. (2015).	The model may be complex for practical use, as it requires a comprehensive understanding of various IoT systems and their interactions, which can also be challenging to implement in real-life investigations. The model does not adapt well to the rapid evolution of IoT technologies and devices, leading to potential obsolescence as new devices and protocols emerge. The model emphasizes a structured approach, which may overlook the dynamic environments typical of IoT settings, where devices interact autonomously and unpredictably.
Forensic-aware IoT by (Zawoad & Hasan, 2015)	The model lacks adequate tools for forensic analysis because many IoT devices use proprietary software and closed-source structures, making evidence extraction challenging.

	The model does not sufficiently address the need for real-time data collection from IoT devices, which is critical in fast-moving investigations. The model struggles with the diversity of communication protocols used by different IoT devices, complicating evidence collection and analysis.
A generic digital forensic investigation framework for the Internet of Things by (V. R. Kebande & Ray, 2016)	The generic nature of this framework may not account for the specific needs and characteristics of different IoT environments, leading to ineffective forensic investigations. The framework does not address the integrity of evidence during collection and analysis, which is crucial for legal admissibility. The framework may not effectively manage the large volumes of data generated by IoT devices, potentially leading to evidence loss or oversight.
Application-Specific Digital Forensics Investigative Model in Internet of Things (IoT) by Zia et al., 2017)	While application-specific models can be beneficial, they may limit the applicability of the findings to broader contexts or other types of IoT applications. The model's effectiveness may depend on specific technologies that could become outdated or unsupported over time. It may not encompass all aspects of digital forensics needed for a thorough investigation across various IoT ecosystems.
A forensic investigation framework for a smart home environment by (Goudbeek et al., 2018).	The reliance on specific case studies may limit their generalizability and effectiveness in varied real-world scenarios. It may face interoperability issues among different smart home devices, complicating evidence collection. The reliance on specific case studies may limit their generalizability and effectiveness in varied real-world scenarios.

IOT Forensic: A digital investigation framework for IoT systems by (Sathwara et al., 2019)	The framework may not adequately address the limitations of existing forensic tools that struggle with diverse IoT architectures and data types. There is often insufficient focus on strategies for real-time data collection from interconnected devices during an active investigation. The framework might not adequately address privacy issues arising from the collection of sensitive information from users' devices during investigations.
Internet of Things digital forensic investigation using open-source tools by Al-Sadi et al. (2018).	While open-source tools can be beneficial, they may lack robustness or support compared to commercial solutions, potentially limiting their effectiveness in critical investigations. Open-source tools might not be compatible with all types of IoT devices or their proprietary systems, hindering comprehensive forensic analysis. There are risks to data integrity when using open-source solutions, which could affect the admissibility of evidence in legal contexts. Limited scalability in large-scale IoT environments can hinder its effectiveness in extensive investigations.
IoTDots: A Digital Forensics Framework for Smart Environments by Babun et al., 2018)	The framework may struggle to correlate evidence across multiple devices and platforms due to the inherent complexity of intelligent environments. As innovative environments grow in size and complexity, the framework might face challenges scaling effectively to handle increased data volumes and device interactions.

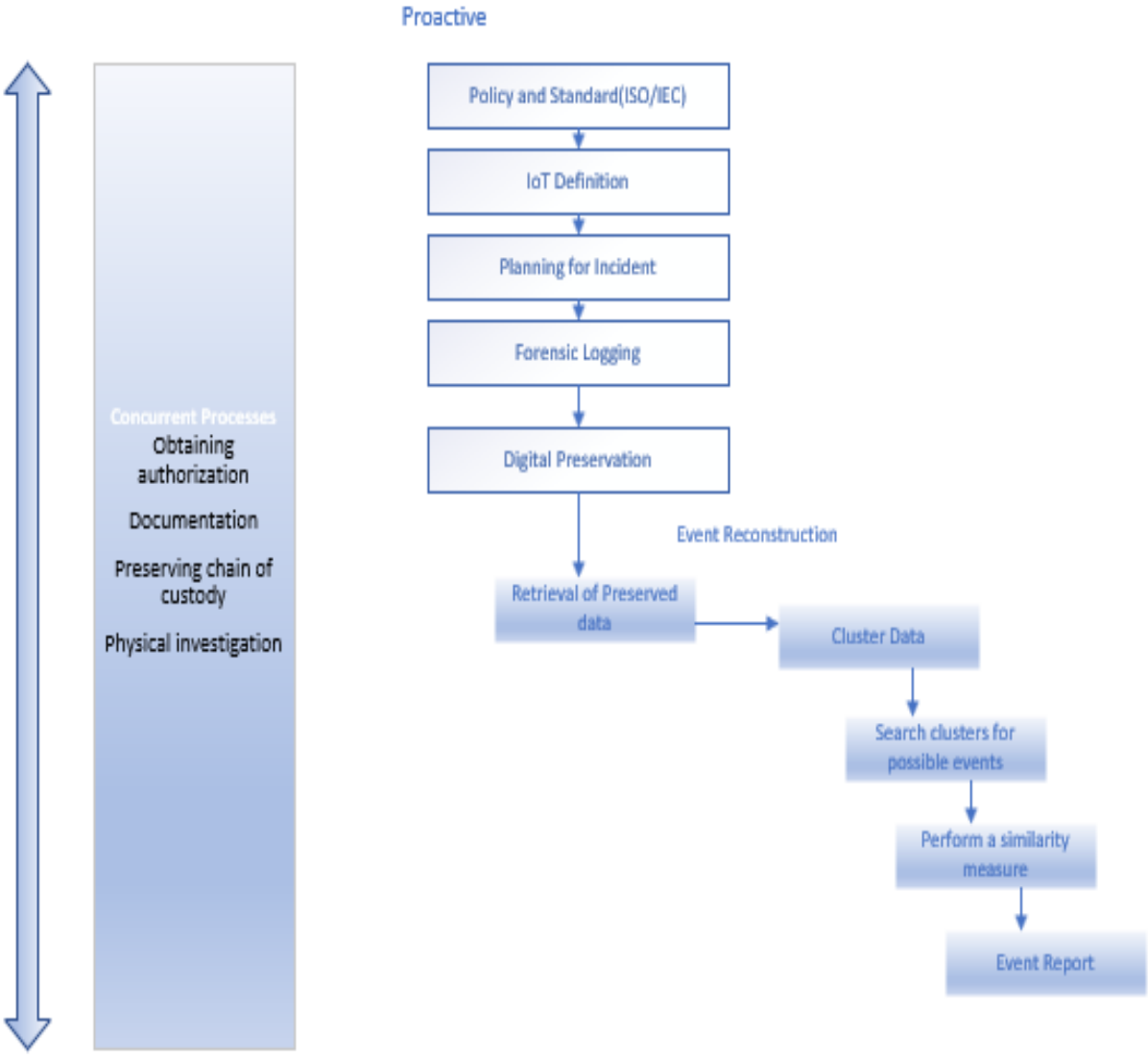
There may be insufficient empirical testing in real-world scenarios to validate its effectiveness across diverse innovative environments.

Lacks user-friendliness, which could hinder adoption by forensic practitioners unfamiliar with complex frameworks.

2.6 Conceptual framework

Figure 2

Conceptual Model



Source: Author (2025)

In the figure above, concurrent processes are activities that occur throughout the investigative process. It will begin with obtaining authorization from the relevant sources, and everything that happens throughout the investigative process will be documented. In addition, any person who handles digital evidence must acknowledge that they handled it before the physical examination of devices.

The next part is the proactive process. This section covers what happens before a security incident occurs. First, it begins with a standard or policy that will define how the smart home environment will operate. Thereafter, a forensic investigator prepares for possible scenarios that a security incident may pose. Moreover, forensic logging will be performed on data generated by the "things" in the smart home. Thereafter, the log information is digitally preserved.

The last phase is event reconstruction. In event reconstruction, the preserved digital evidence will be retrieved from the storage. Hereafter, data clustering will be performed. The clusters will be based on their occurrence and similarity (Kebande & Venter, 2015). Moreover, the similarity measure will be based on the Manhattan distance, as depicted in equation 1 below.

2.7 Summary

This chapter began by describing existing models; it then presented the challenges in the IoT landscape, the various ways to design a model, the research gap, and, lastly, the conceptual framework.

CHAPTER THREE

RESEARCH DESIGN AND METHODOLOGY

3.1 Introduction

This chapter presents the research methodology and discusses the following areas: research design, data collection, model implementation, and model evaluation.

3.2 Research Philosophy

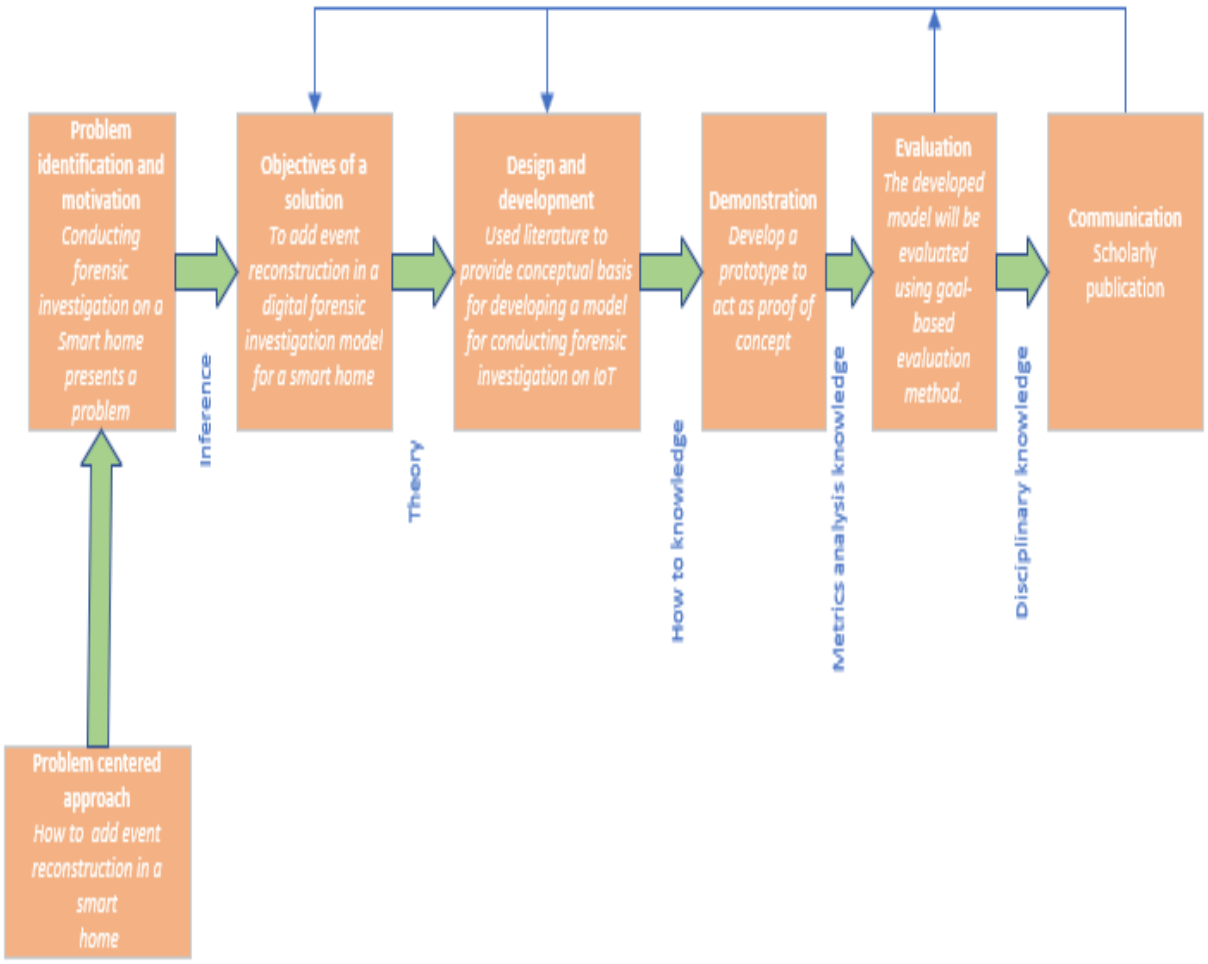
A researcher's research philosophy is an individual's conceptualization of truth, reality, and knowledge, reflected in the decisions they make regarding the design, data collection, and data analysis of a research study. These choices are based on philosophical principles and values (Ryan, 2018). From a philosophical perspective, this study will adopt positivism. According to Park et al. (2020), positivism emphasizes identifying explanatory associations or causal relationships through quantitative approaches, where empirically based findings from large sample sizes are favored—in this regard, generalizable inferences, replication of findings, and controlled experimentation have been principles guiding positivist science.

3.3 Research Design

Research design refers to the overall strategy for integrating the study's components coherently and logically, ensuring that the research problem is effectively addressed (Kothari & Gaurav, 2019). This study employed a multi-methodological design that integrated three complementary approaches: a systematic literature review (SLR) to investigate existing models, design science research methodology (DSRM) to design the model, and rapid prototyping to implement and test the model. The integration of these approaches was guided by the Design Science Research (DSR) framework proposed by Peffers et al. (2006), which consists of six iterative steps: problem identification and motivation; solution objectives; design and development; demonstration; evaluation; and communication, as depicted in Figure 3.

Figure 3

Design science research process



The diagram depicted in Figure 3 shows the overall methodology used to conduct the research. The DSR approach has different starting points. For this study, the approach began with identifying a problem. After which, we determined the objectives of the solution that solved the identified problem. The third phase focused on determining the requirements for designing the model, followed by the development of a prototype that solved the problem. The model was evaluated, and the solution was finally published.

The methodology for achieving each objective is detailed in the sections that follow.

3.3.1 Systematic Literature Review

A systematic literature review was conducted to achieve the first objective: to investigate existing models of digital forensic investigation for IoT devices. The review established the current state of knowledge in IoT forensics and identified gaps that this research would address.

Searches were conducted across databases, including IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, and Google Scholar, using terms such as “IoT forensics,” “digital forensic investigation model,” “smart home forensics,” “event reconstruction,” and “forensic readiness IoT.” The review focused on publications from 2013 to 2024 to capture contemporary developments. After screening, nine key models were selected for detailed analysis. Each was examined with respect to architecture, investigative phases, technical requirements, strengths and limitations, and applicability to smart home environments. The models were then evaluated using ten assessment criteria, including forensic readiness, chain of custody, accuracy, scalability, auditability, completeness, reliability, ease of integration, data security, and timeliness. The review revealed that existing models lacked event reconstruction capabilities, had limited practical implementation, and were predominantly theoretical, thereby justifying the need for this research.

3.3.2 Design Science

Design Science Research was employed to achieve the second objective: to design a digital forensic investigation model for a smart home with event reconstruction. DSRM was appropriate because it emphasizes the creation and evaluation of innovative artifacts that address practical problems while contributing to theoretical knowledge. The design process began with problem identification, during which the absence of event reconstruction in existing IoT forensic models was identified as a critical gap. Objectives of the solution were then defined, including functional requirements such as user registration, device management, log capture, event correlation, and timeline generation, as well as non-functional requirements such as security, reliability, and accuracy. Technical requirements were specified, encompassing hardware components, software frameworks, and integration protocols.

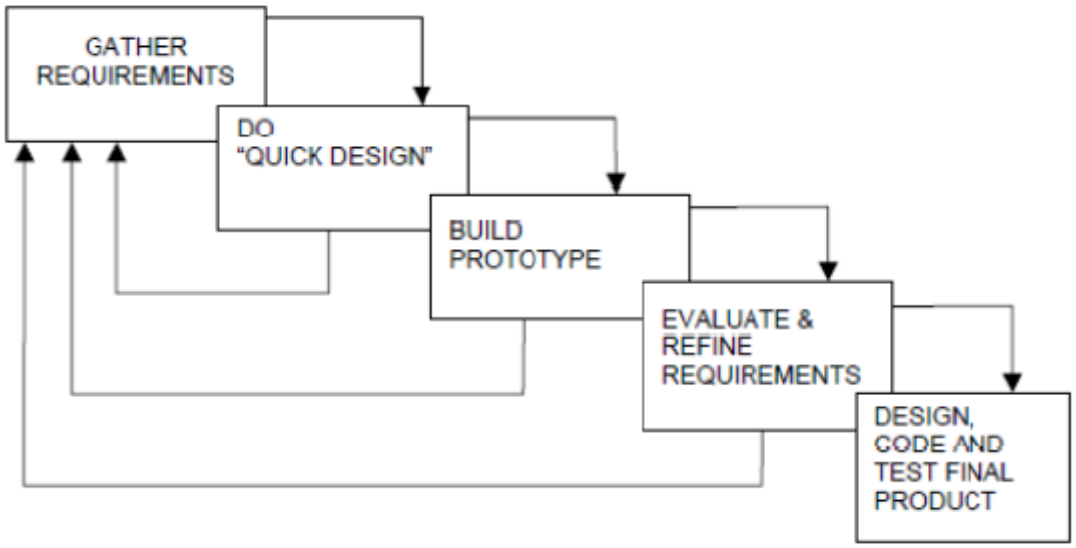
The development phase involved requirements engineering through user stories and use case analysis, architectural design using a layered approach, and security design incorporating HMAC, encrypted transmission, and role-based access control. Event reconstruction was designed to operate through log correlation, anomaly detection, timeline generation, and integrity verification. The demonstration phase was realized through rapid prototyping, while evaluation was conducted using goal-based experiments and expert validation. Finally, results will be communicated through this thesis and planned scholarly publications.

3.3.3 Prototype implementation

Rapid prototyping was used to achieve the third objective: implementing the model. This approach allowed iterative development, early testing, and continuous refinement. The implementation process followed the NASA rapid prototyping model, beginning with requirements identification, quick design, and prototype building. Hardware implementation involved configuring a Raspberry Pi as the central forensic hub, integrating ESP32 microcontrollers, connecting DHT11 sensors, and installing relay modules for smart home control. Software implementation included developing a web application with Flask, implementing authentication and registration modules, creating a real-time log-monitoring dashboard, building event-correlation and timeline-generation engines, and integrating anomaly-detection algorithms. The prototype was evaluated by potential users, refined based on feedback, and ultimately engineered into a functional product capable of real-time log capture, event reconstruction, anomaly detection, and forensic evidence integrity verification. Figure 4 depicts the steps for implementing the model.

Figure 4

Rapid prototyping (NASA, 2004)



3.3.4 Evaluation of the Model

The model evaluation was undertaken to address the fourth objective, which focused on assessing its reliability and effectiveness. Although limited research has been conducted on the parameters suitable for evaluating forensic models (Flandrin et al., 2014), this study employed a combination of goal-based evaluation and expert validation. Goal-based evaluation was used

to determine whether the model achieved its intended functional outcomes, and expert validation was conducted through purposive sampling of a qualified forensic specialist. The evaluation framework incorporated metrics such as accuracy, auditability, completeness, and reliability, as outlined by Ayers (2009). Each of these metrics was measured through controlled experiments and expert review: accuracy was determined by comparing expected outcomes with actual results; auditability was assessed by verifying the chain of custody and detecting potential tampering; completeness was evaluated by examining the recovery of relevant artifacts; and reliability was tested by analyzing the consistency of results under varying load conditions. A comparative analysis against nine existing models further highlighted the strengths of the proposed approach, particularly its practical implementation and its unique capability to support event reconstruction in smart home forensic investigations.

3.4 Data Collection Methods

Data collection involved both secondary and primary sources. Secondary data were obtained from the published literature, while primary data were collected in the smart home prototype environment. Tools such as MQTT Explorer, MongoDB Compass, and custom Python scripts were used to capture logs, monitor networks, and validate data integrity. Data was stored securely in MongoDB Atlas and locally on the Raspberry Pi, with encryption, hashing, and access controls to ensure security.

3.5 Population and Sampling

The study population consisted of IoT devices in a smart home environment and forensic experts for evaluation. Purposive sampling was used to select 9 forensic models for review and 1 expert for validation, while convenience sampling was used for the IoT devices. Although the use of a single expert evaluator was acknowledged as a limitation, it was deemed sufficient for proof-of-concept validation.

3.6 Ethical issues

To conduct the research, formal approvals were first obtained from the relevant institutional bodies. A letter of authorization was secured from the Institute of Postgraduate Studies (IPGS) at Kabarak University, followed by ethical clearance from the Kabarak University Research Ethics Committee (KUREC), as required by university policy. In addition, a research permit was granted by the National Commission for Science, Technology, and Innovation (NACOSTI), thereby providing national-level approval to carry out the study. All data sources

used in the research were duly acknowledged, and when consent was required to handle specific data, the researcher sought and obtained permission from the respective data providers.

3.7 Summary

In summary, this chapter has presented the research methodology employed in the study. Guided by positivism, the research integrated systematic literature review, design science research methodology, and rapid prototyping to investigate, design, implement, and evaluate the proposed model. Data collection and evaluation methods ensured rigor, while ethical considerations safeguarded integrity. The next chapter presents the results of applying this methodology to develop and assess the event reconstruction model for smart home forensic investigation.

CHAPTER FOUR

DATA ANALYSIS, PRESENTATION, AND DISCUSSION

4.1 Introduction

This chapter discusses the study's findings in relation to the objectives. It addresses section 1.5 of this study. Consequently, it includes an analysis of existing models in IoT device forensics, as well as the design, implementation, and evaluation of the event reconstruction model for a smart home.

4.2 Models in Digital Forensic Investigation for IoT devices

This section presents the results of objective one of the study. Objective one required the identification of existing digital forensic models for Internet of Things devices. Accordingly, a systematic literature review was used. The results are discussed as follows.

4.2.1 Weaknesses of existing models

In recent years, significant work has been done to conduct digital forensic investigations in the Internet of Things. Many academicians have developed several models in parallel to find a lasting solution to the same problem. To ensure the models created are “fit for purpose”, several metrics have been used to evaluate them. Metrics such as forensic readiness, accuracy, auditability, completeness, reliability, repeatability, timeliness, scalability, data security, and chain of custody were used as measures. Despite researchers' cultivation of a wide range of knowledge, they have also fallen short in meaningful ways. Table 2 below shows the key assessment criteria for the models discussed in Chapter Two.

Table 2

Assessment Criteria Key

Assessment Criteria	Definition	Traceability
i. Forensic Readiness	Preparation to handle investigations proactively, including policies, tools, and training.	A

ii.	Chain of custody	Documenting and maintaining the integrity of evidence handling.	B
iii.	Accuracy	Precision and correctness while handling evidence.	C
iv.	Scalability	Ability to handle large volumes and diverse IoT devices.	D
v.	Auditability	Ability to review and verify forensics processes and results.	E
vi.	Completeness	Coverage of all necessary forensic phases and evidence types.	F
vii.	Reliability	Consistency and dependability of forensic results.	G
viii.	Ease of Integration	The model's ability to work with other systems.	H
ix.	Data Security	Protection of evidence from unauthorized access or tampering.	I
x.	Timeliness	Speed of investigation and evidence processing.	J

Table 3

Existing Models Assessment

Model	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**
1-2-3 zone and next best thing triage by (Oriwoh et al., 2013)	×	×	✓	✓	×	×	×	×	×	✓
Internet of things Digital forensic investigation model: Top-down forensic approach methodology by Perumal et al. (2015).	×	✓	✓	×	✓	×	×	×	×	×
Forensic-aware IoT by (Zawoad & Hasan, 2015)	✓	✓	✓	×	✓	×	✓	✓		×
A generic digital forensic investigation framework for the Internet of Things by V. R. Kebande & Ray (2016)	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Application-Specific Digital Forensics Investigative Model in Internet of Things (IoT) by Zia et al., 2017)	×	×	✓	×	×	✓	×	×	×	×
A forensic investigation framework for a smart home environment by (Goudbeek et al., 2018).	×	✓	✓	×	✓	✓	×	×	×	×

IOT Forensic: A digital investigation framework for IoT systems by (Sathwara et al., 2019)	×	×	✓	✓	×	×	×	×	×	✓
Internet of Things digital forensic investigation using open-source tools by Al-Sadi et al. (2018).	×	×	✓	×	×	×	×	×	×	×
IoTDots: A Digital Forensics Framework for Smart Environments by Babun et al., 2018)	✓	×	✓	✓	✓	×	×	×	×	×

As Table 3 illustrates, there are significant fundamental flaws in current IoT forensic models that hinder efficient digital investigations. While some, such as DFIF-IoT, IoTDots, Zawoad, and Hasan's Forensic-aware IoT, emphasize forensic preparedness, most models do not take a holistic approach to addressing the rapid changes and variety of IoT devices. Inadequate implementation of chain-of-custody procedures compromises the validity of the evidence. Furthermore, many frameworks are unable to guarantee the veracity of evidence due to insufficient data integrity controls and difficulties with proprietary and heterogeneous devices. In addition to these problems, many models lack cryptographic mechanisms and safeguards, leading to unreliable IoT forensics.

Scalability and auditability are key barriers limiting IoT forensic models. Current frameworks are often overwhelmed by device data volume and lack essential traceability and verifiability features. Many models neglect crucial steps such as real-time data collection, legal compliance, and privacy, thereby weakening their reliability and adaptability in changing environments.

Integration, data security, and timeliness issues all reduce the usefulness of IoT forensic models. Interoperability difficulties resulting from the use of multiple platforms and protocols hinder cross-system collaboration. Inadequate evidence protection increases the risk of tampering and unauthorized access, exacerbated by IoT device vulnerabilities and the limitations of open-source tools. A lack of real-time evidence collection delays investigations, particularly in dynamic contexts. These repeated failures underscore the urgent need for flexible, secure, and compliant IoT forensic systems.

4.2.2 Design Recommendations to Counter Identified Weaknesses

To obtain the recommendations, the weaknesses of existing models were analyzed. Accordingly, Table 4 below presents the recommendations.

Table 4

Design Recommendations

Assessment Criterion	Design Recommendations
Forensic Readiness (A)	Design IoT systems that embed forensic readiness: proactive policies, secure logging, timely evidence capture, and training tools to prepare for investigations early. This includes integrating

	evidence logging and secure timestamps in IoT devices.
Chain of Custody (B)	All evidence handling should be documented using a physical, paper-based chain of custody log. Every transfer, inspection, or change of possession must be documented in a bound ledger with numbered pages to avoid removal or manipulation. Each record must include the date, time, handler name, role, signature, and an explanation of what the person was doing with the evidence.
Accuracy (C)	Integrate AI and machine learning to automate anomaly detection, reduce human error, and correlate evidence across heterogeneous devices, thereby improving the accuracy and completeness of forensic data. Use cryptographic measures to validate data authenticity.
Scalability (D)	Adopt hybrid fog-cloud forensic architectures that distribute processing closer to data sources while leveraging the cloud for extensive storage and analytics, enabling horizontal scalability and low-latency evidence acquisition. Implement big data management frameworks.
Auditability (E)	Implement standardized logging formats and protocols to ensure comprehensive, tamper-proof audit trails that provide a clear record of events. This allows independent parties to review and verify forensic procedures and results fully.
Completeness (F)	Cover all forensic phases end-to-end: readiness, identification, acquisition (including real-time), examination, analysis, and presentation. Implement

	adaptive workflows that are sensitive to IoT's dynamic, distributed environment.
Reliability (G)	To enhance the robustness of evidence collection and analysis in IoT environments, incorporate redundant systems, such as parallel data stores, to prevent data loss. Implement comprehensive error handling protocols by detecting and logging exceptions, and apply fault-tolerance mechanisms, such as system rollback or failover. Conduct continuous monitoring with real-time alerts and perform regular validation of system performance by scheduled audits and performance tests.
Ease of Integration (H)	To implement open interfaces and protocol abstraction layers, identify target interoperability standards, select compatible APIs, and design modular architecture components. This enables seamless integration with forensic and security tools, IoT management platforms, and cross-jurisdictional systems.
Data Security (I)	The model should implement multilayered security controls: encryption, access controls, secure transmission, and privacy-preserving techniques (data minimization, anonymization) throughout the forensic process to protect evidence confidentiality and integrity.
Timeliness (J)	The model should enable near-real-time or live evidence acquisition and automated analysis using fog computing and AI, facilitating swift investigation responses vital in ephemeral, fast-changing IoT environments.

By addressing these recommendations, the model could be developed.

4.3 Design of a digital forensic investigation model for a smart home with event reconstruction

This section presents the results for objective two of the study. The objective was to design a digital forensic investigation model for a smart home with event reconstruction. This was accomplished using the design science methodology. This section covers the design of the model's functional, non-functional, and technical requirements.

4.3.1 Functional Requirements for the Model

4.3.1.1 User Roles

The implementation in this research focused on three user groups: homeowners, forensic investigators, and threat actors. To capture the functional requirements for each user, user stories were developed as shown in Table 5. Furthermore, it is essential to note that the user stories and requirements were intentionally condensed to their core components, while ensuring the proof of concept remains useful and secure.

Table 5

User Stories

Archetypical User	Role	Primary Responsibility
Home Owner	- Smart home owner	- Manage and control smart home devices(lights) and security settings - Monitor notifications and alerts - Report suspicious activities or anomalies - Ensure privacy and data protection within the home - Cooperate with investigators by providing access or consent for forensic analysis
Forensic Investigator	-Digital forensics specialist	- Collect, preserve, and analyze IoT device and network data

		- Perform event reconstruction and timeline generation - Validate evidence integrity and chain of custody - Correlate multi-device evidence to establish incident chronology - Prepare event reports
Threat Actor	- Malicious intruder	- Introduce or manipulate events to disrupt normal operations in the smart home.

4.3.1.2 Modules Functional Overview

Table 6 provides the main modules found in the digital forensic investigation model for a smart home with event reconstruction. The table shows the module, its functionality, description, and purpose.

Table 6

Model Functional Overview

Functionality Name	Description	Purpose
User Registration	To facilitate user registration in the model and ensure secure access and account management.	- It provides feedback on registration success or failure - It stores user information in a secure database. - It validates user credentials during registration.
Log Management	Securely captures and stores log details, such as IP addresses and date-and-time	- It enables the model to capture the logs of all events and

	associations, in the model's database.	activities, including device operations.
Device Registration	To facilitate device registration in the model and ensure successful device addition.	- It establishes an accurate association between devices and their smart home context for reliable data acquisition and event reconstruction.
IoT Integration	To facilitate the integration of the various devices in the smart home.	- It enables comprehensive data acquisition from multiple IoT devices (DHTII, ESP32), supporting uniform analysis and evidence collection.

4.3.1.3 User Interaction and Interface Requirements

The model provides a web application accessible to all system users, as shown in Table 5. The UI is responsive design-wise, ensuring compatibility with desktop computers, tablets, and mobile devices. The layout includes forms, dashboards, and log displays for an intuitive user experience. The following are key features of the UI provided by the model;

i. **Dashboard:** Displays an overview of model modules, including logs, devices, and features.

ii. **User Registration and Login**: The model provides a secure registration process for new users, including homeowners and digital forensic specialists. A login page with email and password fields ensures secure access. Users can reset their passwords via email verification.

iii. **Device Registration:** The model provides a device registration process and a way to add devices to the smart home.

iv. **Log Monitoring Interface:** The page displays logs generated in real time.

4.3.1.4 Integration with External Systems: Hardware Integration

The following shows how the various hardware components were integrated within the model.

i. Microcontroller integration: The microcontroller serves as the central hub, enabling interaction among components. It collects raw data from the perception layer, preprocesses it, and sends it to a centralized server for storage over Wi-Fi. It also assigns timestamps to all activities in the smart home.

ii. Wi-Fi Integration: This component ensures devices connect wirelessly. Using Wi-Fi ensures that sensor data, device states, and microcontroller logs are continuously streamed without requiring physical connections. Furthermore, the Wi-Fi module is responsible for providing secure data transmission and synchronization with the forensic server. It supports encrypted protocols such as HTTPS, which protect evidence from tampering or interception during transit.

iii. Server Integration: This ensures that all the forensic data is collected and centrally stored. It receives processed logs, timestamps, device states, and event correlations, then stores them in a forensic database. This centralization ensures that all evidence from different smart home devices is stored in a single trusted location, making it easier to query, search, and correlate events during reconstruction.

4.3.1.5 Device Management

This section describes the management of various devices in the model.

i. Device Registration and Identification: Any device used in the model must be registered. The registration is done through the web application interface. A unique, autogenerated ID is assigned to the device to identify it in the model. The information is then stored in a central database, thus maintaining a centralized device inventory for forensic accountability.

ii. Configuration and Access Control: After a device is added, the initial configuration helps ensure that each device communicates only with a trusted controller. Proper configuration ensures that secure communication protocols, such as TLS, are enforced for Wi-Fi connections. This will help ensure that a device operates within a well-defined boundary, minimizing the risk of manipulation. On the other hand, access control determines who or what can interact with devices and what their logs record. It is implemented using role-based access control (RBAC), so that different users, that is, homeowners and the forensic investigator, have access to other parts of the model.

iii. Monitoring and Logging: This is critical since it provides raw evidence for forensic reconstruction. In the model, monitoring involves tracking device states and activities

such as temperature changes. Logging complements this by recording events with secure timestamps, enabling the reconstruction of their chronological order. The logs should be cryptographically hashed so that, in the event of modification, it is possible to detect it.

iv. Security and Firmware Management: Given the inherent vulnerabilities in IoT devices, such as outdated firmware, weak default settings, or unpatched vulnerabilities, the model can be an unreliable source of forensic evidence. Therefore, it is prudent to patch those vulnerabilities and have regular updates for those devices. Enabling secure boot and using digitally signed software will help to prevent malicious code injection.

v. Communication and Connectivity: Forensic models require secure, reliable communication among devices, the Raspberry Pi forensic hub, and the server. Smart home devices often use Wi-Fi, Bluetooth, or Zigbee, so these connections must be protected from eavesdropping and tampering. Using encrypted protocols such as TLS/SSL, VPNs, or DTLS, along with authentication tokens, helps confirm that data originates from trusted sources. Forensic logs must be sent in real time without gaps, and the data must be integrity-checked. This can only be possible when you have reliable connections.

vi. Fault Tolerance and Recovery: In the proposed event reconstruction model, fault tolerance and recovery are achieved through a combination of hardware resilience, software safeguards, and data management strategies designed to preserve forensic evidence even under adverse conditions. Each IoT device maintains a secure local cache of logs in non-volatile memory when offline, ensuring that data persists during power loss or network interruptions, and synchronizes with the forensic hub once connectivity is restored. The Raspberry Pi hub provides redundant storage and employs journaling techniques to recover incomplete transactions, while MQTT protocols with appropriate Quality of Service levels guarantee message delivery despite disruptions. To maintain evidentiary integrity, cryptographic hashing, such as HMAC and SHA-256, is used to detect tampering during restoration, and immutable storage policies prevent logs from being overwritten. Power-related risks are mitigated through battery backup systems and low-power logging modes, while heartbeat monitoring between devices and the hub enables rapid detection of failures and initiation of recovery actions. Upon restoration, devices resynchronize logs incrementally, and the forensic dashboard transparently flags recovered entries to preserve the chain of custody. Collectively,

these mechanisms ensure that event timelines remain complete, digital evidence is safeguarded against loss or manipulation, and attackers cannot easily erase forensic records by inducing device failures, thereby enhancing the reliability of investigations in smart home environments.

4.3.1.6 Security Functions

The following are the security features needed for the model's design.

i. User Authentication and Authorization: The model implements strong user authentication. During registration, each user is assigned a role, which is securely stored and managed by the model. Moreover, it is enforced by role-based access control, where each role is mapped to specific privileges. Authentication is implemented using two-factor authentication that requires a username and a password. This helps to ensure that access requests are validated against credentials and assigned roles. This helps to prevent unauthorized access.

ii. Data Confidentiality and Integrity: The model ensures that data collected from sensors, the server, and the microcontroller is protected against tampering. This is achieved through hashing, a process that generates a fixed-size string from data, enabling verification that the data has not been altered. Additionally, when the data is transmitted over wi-fi, the model uses the HTTPS protocol, which encrypts the information during transfer from the microcontroller to the cloud storage, making it unreadable to unauthorized parties.

iii. Event Validation and Anomaly Shielding: The model can filter out malicious activities before event reconstruction. This is done by validating sensor readings, for example, by checking a temperature sensor reading against others and flagging inconsistencies. This helps to prevent the attackers from suppressing real events or injecting false ones. Lastly, it helps to protect the accuracy of timelines.

iv. Data Backup and Recovery: The model implements data backup mechanisms via a local repository on the Raspberry Pi and secure cloud storage. This helps ensure data availability in the event of a security incident, such as device failures, accidental misuse, or cyberattacks.

4.3.2 Non-Functional Requirements

Table 7 below shows the non-functional requirements.

Table 7

Non-Functional Requirements

Requirement Category	Specific Requirement	Description
Performance	Response Time	The forensic model should reconstruct events and generate a timeline within 3 seconds after receiving device logs.
	Data Transmission Latency	Sensor and device event data (e.g., motion triggers, bulb status) transmitted via Wi-Fi from ESP32/Raspberry Pi to the forensic server should have a maximum latency of 2 seconds.
	Alert Delivery Time	Anomaly alerts (e.g., unauthorized access, tampering attempts) should be delivered to investigators within 5 seconds of detection.
	Concurrent Device Handling	The model should support at least 50 smart home devices reporting simultaneously without degradation in reconstruction accuracy or performance.

Reliability and Availability	Model Uptime	The forensic system should maintain 99.5% uptime to ensure continuous evidence collection and event logging.
	Data Backup Frequency	Automatic encrypted backups of forensic logs and reconstructed timelines should occur every 6 hours.
	Fault Tolerance	The Raspberry Pi and local logging modules should continue to capture and buffer evidence even if the forensic server is temporarily unavailable.
	Data Recovery	In the event of a system failure, the model should be able to restore evidence to the last backup period (6 hours).
Scalability	Scalable Architecture	The forensic server and event reconstruction engine should support horizontal scaling to handle increased device and user counts.
	Device Scalability	The system should support up to 200 IoT devices in a smart home environment without loss of performance.

	User Scalability	The forensic dashboard should support adding more investigator and administrator accounts without requiring significant architectural changes.
Usability	User Interface Design	The forensic dashboard should present reconstructed timelines in a clear, intuitive format, minimizing the need for investigator training.
	Mobile Compatibility	The dashboard should be fully responsive and accessible from mobile devices for field investigators.
Maintainability	Modular Code Structure	The forensic engine and server APIs should use a modular design to simplify updates and bug fixes.
	Logging and Monitoring	The model should maintain detailed logs of device activities, event reconstructions, and server performance to support maintenance and troubleshooting.

Interoperability	Standard Protocol Support	The system should support MQTT and RESTful APIs for integration with diverse IoT devices.
	Hardware Integration	The Raspberry Pi forensic hub should integrate with various sensors (motion, temperature, cameras) without requiring significant modifications.
	Data Export	The system should support exporting reconstructed timelines and logs in standard formats (CSV, JSON) for further forensic analysis.
Data Integrity	Validation Checks	Incoming device data should undergo validation checks to confirm expected formats and authenticity before reconstruction.
	Data Consistency	All updates to forensic databases should comply with ACID principles to avoid inconsistencies.
	Audit Logs	The system should maintain immutable audit logs of all data changes and reconstructions.

	Redundancy	Critical forensic data should be replicated across multiple storage nodes to minimize the risk of data loss.
Responsiveness	Real-Time Updates	The forensic dashboard should display event updates and reconstructed timelines in near real-time (≤5 seconds refresh rate).
	Dynamic Alerts	The anomaly detection module should adapt to changing device states and notify investigators instantly when suspicious events occur.
	Adaptive UI	The dashboard interface should dynamically adapt to different devices and screen resolutions, ensuring a smooth user experience across platforms.

4.3.3 Technical Requirements for the Model

Technical requirements are needed to construct the model. Table 8 below shows both the hardware and software requirements for the same.

Table 8

Model Technical Requirements

Requirement Category	Specific Requirement	Description
Hardware Requirements	ESP32 WiFi Bluetooth 2-in-1	A dual-function microcontroller that supports both WiFi and Bluetooth connectivity, enabling versatile communication options within the smart home environment. It has sufficient processing power and memory to handle real-time data and control tasks.
	5V 2 Channel Relay Module	Used to control high-power home devices, such as lights or alarms, remotely via the microcontroller. Supports two independent channels, enabling the simultaneous switching of two circuits with 5V control signals.
	LM2596HVS DC-DC Buck Converter	A voltage regulator module that steps down higher DC voltages to a stable 5V output, ensuring a safe and reliable power supply to the IoT microcontroller and connected peripherals.
	Li Ion Battery 18650 (3800mAh)	A rechargeable lithium-ion battery used as a backup

	power source for the smart home system, providing sustained operation during power outages with a high energy density suitable for extended use.
18650 Battery Holder Case	Protective casing for securely housing the 18650 lithium-ion battery, ensuring safe electrical contacts and preventing physical damage or short circuits within the hardware setup.
2mm Electric Wire	Insulated electrical wire with a 2mm thickness is used for reliable, safe interconnections between hardware components, ensuring low resistance and preventing signal loss or interference.
Bulb Holders	Fixtures designed to securely hold 5W bulbs in place and maintain electrical connections within smart lighting circuits, supporting easy installation and replacement.
5W Bulbs	Low-power light bulbs compatible with the relay module and bulb holders for

	energy-efficient, controlled lighting in the smart home.
Temp & Humidity Sensor	An environmental sensor module that measures temperature and humidity levels in the smart home, feeding crucial context data for event reconstruction and automation triggers.
Raspberry Pi 3 Model B+	A compact, affordable single-board computer used to host the forensic software stack locally. It features 1 GB RAM, 1.4 GHz quad-core CPU, and integrated Wi-Fi/Ethernet for network connectivity, making it ideal for DIY event reconstruction setups. Provides an energy-efficient and portable platform for real-time data processing and storage.
5V 3A Raspberry Pi 3 Adapter	Power adapter providing a stable 5V/3A power supply, suitable for powering Raspberry Pi or other microcontroller-based systems in the smart home, ensuring uninterrupted system operation under typical load conditions.

Software Requirements	Web Application Framework	Use Flask, a lightweight and easy-to-learn Python web framework, to develop the user interface and backend APIs. Flask supports rapid development, modular design, and integration with IoT devices.
	Backend Logic and API Handling	Implement backend logic in Python using Flask to manage data ingestion from IoT devices, perform event correlation, and expose RESTful APIs for frontend communication.
	Data Storage	Use MongoDB as a NoSQL database for flexible storage of diverse forensic data, including device logs, event metadata, and user information. Local or cloud-hosted instances can be used.
	Data Integrity Features	Integrate basic cryptographic hashing (SHA-256) on collected logs and reconstructed events to verify data integrity and detect tampering during forensic analysis.
	User Authentication	Implement simple authentication mechanisms

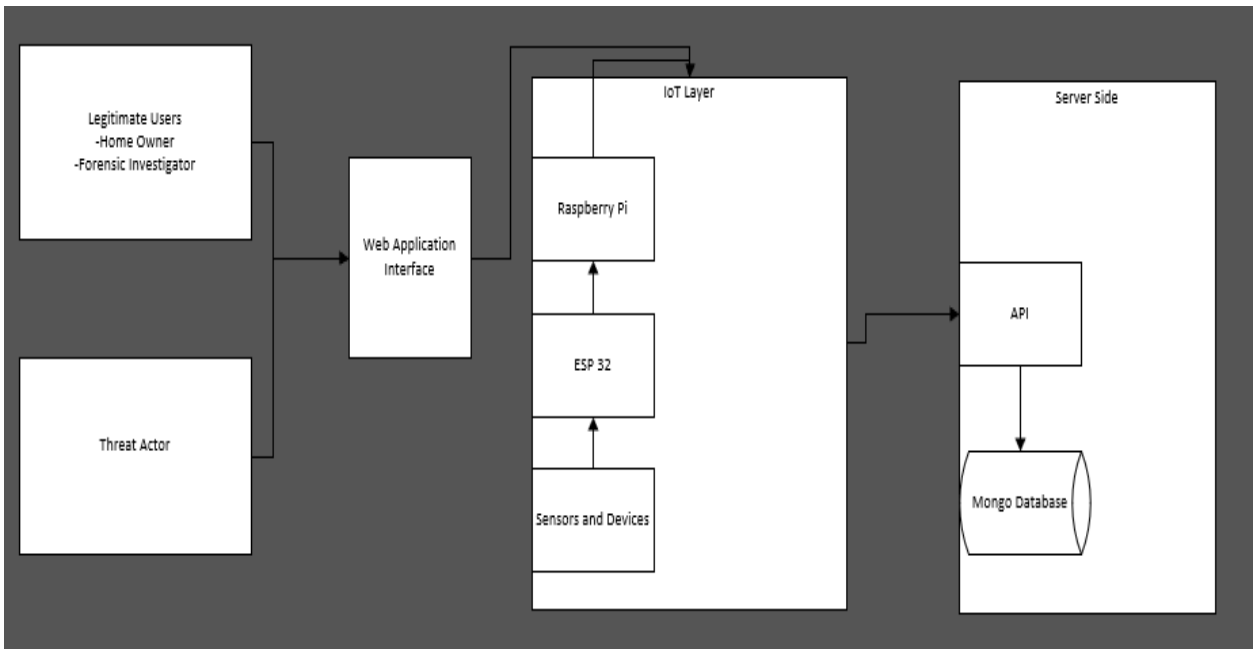
	using Flask-Login to secure the forensic interface and restrict access to authorized users only.
Visualization Dashboard	Develop a straightforward web dashboard using Flask templates or external JavaScript libraries (e.g., Chart.js) to visualize event timelines and correlation graphs for more straightforward interpretation.
IoT Device Communication	Use MQTT for testing and HTTPS for reliable, secure communication between IoT devices and the Flask API server, ensuring real-time data updates and command execution.
Logging and Audit Trails	Enable application-level logging of all user actions, API requests, and data changes to assist with accountability and basic chain-of-custody documentation.
Backup and Recovery	Employ automated scheduled backups of the MongoDB database and essential files to local or cloud storage (e.g.,

	Cloudflare) to prevent data loss in case of failures.
Software Update Mechanism	Set up manual or automated update processes for the Flask app and related components (e.g., GitHub Actions) to keep the system secure and up to date with bug fixes and security updates.
Lightweight Local Development Environment	Support running the entire software stack locally on a Raspberry Pi or personal PC, allowing offline forensic investigations and easier debugging during development.

4.3.4 Model Design Architecture

Figure 4

Model Architecture



The event reconstruction model used a layered architecture, enabling efficient operation. This architecture integrates components such as the user layer, application layer, IoT layer, and server side, all working together to reconstruct events. The user layer consists of two categories: legitimate users (such as home owners and forensic investigators) and threat actors. Legitimate users access logs and reports, while threat actors attempt to attack the system, thereby generating the events that are later reconstructed. They interact with the application layer through the web application interface.

At the center of the model is the IoT layer. This layer includes the Raspberry Pi microcontroller, ESP32 Wi-Fi modules, and various sensors and devices. The Raspberry Pi acts as the local database cache, forensic hub that correlates events, runs anomaly detection, and acts as the gateway. The ESP32 collects sensor data, sends commands, and connects the components via Wi-Fi. Lastly, the sensor and devices send raw data to the ESP32.

At the back of the model is the server-side. The server-side is responsible for storing and retrieving data. An API server collects data from the IoT layer and stores it in the MongoDB database. Additionally, the database stores information about users, IP addresses, and event timestamps. All this information is accessible to homeowners and forensic investigators via the web application 'IoT Log Monitor.' The integration of these components ensures effective event reconstruction in a smart home.

4.4 Development of the Event Reconstruction Model

This section addresses objective three of the study. The objective was to implement event reconstruction within a digital forensic investigation model for a smart home. To achieve this objective, rapid application development and experimentation were employed. This proved that the project was technically feasible to implement. This section aims to demonstrate the deployment of a functional model that will collect event logs and correlate them to reconstruct the events.

The model deployment was done in two parts: hardware and software. The hardware was used to demonstrate how various components were to be integrated to generate real-time event logs. At the same time, the software involved developing a web application interface that enabled users (homeowners, forensic investigators, and threat actors) to interact with the model efficiently.

This section covers the entire development process, from the hardware components to the software components and their integration. Additionally, the deployment strategy will be addressed. Lastly, challenges encountered during deployment will be discussed, along with the corresponding mitigation strategies.

Appendix II contains the project's sample code.

4.4.1 Hardware Setup and Configuration

This section covers how the hardware components were integrated. Table 9 below shows the purpose of the hardware components and their setup.

Table 9

Set up steps for Hardware

Hardware Device	Purpose	Initial Setup Steps
Raspberry Pi 3 Model B+	Central controller for the smart home forensic model; runs data logging, event reconstruction scripts	Install Raspberry Pi OS on the 64GB SD card using Raspberry Pi Imager Enable SSH and Wi-Fi via raspi-config. Update system (sudo apt update && sudo apt upgrade). Install Python 3, pip, and the MQTT/HTTP libraries. Enable GPIO for communication if needed.
64GB MicroSD Card	Storage for OS, logs, and forensic data	Format using FAT32. Flash Raspberry Pi OS (Lite or Full) with Raspberry Pi Imager 3.

		Insert the Raspberry Pi SD card into the SD slot before powering on.
ESP32 Development Board	Captures IoT sensor data (temperature, motion, bulbs, etc.) and transmits it to the Pi via Wi-Fi.	Install Arduino IDE or PlatformIO on Pi. Install ESP32 board support in Arduino IDE. Flash an ESP32 with firmware that sends sensor data to a Raspberry Pi via MQTT or HTTP. Configure the ESP32 to connect to the same Wi-Fi as the Pi.
5V 2A Power Supply (for Pi)	Provides stable power to Raspberry Pi	Connect the official 5V/2A micro-USB or USB-C power adapter to Pi 2. Ensure uninterrupted power using a UPS or battery backup.
18650 Li-Ion Batteries (with Holder)	Backup power source to keep the system alive during outages	Use a UPS HAT for Raspberry Pi or a power bank with passthrough charging. Insert 18650 batteries into the holder.

		Connect the holder to the UPS module or the DC-DC converter (LM2596).
LM2596HVS DC-DC Buck Converter	Step down the voltage from the battery pack to 3.3V for the ESP32	Adjust output to 3.3V for ESP32. Connect the input to the battery pack and the output to devices.
Relay Module (5V, 2-Channel)	Controls home appliances (bulbs, etc) for event simulation	Connect relay input pins to ESP32 GPIO. Provide 5V power from Pi or buck converter. Test with a simple ON/OFF script in Arduino IDE.
Wi-Fi Router (or Hotspot)	Network for Raspberry Pi and ESP32 to communicate	Connect the Raspberry Pi to Wi-Fi via raspi-config. Configure ESP32 with the identical SSID and password. Verify connectivity by pinging the ESP32 from the Raspberry Pi.

Figure 5

Raspberry Pi Microcontroller

Figure 5 above shows the Raspberry Pi microcontroller. It serves as the central controller for the smart home system, running the main programs and managing communication among devices. Additionally, it handles data processing, user interface, and interaction with other peripherals and external systems, such as cloud services.

Figure 6

ESP 32 Microcontroller

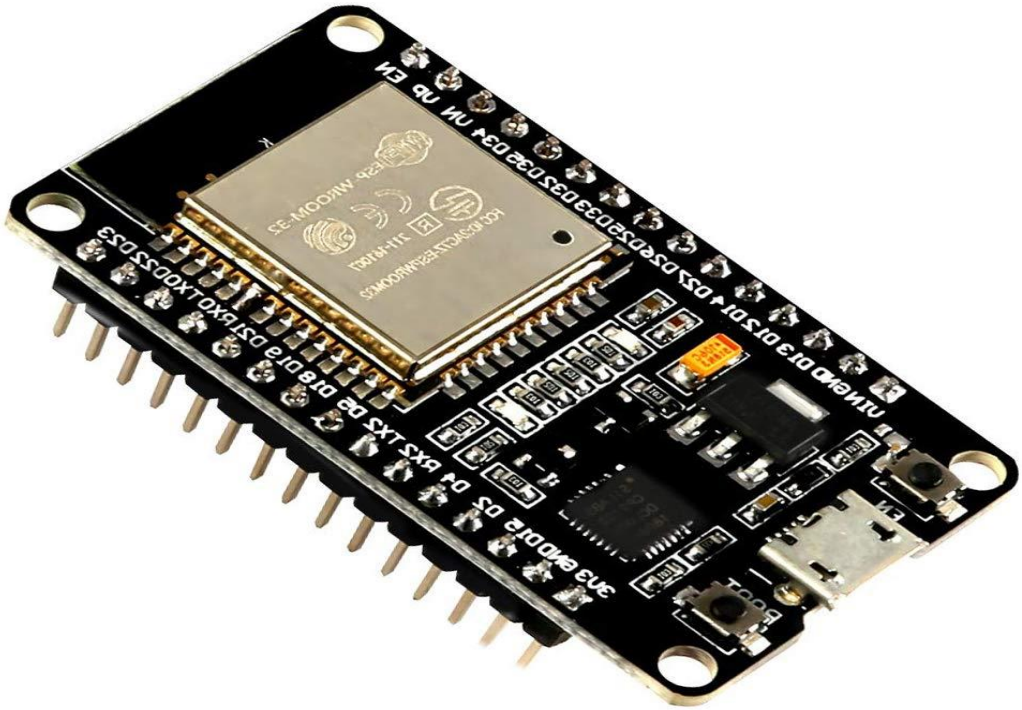


Figure 6 above shows the ESP32, a 2-in-1 Wi-Fi and Bluetooth microcontroller. The microcontroller is used to establish wireless communication in the smart home environment. It connects sensors and actuators to the local network or the internet, allowing remote control and monitoring of the smart home system. Furthermore, it controls operations such as turning the smart home lighting system on and off over the internet and collects and updates temperature readings on the web interface.

Figure 7

DHT II Sensor

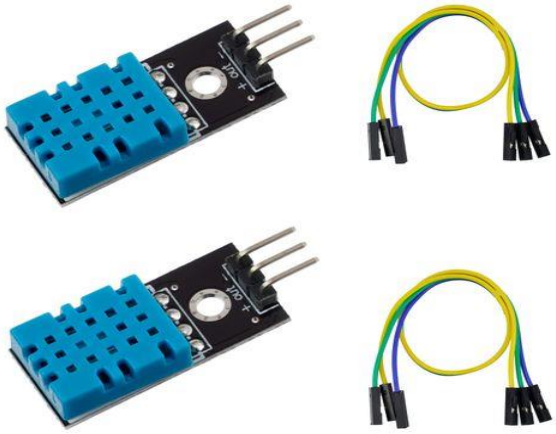


Figure 7 above shows a DHT II sensor. It measures temperature and humidity in the smart home.

Figure 8

Relay

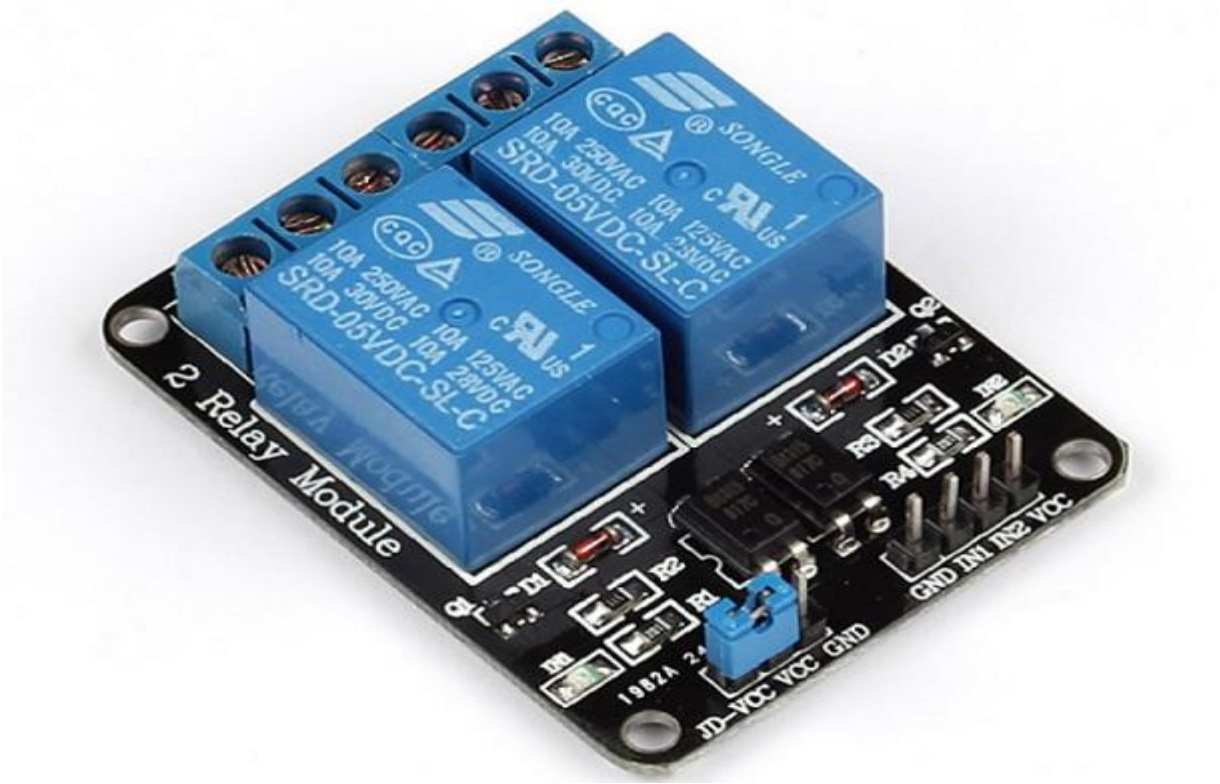


Figure 8 above shows the relay. The relay acts as a switch, controlling high-power devices such as lights, fans, and home appliances. It is activated by the microcontroller (ESP32 or Raspberry Pi) to turn devices on or off via our web interface remotely.

Figure 9

Buck Converter

Figure 9 above shows the buck converter. The converter is a step-down voltage regulator that converts higher input voltages to lower ones suitable for powering components like the ESP32 or sensors. Additionally, it ensures efficient power management for our smart home model.

4.4.2 Software Set up

For the hardware to function correctly, the software had to be configured for all components. The components were the web application interface, the API server, and the database. Table 10 below shows the components and their configurations.

Table 10

Software Implementation

Component	Framework	Purpose	Implementation
Web Application Development	Flask	A web server for handling client requests and a dashboard.	Hosted on VPS (Cloudflare) and proxied using nginx.
API Server Development	Flask	Server for handling API requests from the device and the clients.	Defined endpoints (/log-event, /compare-events). ESP32 sends HTTP POST requests over Wi-Fi.
Database Configuration (MongoDB)	MongoDB	Storage for data	Hosted on Atlas Cloud for MongoDB. Significant improvement in speed and availability.
Integration of IoT Devices with the API Server	MQTT	Transport for device logs and messages	Used a public MQTT server with MQTTX, which significantly reduced transport infrastructure costs. The messages and logs are encrypted on the device and decrypted on the server to minimize the chances of unauthorized access.

4.4.3 Software Modules

This section covers the modules that were included in the model. Accordingly, the modules developed ensured the model functioned efficiently.

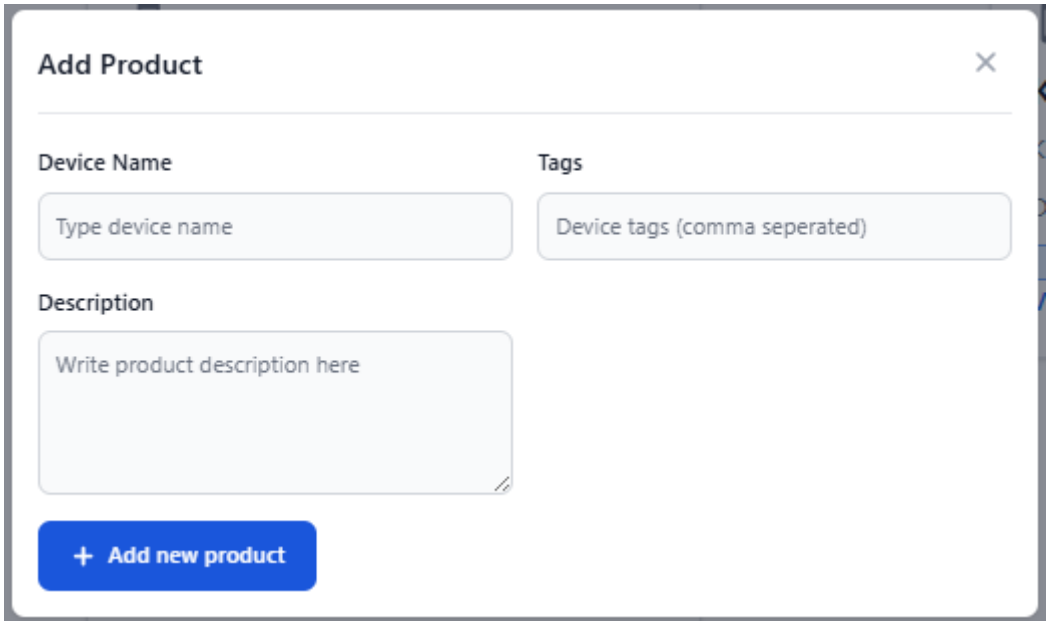
4.4.3.1 Device Registration

This module is a key component in the model. It enables devices to be added to the smart home for management. When a device is added, this module automatically generates a unique ID that serves as the device's digital fingerprint. This unique ID enables tracking each device, thereby improving the model's integrity. Integrity is maintained, as there is no confusion or overlap

among devices during event reconstruction. Therefore, a forensic investigator can pinpoint which device was involved and when. The figure below shows how a device is added to the model.

Figure 10

Device Registration



4.4.3.2 User Registration and Authentication Module

This is a critical component in the model. It handles the registration and verification of individuals who have authorized access to and control of smart home devices. In the model, only the homeowner (administrator) can add users. To add a user, they must provide an email address and name, and they will be assigned a role based on their responsibilities. The model has two leading roles: administrator and an ordinary user. The administrator has more privileges than the ordinary user.

When a user is successfully registered and accesses the model, the module securely collects and stores relevant user information and associates each user with permissions within the model. This linkage is critical in an investigation because it provides a clear trail of user activities. As a result, it becomes easy to attribute actions to an individual. This action not only enhances accountability but also helps identify misuse and potential points of intrusion. Figure 11 below shows the registration page, and Figure 13 shows the login page.

Figure 11

User Registration

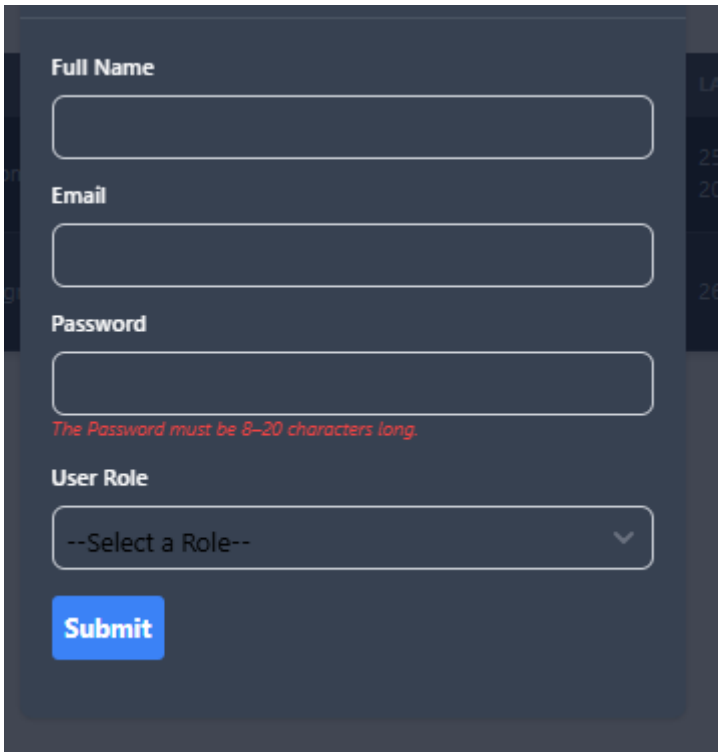
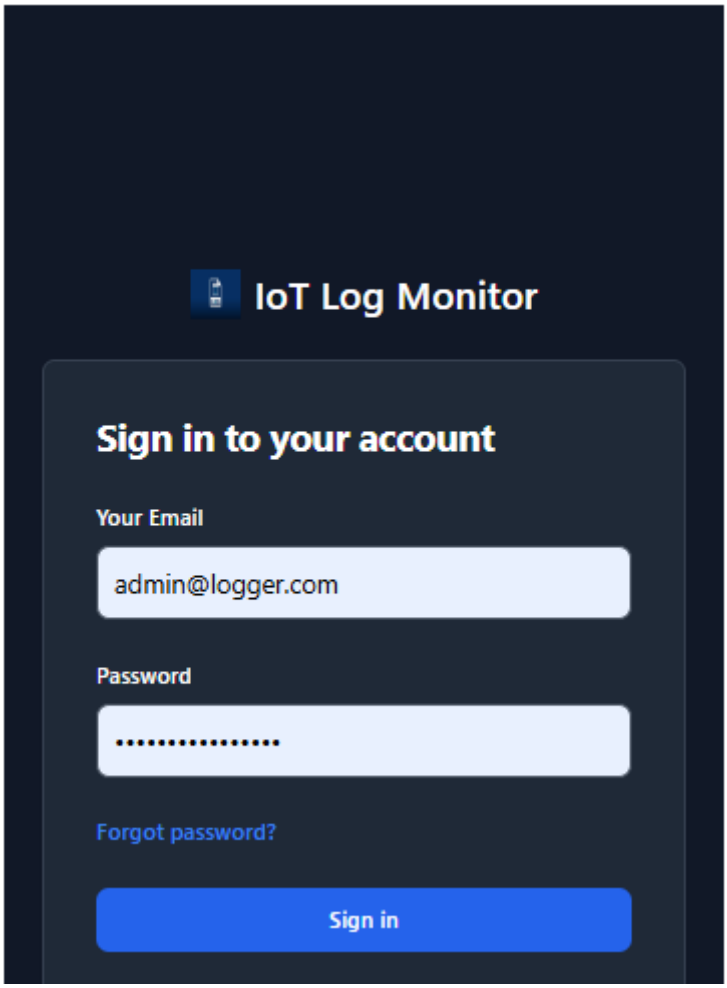


Figure 12

User Login Page

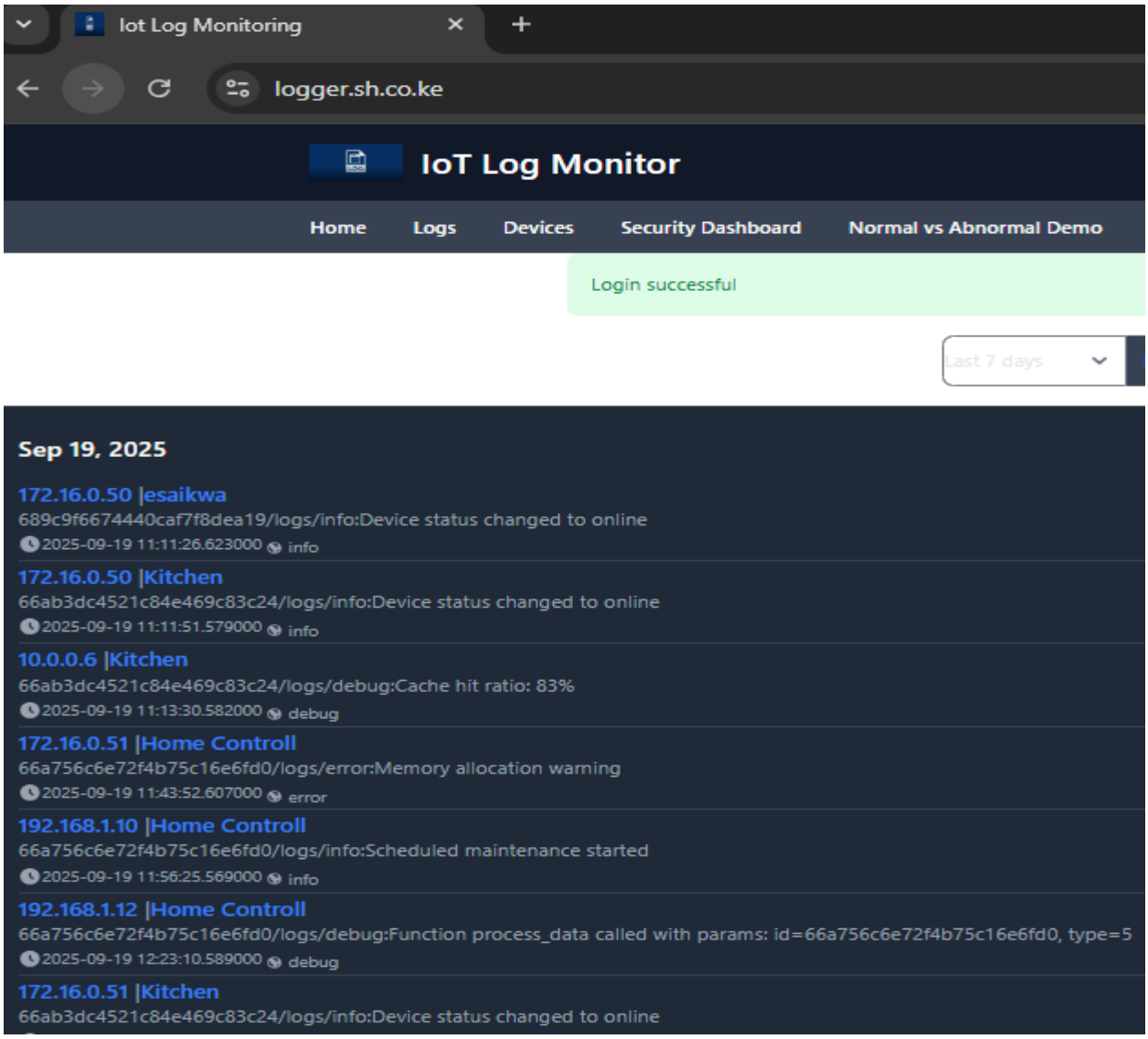


4.4.3.3 Logs Monitoring Module

This is another vital module in the model. The module continuously tracks and records data from all connected devices and users. Just like an observer, it captures real-time stamped entries that paint a detailed, chronological timeline of the smart home's operations. This information is key to event reconstruction because it enables forensic investigators to sift through logs and correlate activities, thereby identifying anomalies or specific events. The IoT log monitor ensures that no event goes unnoticed, laying a solid foundation for thorough, reliable forensics. Figure 13 below shows a sample capture of the logs.

Figure 13

Sample Capture of Logs



4.4.3.4 Security Dashboard

The security dashboard provides interactive visualizations of IoT device logs for security and event reconstruction. Time, device, and severity levels can filter the logs. Furthermore, graphs have been used to show log activity by type and severity distribution, with levels 0 to 5 (Level 0 indicates no security event, and Level 5 indicates a high likelihood of a security event). Additionally, the anomaly scores are shown, and when one hovers over the graph, they can see the specific event that triggered it. Also, one can view the number of activities generated by a device in the model, along with the correlated events. Figures 14, 15, and 16 show the graphical representations.

Figure 14

Log Activity and Severity Distribution

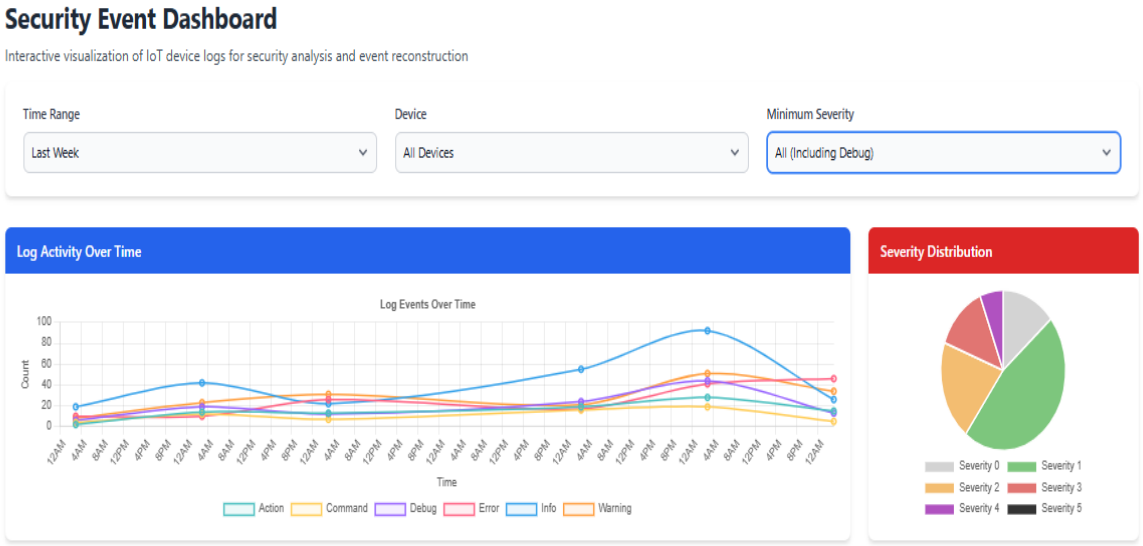


Figure 15

Anomaly Scores

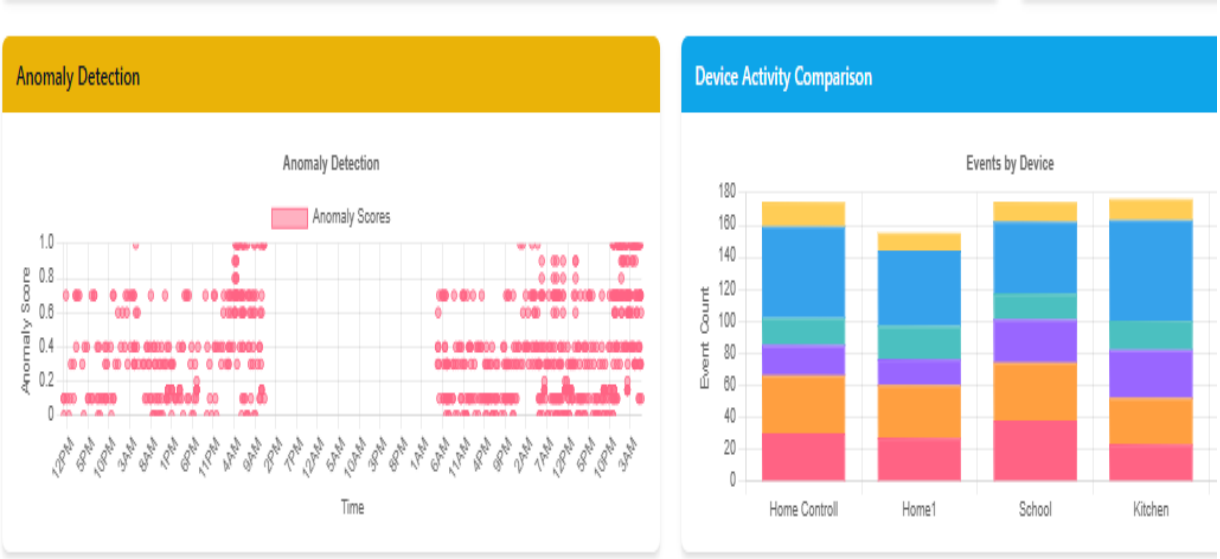
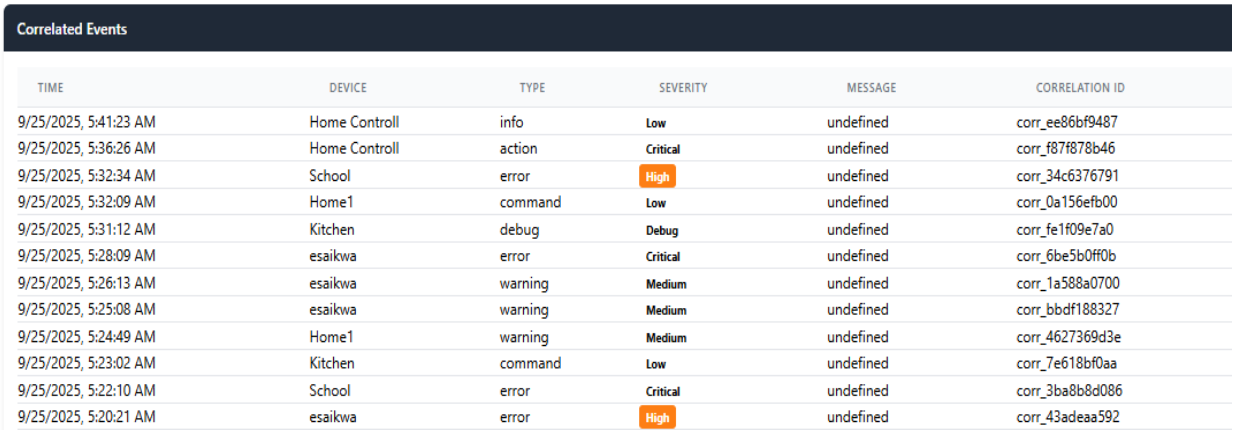


Figure 16

Event Correlation



Correlated Events

TIME	DEVICE	TYPE	SEVERITY	MESSAGE	CORRELATION ID
9/25/2025, 5:41:23 AM	Home Controll	info	Low	undefined	corr_ee86bf9487
9/25/2025, 5:36:26 AM	Home Controll	action	Critical	undefined	corr_f87f878b46
9/25/2025, 5:32:34 AM	School	error	High	undefined	corr_34c6376791
9/25/2025, 5:32:09 AM	Home1	command	Low	undefined	corr_0a156efb00
9/25/2025, 5:31:12 AM	Kitchen	debug	Debug	undefined	corr_fe1f09e7a0
9/25/2025, 5:28:09 AM	esaikwa	error	Critical	undefined	corr_6be5b0ff0b
9/25/2025, 5:26:13 AM	esaikwa	warning	Medium	undefined	corr_1a588a0700
9/25/2025, 5:25:08 AM	esaikwa	warning	Medium	undefined	corr_bbdf188327
9/25/2025, 5:24:49 AM	Home1	warning	Medium	undefined	corr_4627369d3e
9/25/2025, 5:23:02 AM	Kitchen	command	Low	undefined	corr_7e618bf0aa
9/25/2025, 5:22:10 AM	School	error	Critical	undefined	corr_3ba8b8d086
9/25/2025, 5:20:21 AM	esaikwa	error	High	undefined	corr_43adeaa592

4.4.3.5 Anomaly Detection

The anomaly detection dashboard provides a visualization of normal and abnormal patterns and verifies the integrity of logs. Accordingly, the model can filter data based on the mentioned parameters, which are visualized in graphs as shown in Figure 17. Forensic investigators can focus on specific areas of interest because they can conduct causal analysis. Moreover, an event timeline graph is generated, as depicted in Figure 18, and, lastly, Figure 19 shows the verification of logs, which have been done using HMAC and verified using bcrypt.

Figure 17

Normal vs Abnormal Operations

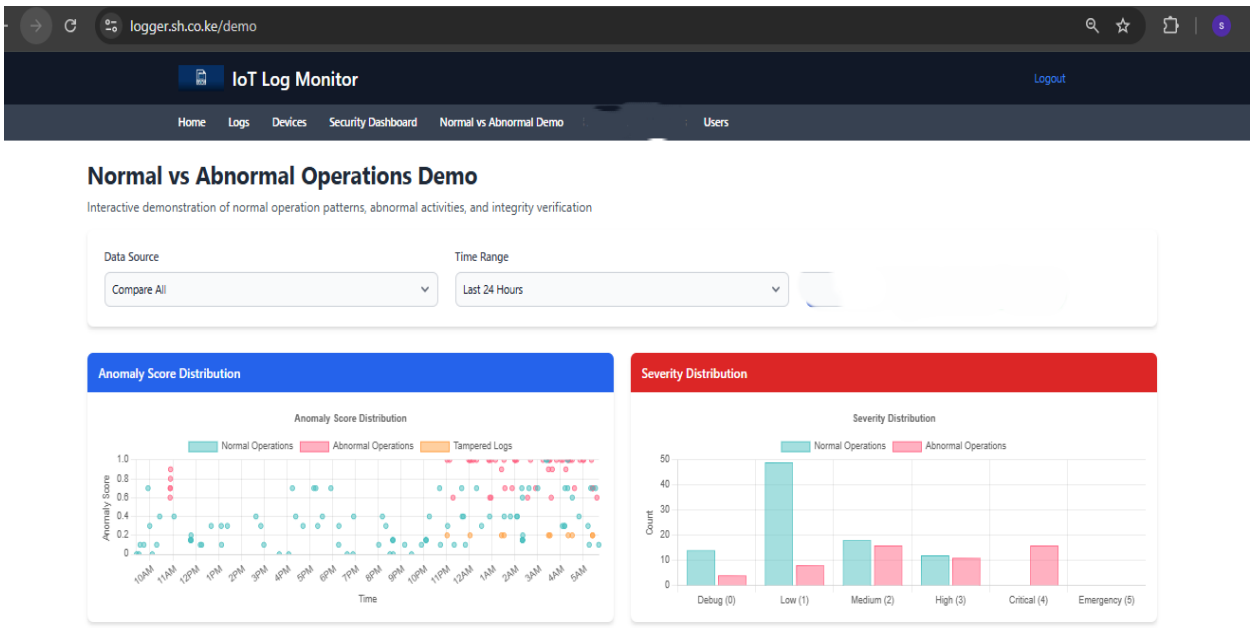


Figure 18

Event Timeline

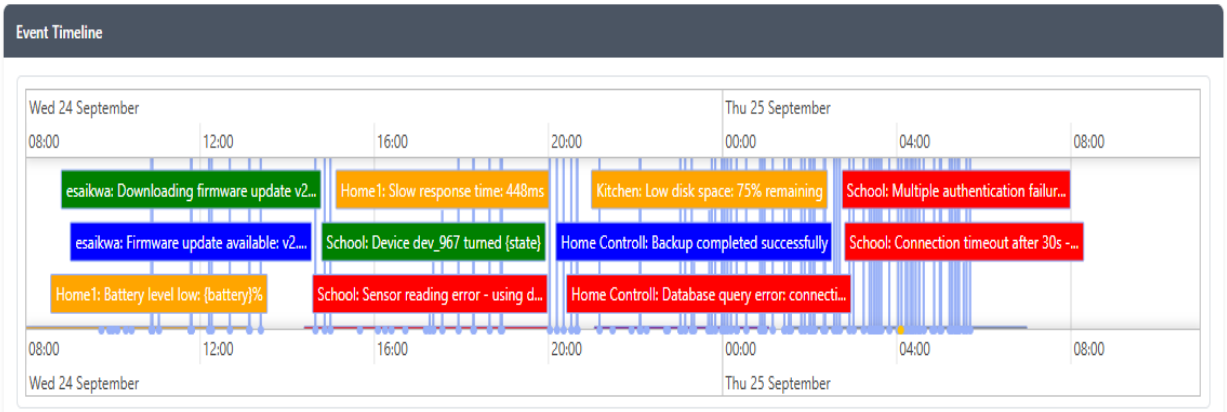


Figure 19

Verified Logs

Log Entries — Integrity Check: All 155 logs verifie

TIME	DEVICE	TYPE	SEVERITY	ANOMALY	MESSAGE	INTEGRITY
9/24/2025, 9:44:11 AM	Home Controll	info	Low	0.10	Scheduled maintenance started	Verified
9/24/2025, 9:55:11 AM	Home Controll	error	High	0.70	Failed to update cache - using stale data	Verified
9/24/2025, 9:59:35 AM	Kitchen	command	Low	0.30	UPDATE firmware version=3.2.4	Verified
9/24/2025, 10:06:11 AM	School	debug	Debug	0.00	Thread pool status: 79/35 active	Verified
9/24/2025, 10:17:55 AM	Home1	info	Low	0.10	Configuration updated by system	Verified
9/24/2025, 10:25:46 AM	Home1	warning	Medium	0.40	Connection attempt rate limiting applied	Verified
9/24/2025, 10:53:14 AM	School	info	Low	0.60	User admin authentication attempt	Verified
9/24/2025, 10:53:26 AM	School	warning	Medium	0.70	Invalid credentials for user admin	Verified
9/24/2025, 10:53:50 AM	School	warning	Medium	0.70	Multiple failed login attempts for user admin	Verified

4.4.4 Deployment of the Model

The software was built with Python as the core programming language, leveraging Flask as the web development framework to create an accessible online interface found at (https://logger.sh.co.ke/login). The interface is structured into several main pages for ease of use and functionality. The Home Page acts as the entry point, providing users with general information about the system and navigation options. The Logs Page is crucial for forensic investigations, as it displays detailed records of all events and activities within the smart home,

including device operations, timestamped for accurate event reconstruction. Users can filter these logs by parameters such as IP address, date, and time, enabling efficient pinpointing of relevant activities during forensic analysis.

The **Device Page** lists all the smart devices connected to the home network. Every device added is assigned a unique, auto-generated Device ID that ensures clear identification across the system. Users can control devices directly on this page, performing actions such as turning devices on or off, with every interaction logged for traceability. Additional metadata, such as device location and condition, can be added to improve management and provide context during investigations. The Security Dashboard showcases the visualization of security event logs. Events can be filtered by time, device, and severity level. In addition, there is a dashboard that compares regular events and abnormal events generated by the devices in the smart home. From this dashboard, one can see events as they occur in real time.

On the backend, IoT integration modules collect data from the smart devices and relay it to a central server. This setup allows all device data to be accessed and controlled remotely via the online interface, providing a unified management platform. To validate the system, MQTTX was used as a simulation tool for the smart home environment. Device IDs generated by the web interface were used to simulate device control commands, such as switching lights on or off, and verifying that all generated logs accurately reflect these operations.

Moving beyond simulation, a physical prototype was constructed, incorporating sensors such as the DHT11 for temperature monitoring and relay modules for lighting control. Each component was independently tested to ensure proper functionality, such as accurate temperature readings and reliable light switching. Once verified, these hardware elements were integrated into the smart home system, completing the deployment by providing real-world interaction capability with the software.

This combined software and hardware deployment enables a comprehensive, functional forensic investigation model that captures and records all smart home activities, supports event reconstruction through detailed logging, and provides remote control and monitoring via an intuitive web interface. The deployment demonstrates the practical application of the model from development and simulation to physical implementation in a smart home environment.

4.4.5 Challenges Faced in Model Implementation

Table 11 below shows the challenges that were encountered

Table 11

Challenges Faced During Implementation

Challenge	Description
Hardware and Software Integration	Integrating physical smart home devices with the software interface was a significant challenge. Ensuring sensor data and device states were accurately transmitted and reflected online required multiple troubleshooting attempts. Communication issues between hardware modules and the web interface, particularly for real-time logging and device control, had to be carefully resolved.
Data Synchronization	Maintaining consistent synchronization between smart home sensors and the central server was difficult, especially during simultaneous operations across multiple devices, leading to potential data inconsistencies in logs and control responses.
Sensor Accuracy and Calibration	Calibrating sensors like the DHT11 for accurate temperature readings and ensuring the reliable operation of relay modules required extensive testing before they could be fully integrated and trusted within the system.

4.5 Model Evaluation

4.5.1 Goal-Based Evaluation

This section addresses model evaluation as mentioned in section 3.3.4. Accordingly, the model was set up in a controlled environment and evaluated using goal-based evaluation. Goal-based

evaluation helps to test a system vis-à-vis the objectives. Accordingly, Table 12 below shows the model evaluation criteria.

Table 12

Goal-Based Evaluation

Objective	Evaluation
Device Registration: The objective of this module is to add devices to the smart home. Accordingly, the model was able to add devices by assigning them an autogenerated unique identification for traceability.	The model successfully added devices to the smart home. The devices were assigned unique identification numbers. This enabled easier event correlation in the model, as it was easier to identify which device was used for a particular purpose, thereby improving investigation completeness and accuracy.
User Registration and Authentication: The objective is to securely register users (homeowners) in the model and authenticate them when they want to access the model.	The model successfully added and authenticated users. As a result, high authentication success with minimal errors ensured only authorized access, thus preserving data confidentiality and auditability.
Log management: The objective of this module is to collect, timestamp, and securely store device logs for later forensic analysis within the model.	The model successfully captured logs generated in real time from the various devices in the smart home. The logs were securely transferred from the devices to the database for forensic storage.
Event Correlation: The objective of the module was to correlate logs and events from devices into coherent timelines for reconstruction	The model effectively leveraged unique IDs, timestamps, and contextual data to reconstruct event sequences, facilitating precise causal analysis accurately.
Alert Notification: The objective of this module was to notify the homeowner whenever changes were detected in the smart home.	The model efficiently triggered alerts upon detecting anomalies or important events via the web application, with low false alarm rates, enabling immediate response.

4.5.2 Expert Survey using Validation Metrics

This section addresses the validation metrics described in section 3.3.4. An expert survey was conducted with one forensic expert. Table 13 shows the evaluation metrics.

Table 13

Evaluation based on Metrics

Evaluation Metric	Description	Evaluation Questions for Investigators	Response
Accuracy	Measures the correctness and error-free output of the model.	How often does the model produce error-free and precise results?	The model achieved 85% accuracy using F1 scores.
		Are false positives or false negatives observed during investigations?	Yes
		How accurately does the model associate evidence with correct devices or users?	81%
Auditability	Assesses the transparency and verifiability of forensic actions and results.	Does the model maintain detailed and tamper-evident logs of all forensic steps?	Yes
		Can external auditors verify the integrity and sequence of investigation procedures?	Yes
		How transparent are evidence handling and chain-of-custody records?	The DB can be accessed through a separate interface with hashes fully available.
Completeness	Evaluates how comprehensively the model recovers digital evidence.	Does the model identify and extract all relevant evidence from IoT devices?	Most. Only a few are not well formatted.
		Are there gaps or missing components of evidence?	No
		How effective is the model at detecting	Most of the common patterns are covered.

		hidden, volatile, or encrypted data?	
Reliability	Checks stability and consistent operation under investigation conditions.	Does the model consistently operate without failure or crashes?	A few failures were corrected.
		How does it perform under concurrent or high-load scenarios?	It performed generally well, but performance decreased with increasing load.
		Are there known scenarios causing unreliable results?	Most of them were corrected before the final version.

4.5.3 Comparison with Other Models

The section addresses how the developed model compares with existing models. Table 14 below shows the comparison.

Table 14

Comparison with Other Models

Model Name (Citation)	Key Features & Approach	Strengths	Limitations	Comparison with Current Model
Current Evaluated Model (2025)	Modular architecture integrating device control, comprehensive logging, IoT data integration, MQTT simulation, and a physical prototype in a	Practical integration of software and hardware; absolute device control, alert notifications; includes timeline-based event reconstruction; grounded by	Primarily tested in a controlled environment; limited large-scale field validation.	Offers the strongest practical implementation, device control, alerting, and explicit event reconstruction capabilities, making it more comprehensive than others.

	controlled environment.	goal-based metrics.		
1-2-3 Zone & Next Best Thing Triage (Oriwoh et al., 2013)	Zone-based forensic triage approach prioritizing investigation focus.	Efficiency in managing large or complex network investigations.	No IoT device heterogeneity handling or real-time forensic controls; lacks event reconstruction.	Lacks event reconstruction and device control; the current model offers complete practical control and event reconstruction.
Top-Down Forensic Approach (Perumal et al., 2015)	Four-tier forensic approach emphasizing volatile data protection and network zones.	Addresses preservation of volatile and hidden data critical to timeline accuracy.	Mainly conceptual; no integrated physical device management or alerts; event reconstruction limited.	Provides foundational concepts; the current model adds complete physical device control and detailed event reconstruction.
Forensics-aware IoT (Zawoad & Hasan, 2015)	Distributed trusted repository with provenance modules focused on evidence integrity and chain of custody.	Strong provenance and secure evidence storage.	Focused on backend systems; lacks on-device control, real-time alerting, and explicit event reconstruction.	Complements integrity with physical controls and rich event reconstruction, which is missing in Forensics-aware IoT.
Generic IoT Forensic	Layered approach for	Comprehensive coverage across	Complex, less hands-on	Provides standards

Framework (Kebande & Ray, 2016)	proactive readiness and reactive investigations compliant with ISO standards.	device, network, and cloud layers.	physical control; event reconstruction not fully addressed.	compliance; the current model offers practical device control and integrated event reconstruction.
Application-Specific IoT Forensics Model (Zia et al., 2017)	Application domain-specific framework addressing synchronization, privacy, and environmental dynamics.	Flexibility to tailor to applications; privacy-focused.	Limited generalizability; partial event reconstruction only.	More generic and integrated physical reconstruction and alert system compared to a domain-specific focus.
Smart Home Forensics Framework (Goudbeek et al., 2018)	Framework for sensor data acquisition and timeline-based event reconstruction within smart homes.	Detailed timeline visualization and reconstruction for smart home contexts.	No active device control or alert notification features.	Adds interactive device control and alerting, alongside event reconstruction, enhancing forensic operations in smart homes.
IoT Forensic Investigative Framework (Sathwara et al., 2019)	Open-source toolchain-based framework supporting the full forensic lifecycle from	Transparency, reproducibility, and tool flexibility.	Limited physical device control and alerting; lacks detailed event reconstruction.	Incorporates physical control and event reconstruction, which are missing in tool-

	acquisition to presentation.			centric frameworks.
Open-Source IoT Digital Forensics (Al-Sadi et al., 2018)	Reliance on open-source forensic tools for evidence acquisition and analysis.	Cost-effective and flexible tool-based forensic operations.	Dependent on tool availability; no integrated physical prototyping or reconstruction.	Adds integrated physical prototyping, control, and event reconstruction to tool-based methods.
IoTDots Framework (Babun et al., 2018)	Lightweight event logging and correlation for resource-constrained IoT devices.	Efficient, low-overhead logging with event correlation.	Logging only; lacks device control, alerting, and complete event reconstruction.	Extends basic logging to full event reconstruction with active control and alerting mechanisms.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

This chapter presents the conclusion and recommendations for the research on event reconstruction in a digital forensic investigation model for a smart home design, implementation, evaluation, and results. It then provides suggestions for further study.

5.2 Summary

This study investigated integrating event reconstruction into a digital forensic investigation model for smart home environments. Recognizing that IoT devices often transmit but do not store data, the research addressed the challenge of reconstructing events during cyber incidents. Using a multi-methodological approach—systematic literature review, design science research, and rapid prototyping—the study designed and implemented a proof-of-concept model incorporating user and device registration, real-time log management, anomaly detection, and timeline generation. Evaluation demonstrated that the model successfully captured logs and reconstructed event sequences, thereby enhancing the reliability and efficiency of forensic investigations in smart homes. Despite challenges such as data synchronization, sensor calibration, and hardware/software integration, the study concluded that event reconstruction significantly improves forensic readiness and evidentiary reliability in IoT environments.

5.3 Conclusions

This section presents the conclusions of the study in accordance with the specific objectives. The findings in this study about each of the objectives are presented in this section, and they are summarized as follows:

5.3.1 Investigation of existing models of digital forensics on the Internet of Things devices

Research question 1: What models for digital forensic investigation of Internet of Things devices exist?

This study reviewed various models of digital forensic investigation for the Internet of Things. To achieve this objective, secondary data was used. This highlighted the challenges inherent in the multiple models. A majority of the models reviewed focused on reactive aspects of digital forensics and were heterogeneous; they lacked standardization. Furthermore, some had limited adaptability, some were platform-specific, and some lacked support and training. In addition,

a majority of them were high-level and had not been tested. This review helped identify the IoT device used in the study's simulation.

5.3.2 Design of a digital forensic investigation model with Event Reconstruction

Research Question 2: How can a digital forensic investigation model for a smart home with event reconstruction be designed?

To address research question 2, the model was implemented systematically. First, the key activities in the proactive digital forensic process were identified, informed by the literature review. Further, the devices and data types to be collected during implementation were identified. The various sources of data were categorized in terms of their nature, that is, sensor data, device logs, and IP addresses, to facilitate easier retrieval and analysis of logs for event reconstruction.

Data security was also crucial during the design. This was achieved by hashing the data sent from the devices to the IoT central server. Additionally, communication between the devices was via Wi-Fi.

5.3.3 Implementation of the Digital Forensic Investigation Model for a Smart Home with Event Reconstruction

Research Question 3: How can a digital forensic investigation model for a smart home with event reconstruction be implemented?

Regarding this study, objective three sought to implement the digital forensic investigation model for a smart home with event reconstruction. The model was developed and implemented using rapid prototyping, which enabled quick testing and validation of concepts. Consequently, the implementation was done in two stages. First, the software was divided into modules. A registration module was developed to register new devices. The module also included device control, such as turning them on or off. Also, device condition and location could be viewed from this module. The other module was for log management, where all logs generated by the devices could be gathered to enable event reconstruction.

Upon completion of software development, the MQTTX simulator was used to simulate the smart home. The various devices were added and integrated before deployment in the physical devices. More so, each component was tested independently to ensure accurate functionality.

Finally, the elements were incorporated into a smart home. This covered the physical implementation.

5.3.4 Evaluation of the Digital Forensic Investigation Model with Event Reconstruction

Research Question 4: How can a digital forensic investigation model for a smart home with event reconstruction be evaluated?

Objective four of the study sought to determine whether the developed model was successful. A goal-based evaluation approach was used, based on the following metrics: accuracy, auditability, completeness, and reliability. The goals were defined, and a simulated attack was used to test whether the model was fit for use.

5.4 Recommendations

First, IoT device manufacturers should embed forensic readiness into the design of smart home technologies by adopting standardized logging formats, secure storage mechanisms, and event reconstruction capabilities. These features should not be treated as optional add-ons but as core components of device architecture. Firmware security must also be prioritized through regular updates, secure boot mechanisms, and digitally signed software, as recent studies have shown to be effective in preventing malicious code injection and ensuring evidentiary reliability. By integrating these measures at the design stage, manufacturers can significantly reduce investigation time and enhance accountability in smart home environments.

Second, policymakers and regulators need to establish clear legal and technical frameworks that mandate forensic capabilities in IoT ecosystems. Current standards, such as ISO/IEC 27043:2015, provide a foundation, but they must be adapted to address the unique challenges of smart homes, including heterogeneity, data volatility, and privacy concerns. Governments should also define minimum requirements for log retention, interoperability, and evidentiary admissibility, while investing in capacity building for law enforcement agencies. Training investigators in IoT-specific forensic processes and event reconstruction methodologies will ensure that forensic evidence collected from smart homes can withstand legal scrutiny.

Third, researchers should expand the scope of forensic evaluation by involving multiple experts and larger-scale smart home environments. While the proof-of-concept model validated by a

single expert demonstrates feasibility, broader validation will strengthen confidence in its accuracy, completeness, and reliability. Future studies should also explore advanced techniques such as machine learning for anomaly detection, blockchain for tamper-proof log storage, and AI-driven correlation algorithms to enhance event reconstruction. These innovations will address scalability and automation challenges, transforming the model from a prototype into a comprehensive forensic framework applicable across diverse IoT domains.

Finally, collaboration between academia, industry, and government is essential to operationalize these recommendations. Manufacturers can work with universities to test forensic-ready devices in controlled environments, while policymakers can incentivize compliance through certification schemes or regulatory mandates. Researchers, in turn, can provide evidence-based insights to refine standards and guide implementation. By creating a collaborative ecosystem, stakeholders can ensure that smart home devices are not only secure but also capable of producing reliable forensic evidence, thereby strengthening digital investigations and protecting users in an increasingly interconnected world.

REFERENCES

Ahmed, A. A., Farhan, K., Jabbar, W. A., Al-Othmani, A., & Abdulrahman, A. G. (2024). IoT Forensics: Current Perspectives and Future Directions. *Sensors*, *24*(16). https://doi.org/10.3390/s24165210

Ahmed, S. F., Shawon, S. S., Bhuyian, A., Afrin, S., Mehjabin, A., Kuldeep, S. A., Alam, M. S. Bin, & Gandomi, A. H. (2025). Forensics and security issues in the Internet of Things. *Wireless Networks*, *31*(4), 3431–3466. https://doi.org/10.1007/s11276-025-03942-2

Al-Sadi, M. B., Chen, L., & Haddad, R. J. (2018). Internet of Things Digital Forensic Investigation Using Open-Source Gears. *Conference Proceedings - IEEE SOUTHEASTCON*, *2018-April*. https://doi.org/10.1109/SECON.2018.8479042

Al-Shargabi, B., & Sabri, O. (2018). Internet of Things: An exploration study of opportunities and challenges. *Proceedings - 2017 International Conference on Engineering and MIS, ICEMIS 2017*, *2018-Janua*, 1–4. https://doi.org/10.1109/ICEMIS.2017.8273047

Andress, J., & Leary, M. (2015). *Building a Practical Information Security Program* (1st ed.). Syngress.

Anthi, E., Williams, L., & Burnap, P. (2018). Pulse: An adaptive intrusion detection for the Internet of Things. *IET Conference Publications*, *2018*(CP740), 1–4. https://doi.org/10.1049/cp.2018.0035

Årnes, A. (2018). *Digital Forensics* (1st ed.). Wiley, Hoboken, NJ, 2018.

Ayers, D. (2009). A Second-Generation Computer Forensic Analysis System. *Digital Investigation*, *6*(SUPPL.), S34–S42. https://doi.org/10.1016/j.diin.2009.06.013

Babun, L., Sikder, A. K., Acar, A., & Uluagac, A. S. (2018). *IoTDots: A Digital Forensics Framework for Smart Environments*.

Baho, S. A., & Abawajy, J. (2023). Analysis of Consumer IoT Device Vulnerability Quantification Frameworks. *Electronics (Switzerland)*, *12*(5). https://doi.org/10.3390/electronics12051176

Garcia Avila, D. R., Miller, J. F., & Iyengar, S. S. (2024). Current Challenges in IoT Security and Forensics: Strategies for a Secure Connected Future. In M. Rath & T. Samal (Eds.), *Key*

Issues in Network Protocols and Security. Intech Open. https://doi.org/10.5772/intechopen.1007766

Carrier, B. D., & Spafford, E. H. (2004). Defining Event Reconstruction of Digital Crime Scenes. *Journal of Forensic Sciences*, *49*(6), 1–8. https://doi.org/10.1520/jfs2004127

Cybersecurity Ventures. (2022). *Official Cybercrime Report, 2022*.

Flandrin, F., Buchanan, W. J., Macfarlane, R., Ramsay, B., & Smales, A. (2014). Evaluating Digital Forensic Tools (DFTs). | Semantic Scholar. *... Education &Training*, *September*, 1–16. https://doi.org/10.13140/2.1.3293.6004

Gladyshev, P. (2004). *Formalising event reconstruction in digital investigations*. *August* 212.

Goudbeek, A., Choo, K. K. R., & Le-Khac, N. A. (2018). A Forensic Investigation Framework for Smart Home Environment. *Proceedings - 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering, Trustcom/BigDataSE 2018*, 1446–1451. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00201

Gupta, K., & Shukla, S. (2016). Internet of Things: Security challenges for next-generation networks. *2016 1st International Conference on Innovation and Challenges in Cyber Security, ICICCS 2016*, *Iciccs*, 315–318. https://doi.org/10.1109/ICICCS.2016.7542301

Gupta, R., & Gupta, R. (2016). ABC of Internet of Things: Advancements, benefits, challenges, enablers, and facilities of IoT. *2016 Symposium on Colossal Data Analysis and Networking, CDAN 2016*, 1–5. https://doi.org/10.1109/CDAN.2016.7570875

Ieong, R. S. C. (2006). FORZA - A Digital forensics investigation framework that incorporates legal issues. *Digital Investigation*, *3*(SUPPL.), 29–36. https://doi.org/10.1016/j.diin.2006.06.004

ISO/IEC 27043:2015. (2015). *Information Technology — Security Techniques — Investigation principles and processes*. International Standard Organization.

Jeyaraman, S., & Atallah, M. J. (2006). An empirical study of automatic event reconstruction systems. *Digital Investigation*, *3*(SUPPL.), 108–115. https://doi.org/10.1016/j.diin.2006.06.013

Kaul, L., & Goudar, R. H. (2017). Internet of Things and Big Data - Challenges. *Proceedings of 2016 Online International Conference on Green Engineering and Technologies, IC-GET 2016*. https://doi.org/10.1109/GET.2016.7916854

Kebande, V. (2018). *A Novel Cloud Forensic Readiness Service Model*. University of Pretoria.

Kebande, V. R., Karie, N. M., Michael, A., Malapane, S. M. G., & Venter, H. S. (2017). How an IoT-enabled "smart refrigerator" can play a clandestine role in perpetuating cybercrime. *2017 IST-Africa Week Conference, IST-Africa 2017*, 1–10. https://doi.org/10.23919/ISTAFRICA.2017.8102362

Kebande, V. R., & Ray, I. (2016). A generic digital forensic investigation framework for Internet of Things (IoT). *Proceedings - 2016 IEEE 4th International Conference on Future Internet of Things and Cloud, FiCloud 2016*, 356–362. https://doi.org/10.1109/FiCloud.2016.57

Kebande, V. R., & Venter, H. S. (2015). Adding event reconstruction to a Cloud Forensic Readiness model. *2015 Information Security for South Africa - Proceedings of the ISSA 2015 Conference*, 1–9. https://doi.org/10.1109/ISSA.2015.7335050

Koley, S., & Ghosal, P. (2016). Addressing hardware security challenges in the Internet of Things: Recent trends and possible solutions. *Proceedings - 2015 IEEE 12th International Conference on Ubiquitous Intelligence and Computing, 2015 IEEE 12th International Conference on Advanced and Trusted Computing, 2015 IEEE 15th International Conference on Scalable Computing and Communications, 20*, 517–520. https://doi.org/10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.105

Kothari, C. R., & Gaurav, G. (2019). *Research Methodology: Methods and Techniques* (4th ed.). New Age International Publishers.

Leverage. (2018). *An Introduction to the Internet of Things* (1st ed.). Leverage.

Mahmood, H., Arshad, M., Ahmed, I., Fatima, S., & ur Rehman, H. (2024). Comparative study of IoT forensic frameworks. *Forensic Science International: Digital Investigation*, *49*, 301748. https://doi.org/10.1016/J.FSIDI.2024.301748

Michael Donovan. (2012). *Integrated Digital Forensic Process Model*. University of Pretoria.

Mohammed, H., Koroniotis, N., & Moustafa, N. (2023). *Digital Forensics based on Federated Learning in IoT Environment*. https://doi.org/10.1145/3579375.3579387

Nzabahimana, J. P. (2018). Analysis of security and privacy challenges in the Internet of Things. *Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies, DESSERT 2018*, 175–178. https://doi.org/10.1109/DESSERT.2018.8409122

Oriwoh, E., Jazani, D., Epiphaniou, G., & Sant, P. (2013). Internet of Things Forensics: Challenges and approaches. *Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, COLLABORATECOM 2013*, 608–615. https://doi.org/10.4108/icst.collaboratecom.2013.254159

Palmer, G. (2001). *A Road Map for Digital Forensic Research*. https://doi.org/10.1016/0032-3950(82)90064-8

Park, Y. S., Konge, L., & Artino, A. R. (2020). The Positivism Paradigm of Research. *Academic Medicine*, *95*, 690–694. https://doi.org/10.1097/ACM.0000000000003093

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V., & Bragge, J. (2006). The Design Science Research Process : A Model for Producing and Presenting Information Systems Research. *First International Conference on Design Science Research in Information Systems and Technology DESRIST*, 83–106.

Perumal, S., Md Norwawi, N., & Raman, V. (2015). Internet of Things(IoT) digital forensic investigation model: Top-down forensic approach methodology. *2015 5th International Conference on Digital Information Processing and Communications, ICDIPC 2015*, 19–23. https://doi.org/10.1109/ICDIPC.2015.7323000

Pressman, R., & Maxim, B. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.

Reddy, R. R., Mamatha, C., & Reddy, G. R. (2018). A Review on Machine Learning: Trends and Future. *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, *November 2017*, 2389–2397.

Rekha, G., & Sudha, T. (2025). Digital forensics in the IoT paradigm: Blockchain-assisted digital forensic framework with an improved cryptosystem. *Intelligent Decision Technologies*, 18724981241305870. https://doi.org/10.1177/18724981241305872

Rudrakar, S., Rughani, P., & Sadineni, L. (2025). Digital forensics and incident response management model for IoT-based agriculture. *Scientific Reports*, *15*(1). https://doi.org/10.1038/S41598-025-02635-2

Ryan, G. (2018). Introduction to positivism, interpretivism, and critical theory. *Nurse Researcher*, *25*(4), 14–20. https://doi.org/10.7748/nr.2018.e1466

Sathwara, S., Dutta, N., & Pricop, E. (2019). IoT Forensic: A digital investigation framework for IoT systems. *Proceedings of the 10th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2018*, *June* 1–4. https://doi.org/10.1109/ECAI.2018.8679017

Silva, T., OliveiraJr, E., Pereira, M., & Zorzo, A. (2025). A review study of digital forensics in IoT: Process models, phases, architectures, and ontologies. *Forensic Science International: Digital Investigation*, *53*, 301912. https://doi.org/10.1016/j.fsidi.2025.301912

Singh, S., & Singh, N. (2016). Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce. *Proceedings of the 2015 International Conference on Green Computing and Internet of Things, ICGCIoT 2015*, 1577–1581. https://doi.org/10.1109/ICGCIoT.2015.7380718

Soltani, S., & Seno, S. A. H. (2019). A formal model for event reconstruction in digital forensic investigation. *Digital Investigation*, *30*, 148–160. https://doi.org/10.1016/j.diin.2019.07.006

Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.

Tan, J. (2001). *Forensic Readiness*. *July* 1–12.

Tongay, K. N. (2016). Sensor data computing as a service in the Internet of Things. *2016 Symposium on Colossal Data Analysis and Networking, CDAN 2016*, 1–4. https://doi.org/10.1109/CDAN.2016.7570963

Valjarevic, A., Venter, H., & Petrovic, R. (2017). ISO/IEC 27043:2015 - Role and application. *24th Telecommunications Forum, TELFOR 2016*. https://doi.org/10.1109/TELFOR.2016.7818718

Valjarevic, A., & Venter, H. S. (2015). A Comprehensive and Harmonized Digital Forensic Investigation Process Model. *Journal of Forensic Sciences*, *60*(6), 1467–1483. https://doi.org/10.1111/1556-4029.12823

Whitman, M. E., & Mattord, H. J. (2021). *Principles of Information Security, Loose-leaf Version* (7th ed.). Cengage Learning.

Whittemore, R., Chao, A., Jang, M., Minges, K. E., & Park, C. (2014). Methods for knowledge synthesis: An overview. *Heart and Lung: Journal of Acute and Critical Care*, *43*(5), 453–461. https://doi.org/10.1016/j.hrtlng.2014.05.014

Yadav, E. P., Mittal, E. A., & Yadav, H. (2018). IoT: Challenges and Issues from an Indian Perspective. *Proceedings - 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages, IoT-SIU 2018*, 1–5. https://doi.org/10.1109/IoT-SIU.2018.8519869

Yeo, K. S., Chian, M. C., Wee Ng, T. C., & Tuan, D. A. (2015). Internet of things: Trends, challenges, and applications. *Proceedings of the 14th International Symposium on Integrated Circuits, ISIC 2014*, 568–571. https://doi.org/10.1109/ISICIR.2014.7029523

Zawoad, S., & Hasan, R. (2015). FAIoT: Towards Building a Forensics Aware Eco System for the Internet of Things. *Proceedings - 2015 IEEE International Conference on Services Computing, SCC 2015*, 279–284. https://doi.org/10.1109/SCC.2015.46

Zia, T., Liu, P., & Han, W. (2017). Application-specific digital forensics investigative model for the Internet of Things (IoT). *ACM International Conference Proceeding Series*, *Part F1305*(September). https://doi.org/10.1145/3098954.3104052

APPENDICES

Appendix I: Sample System Code

Listing 1:

```
"""
API endpoints for IoT Log Monitor demo visualization data.
This module provides Flask-RESTful API endpoints for the normal vs abnormal operations demo.
"""
from flask import Blueprint, jsonify, request
from flask_restful import Api, Resource
from datetime import datetime, timedelta
from bson.objectid import ObjectId
import pymongo
from integrity import verify_log_integrity, verify_log_collection_integrity

# Create a Blueprint for API routes
demo_bp = Blueprint('demo_api', __name__, url_prefix='/api/demo')
demo_api = Api(demo_bp)

# Store reference to MongoDB connection (will be set in app.py)
mongo_db = None

def init_demo_api(app, db):
    "Initialize the demo API with the Flask app and database connection."""
```

```
    global mongo_db
    mongo_db = db
    app.register_blueprint(demo_bp)
    return demo_api

class DemoLogsResource(Resource):
    "Resource for retrieving demo logs for normal vs abnormal operations."""

    def get(self):
        """
        Get log data for demonstration purposes.

        Query parameters:
        - source: Data source (normal, abnormal, tampered, compare)
        - hours: Number of hours to look back (default: 24)
        """
        # Parse query parameters
        source = request.args.get('source', 'normal')
        hours = int(request.args.get('hours', 24))

        # Calculate date range
        end_date = datetime.now()
        start_date = end_date - timedelta(hours=hours)
```

```
    # Build response data
    response_data = {}

    # Get normal logs if requested
    if source == 'normal' or source == 'compare':
        normal_logs = list(mongo_db.NormalLogs.find({
            "dateTime": {"$gte": start_date, "$lte": end_date}
        }).sort("dateTime", pymongo.ASCENDING)

        # Convert ObjectId to string for JSON serialization and ensure all values are JSON serializable
        for log in normal_logs:
            log["_id"] = str(log["_id"])
            # Convert datetime to ISO format string
            if "dateTime" in log and isinstance(log["dateTime"], datetime):
                log["dateTime"] = log["dateTime"].isoformat()
            # Ensure anomalyScore is a float
            If "anomalyScore" in log:
                log["anomalyScore"] = float(log["anomalyScore"])

        # Verify integrity of normal logs
        integrity_results = verify_log_collection_integrity(normal_logs)

        # Mark logs with integrity status
```

```
        for log in normal_logs:
            log["integrityVerified"] = log["_id"] not in [failed["id"] for failed in
integrity_results["failed_logs"]]

        response_data["normal"] = {
            "logs": normal_logs,
            "count": len(normal_logs)
        }

    # Get abnormal logs if requested
    if source == 'abnormal' or source == 'compare':
        abnormal_logs = list(mongo_db.AbnormalLogs.find({
            "dateTime": {"$gte": start_date, "$lte": end_date}
        }).sort("dateTime", pymongo.ASCENDING)

        # Convert ObjectId to string for JSON serialization and ensure all values are JSON
serializable
        for log in abnormal_logs:
            log["_id"] = str(log["_id"])
            # Convert datetime to ISO format string
            if "dateTime" in log and isinstance(log["dateTime"], datetime):
                log["dateTime"] = log["dateTime"].isoformat()
            # Ensure anomalyScore is a float
            If "anomalyScore" in log:
```

```
            log["anomalyScore"] = float(log["anomalyScore"])

        # Verify the integrity of abnormal logs
        integrity_results = verify_log_collection_integrity(abnormal_logs)

        # Mark logs with integrity status
        for log in abnormal_logs:
            log["integrityVerified"] = log["_id"] not in [failed["id"] for failed in integrity_results["failed_logs"]]

        response_data["abnormal"] = {
            "logs": abnormal_logs,
            "count": len(abnormal_logs)
        }

    # Get tampered logs if requested
    if source == 'tampered' or source == 'compare':
        tampered_logs = list(mongo_db.TamperedLogs.find({
            "dateTime": {"$gte": start_date, "$lte": end_date}
        }).sort("dateTime", pymongo.ASCENDING)

        # Convert ObjectId to string for JSON serialization and ensure all values are JSON serializable
        for log in tampered_logs:
```

```
            log["_id"] = str(log["_id"])
            # Convert datetime to ISO format string
            if "dateTime" in log and isinstance(log["dateTime"], datetime):
                log["dateTime"] = log["dateTime"].isoformat()
            # Ensure anomalyScore is a float
            If "anomalyScore" in log:
                log["anomalyScore"] = float(log["anomalyScore"])

        # Verify integrity of tampered logs
        integrity_results = verify_log_collection_integrity(tampered_logs)

        # Mark logs with integrity status
        for log in tampered_logs:
            log["integrityVerified"] = log["_id"] not in [failed["id"] for failed in
integrity_results["failed_logs"]]

        response_data["tampered"] = {
            "logs": tampered_logs,
            "count": len(tampered_logs)
        }

    # Calculate overall integrity statistics
    verified_count = 0
    failed_count = 0
```

```
        for source_key in response_data:
            for log in response_data[source_key]["logs"]:
                if log.get("integrityVerified", False):
                    verified_count += 1
                Else:
                    failed_count += 1

        response_data["integrity"] = {
            "verified": verified_count,
            "failed": failed_count,
            "total": verified_count + failed_count
        }

        return jsonify(response_data)

class IntegrityVerificationResource(Resource):
    "Resource for verifying the integrity of logs."""

    def get(self):
        """
        Verify the integrity of logs.

        Query parameters:
```

```
- source: Data source (normal, abnormal, tampered, all)
"""
# Parse query parameters
source = request.args.get('source', 'all')

# Initialize results
results = {
    "total": 0,
    "verified": 0,
    "failed": 0,
    "failed_logs": []
}

# Verify normal logs if requested
if source == 'normal' or source == 'all':
    normal_logs = list(mongo_db.NormalLogs.find())
    normal_results = verify_log_collection_integrity(normal_logs)

    results["total"] += normal_results["total"]
    results["verified"] += normal_results["verified"]
    results["failed"] += normal_results["failed"]
    results["failed_logs"].extend(normal_results["failed_logs"])

# Verify abnormal logs if requested
```

```
        if source == 'abnormal' or source == 'all':
            abnormal_logs = list(mongo_db.AbnormalLogs.find())
            abnormal_results = verify_log_collection_integrity(abnormal_logs)

            results["total"] += abnormal_results["total"]
            results["verified"] += abnormal_results["verified"]
            results["failed"] += abnormal_results["failed"]
            results["failed_logs"].extend(abnormal_results["failed_logs"])

        # Verify tampered logs if requested
        if source == 'tampered' or source == 'all':
            tampered_logs = list(mongo_db.TamperedLogs.find())
            tampered_results = verify_log_collection_integrity(tampered_logs)

            results["total"] += tampered_results["total"]
            results["verified"] += tampered_results["verified"]
            results["failed"] += tampered_results["failed"]
            results["failed_logs"].extend(tampered_results["failed_logs"])

        return jsonify(results)

# Register the API resources
demo_api.add_resource(DemoLogsResource, '/logs')
demo_api.add_resource(IntegrityVerificationResource, '/verify')
```

Listing 2: Integrity Check

```
"""
Log integrity module for IoT Log Monitor.
Provides functions for generating and verifying hash values to prevent log tampering
"""

import hashlib
import hmac
import os
from datetime import datetime

# Secret key for HMAC - in production, store this securely
# Generate a random secret key if not set
SECRET_KEY = os.environ.get('LOG_INTEGRITY_SECRET', os.urandom(32).hex())

def generate_log_hash(log_data):
    """
    Generate a SHA-256 hash for log integrity verification.

    Args:
        log_data (dict): Log data dictionary containing log fields

    Returns:
        str: Hexadecimal hash string
    """
    # Create a deterministic string representation of critical log fields
    # Include fields that shouldn't change if log integrity is maintained
    hash_input = f"{log_data.get('timestamp', '')}|" \
                 f"{log_data.get('logType', '')}|" \
                 f"{log_data.get('ipAddress', '')}|" \
                 f"{log_data.get('deviceId', '')}|" \
```

```
            f"{log_data.get('logMessage', '')}"

    # Generate HMAC using SHA-256 and the secret key
    , return hmac.new(
        SECRET_KEY.encode('utf-8'),
        hash_input.encode('utf-8'),
        hashlib.sha256
    ).hexdigest()

def verify_log_integrity(log_data, provided_hash):
    """
    Verify log data hasn't been tampered with by comparing hashes.

    Args:
        log_data (dict): Log data dictionary containing log fields
        provided_hash (str): Previously generated hash to verify against

    Returns:
        Bool: True if integrity is verified, False otherwise
    """
    # Generate a new hash from the current log data
    calculated_hash = generate_log_hash(log_data)

    # Use constant-time comparison to prevent timing attacks
    on HMAC.compare_digest(calculated_hash, provided_hash)

def verify_log_collection_integrity(logs):
    """
    Verify the integrity of a collection of logs.

    Args:
```

```
    logs (list): List of log dictionaries with integrity hashes

Returns:
    dict: Results with counts of verified and failed logs
"""
results = {
    'total': len(logs),
    'verified': 0,
    'failed': 0,
    'failed_logs': []
}

For login logs:
    # Skip logs without integrity hash
    If 'integrityHash' not in log:
        results['failed'] += 1
        results['failed_logs'].append({
            'id': str(log.get('_id', 'unknown')),
            'reason': 'No integrity hash found'
        })
        continue

    # Create a copy without the hash for verification
    log_copy = {k: v for k, v in log.items() if k != 'integrityHash'}

    if verify_log_integrity(log_copy, log['integrityHash']):
        results['verified'] += 1
    Else:
        results['failed'] += 1
        results['failed_logs'].append({
            'id': str(log.get('_id', 'unknown')),
```

```
            'reason': 'Integrity verification failed'
        })

    return results
```

Listing 3: Severity Generation

```
from datetime import datetime
import uuid
import re
from collections import Counter
import bcrypt
import math
from integrity import generate_log_hash



def calculate_log_severity(log_type, message):
    """
    Calculate the severity of a log entry based on its type and content.

    Severity levels:
    0 - Debug/Informational
    1 - Low severity
    2 - Medium severity
    3 - High severity
    4 - Critical severity
    5 - Emergency severity
    """
    # Base severity by log type
    base_severity = {
        'debug': 0,
        'info': 1,
        'warning': 2,
```

```
        'error': 3,
        'action': 1,
        'command': 1
    }.get(log_type.lower(), 1)

    # Increase severity based on keywords in the message
    critical_keywords = ['critical', 'emergency', 'failure', 'breach', 'attack', 'compromise', 'unauthorized']
    high_keywords = ['error', 'failed', 'denied', 'invalid', 'timeout', 'exception']
    medium_keywords = ['warning', 'retry', 'degraded', 'slow', 'missing']

    message_lower = message.lower()

    # Count occurrences of keywords
    critical_count = sum(1 for word in critical_keywords if word in message_lower)
    high_count = sum(1 for word in high_keywords if word in message_lower)
    medium_count = sum(1 for word in medium_keywords if word in message_lower)

    # Adjust severity based on keyword counts
    if critical_count > 0:
        base_severity = max(base_severity, 4)  # Critical
    elif high_count > 1:  # Multiple high severity keywords
        base_severity = max(base_severity, 3)  # High
    elif high_count > 0 or medium_count > 1:
        base_severity = max(base_severity, 2)  # Medium

    return min(base_severity, 5)  # Cap at maximum severity of 5

def calculate_anomaly_score(log_type, message, time_of_day):
    """
    Calculate an anomaly score for the log entry.
```

```
    This is a simple placeholder implementation.
    In production, this would use ML models or statistical analysis.

    Returns a score between 0 (normal) and 1 (highly anomalous).
    """
    # This is a placeholder implementation
    # In a real system, this would be based on learned patterns
    base_score = {
        'error': 0.7,
        'warning': 0.4,
        'info': 0.1,
        'debug': 0.0,
        'action': 0.3,
        'command': 0.3
    }.get(log_type.lower(), 0.1)

    # Adjust for time of day (placeholder logic)
    hour = time_of_day.hour
    if 0 <= hour < 5:  # Middle of night - unusual activity time
        time_factor = 0.3
    else:
        time_factor = 0.0

    # Final score calculation
    return min(base_score + time_factor, 1.0)

def extract_tags_from_message(message):
    """
    Extract potential tags from the log message.
    This helps with categorization and filtering of logs.
    """
```

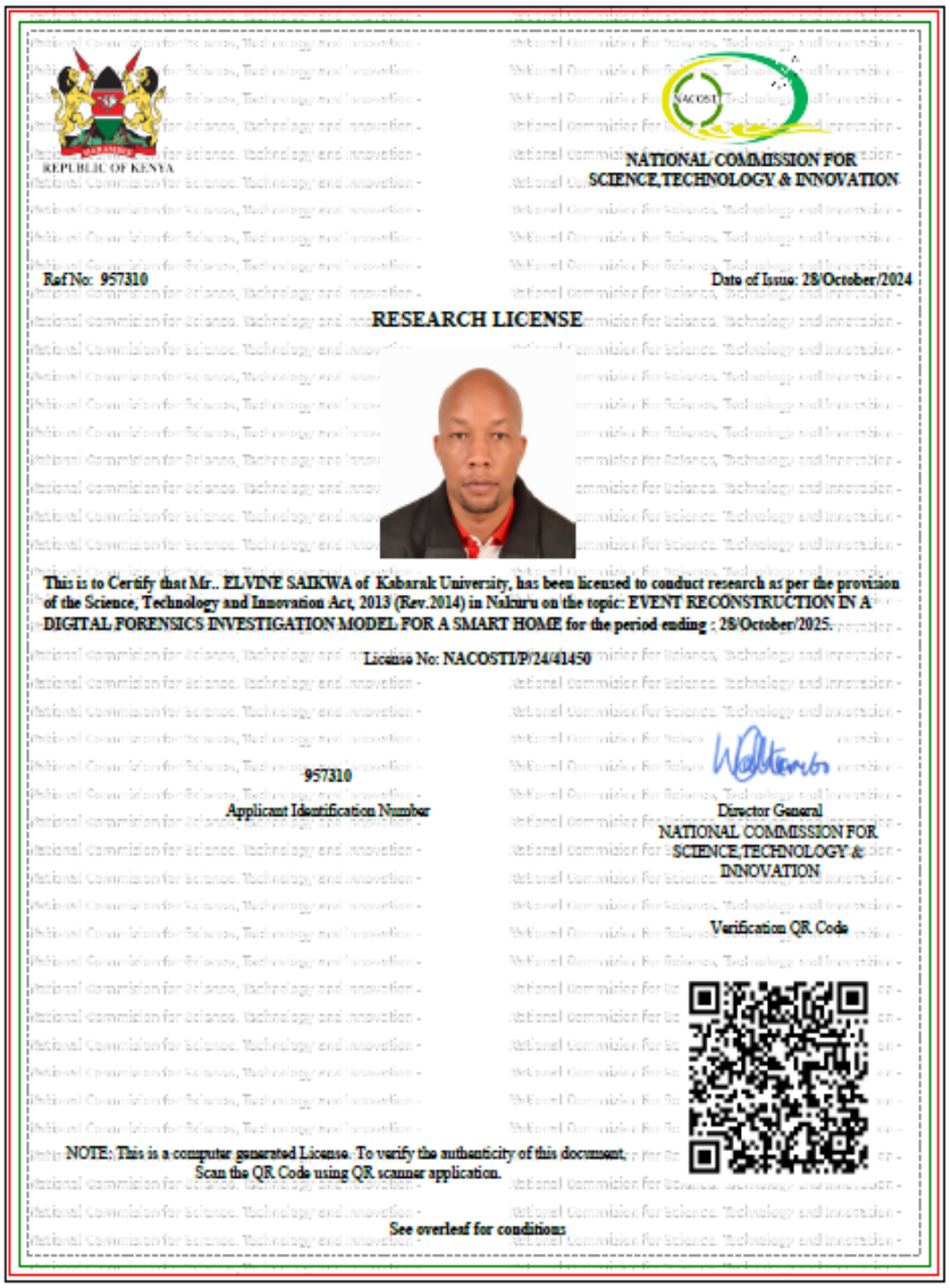
```
    # Common patterns for tags in log messages
    tag_patterns = [
        r'\[([^\]]+)\]',  # Matches content in square brackets
        r'\#([\w\-]+)',    # Matches hashtags
        r'tag:([\w\-]+)',  # Matches explicit tags
    ]

    tags = []
    for pattern in tag_patterns:
        matches = re.findall(pattern, message)
        tags.extend(matches)

    # Also extract potential keywords from the message
    words = re.findall(r'\b\w{3,}\b', message.lower())
    word_counts = Counter(words)

    # Important keywords that might be useful as tags
    important_keywords = ['login', 'logout', 'connect', 'disconnect', 'start', 'stop',
                          'error', 'warning', 'fail', 'success', 'denied', 'granted',
                          'update', 'restart', 'shutdown', 'boot', 'config', 'settings']

    For keyword in important_keywords:
        If keyword in word_counts:
            tags.append(keyword)

    # Return unique tags
    return list(set(tags))

def generate_correlation_id(device_id, log_type, message):
    """
    Generate a correlation ID for potentially related events.
```

```
    This is a simple implementation that tries to find patterns in log messages.

    In a production system, this would use more sophisticated techniques.
    """
    # Look for session IDs, request IDs, or transaction IDs in the message
    id_patterns = [
        r'session[_\s]?id[\s:=]([\w\-]+)',
        r'request[_\s]?id[\s:=]([\w\-]+)',
        r'transaction[_\s]?id[\s:=]([\w\-]+)',
        r'trace[_\s]?id[\s:=]([\w\-]+)',
    ]

    For pattern in id_patterns:
        match = re.search(pattern, message, re.IGNORECASE)
        If match:
            Return f"corr_{match.group(1)}"

    # If no pattern match, generate a unique ID
    # In real systems, related events would share IDs based on more complex logic
    return f"corr_{uuid.uuid4().hex[:10]}"

def process_log_data(log_string, topic=None, name=None):
    """
    Enhanced function to process log data with additional fields for event reconstruction.

    The log format:
    Timestamp | Log Type | IP Address | Log Message

    Example:
    1722240898 | INFO | 244.178.44.111 | Device is online.
```

```
Returns an enhanced log data dictionary with additional fields for event reconstruction.
"""
log_string = log_string.strip()
if log_string == '':
    return None

log_parts = log_string.split("|")
if len(log_parts) < 4:
    return None

device_id = topic.split("/")[0]
timestamp = float(log_parts[0])
datetime_obj = datetime.fromtimestamp(timestamp)
log_type = str(log_parts[1]).replace(" ", "").lower()
ip_address = str(log_parts[2]).replace(" ", "")
log_message = log_parts[3].strip()

# Basic log data
log_data = {
    "name": name,
    "timestamp": timestamp,
    "dateTime": datetime_obj,
    "logType": log_type,
    "ipAddress": ip_address,
    "logMessage": log_message,
    "topic": topic.replace(f"{device_id}/", ""),
    "deviceId": device_id,

    # Enhanced fields for event reconstruction
    "severity": calculate_log_severity(log_type, log_message),
    "correlationId": generate_correlation_id(device_id, log_type, log_message),
```

```
        "anomalyScore": calculate_anomaly_score(log_type, log_message, datetime_obj),
        "tags": extract_tags_from_message(log_message),
        "eventSequence": 0  # Will be updated when correlating events
    }

    # Generate integrity hash for tamper prevention
    log_data["integrityHash"] = generate_log_hash(log_data)

    return log_data



def check_password(hashed_password, user_password):
    # Check the password
    return bcrypt.checkpw(user_password.encode('utf-8'), hashed_password)
```

Appendix II: NACOSTI Permit



REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: 957310

Date of Issue: 28/October/2024

RESEARCH LICENSE

This is to Certify that Mr.. ELVINE SAIKWA of Kabarak University, has been licensed to conduct research as per the provision of the Science, Technology and Innovation Act, 2013 (Rev.2014) in Nakuru on the topic: EVENT RECONSTRUCTION IN A DIGITAL FORENSICS INVESTIGATION MODEL FOR A SMART HOME for the period ending : 28/October/2025.

License No: NACOSTI/P/24/41450

957310

Applicant Identification Number

Director General
NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Verification QR Code

NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.

See overleaf for conditions

Appendix III: Ethical Clearance

KABARAK UNIVERSITY RESEARCH ETHICS COMMITTEE

Private Bag - 20157
KABARAK, KENYA
Email: kurec@kabarak.ac.ke

Tel: 254-51-343234/5
Fax: 254-051-343529
www.kabarak.ac.ke

OUR REF: KABU01/KUREC/001/02/10/24

Date: 11th Oct, 2024

Satia Elvine Saikwa
Reg No.: GMIA/NE/0225/01/18
Kabarak University,

Dear Elvin,

RE: EVENT RECONSTRUCTION IN A DIGITAL FORENSICS INVESTIGATION MODEL FOR A SMART HOME.

This is to inform you that ***KUREC*** has reviewed and approved your above research proposal. Your application approval number is ***KUREC-021024.*** The approval period is 11***/10/2024 – 11/10/ 2025.***

This approval is subject to compliance with the following requirements:

i. All researchers shall obtain an introduction letter to NACOSTI from the relevant head of institutions (Institute of postgraduate, School dean or Directorate of research)
ii. The researcher shall further obtain a RESEARCH PERMIT from NACOSTI before commencement of data collection & submit a copy of the permit to **KUREC.**
iii. Only approved documents including (informed consents, study instruments, MTA Material Transfer Agreement) will be used
iv. All changes including (amendments, deviations, and violations) are submitted for review and approval by ***KUREC***;
v. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to ***KUREC*** within 72 hours of notification;
vi. Any changes, anticipated or otherwise that may increase the risk(s) or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to ***KUREC*** within 72 hours;
vii. Clearance for export of biological specimens must be obtained from relevant institutions and submit a copy of the permit to KUREC;
viii. Submission of a request for renewal of approval at least 60 days prior to expiry of the approval period. Attach a comprehensive progress report to support the renewal and;
ix. Submission of an executive summary report within 90 days upon completion of the study to ***KUREC***

Sincerely,

Prof. Jackson Kiteki PhD.
KUREC-Chairman

Cc Vice Chancellor
DVC-Academic & Research
Registrar-Academic & Research
Director-Research Innovation & Outreach
Institute of Post Graduate Studies

KABARAK UNIVERSITY
INSTITUTIONAL RESEARCH ETHICS COMMITTEE
APPROVED
P. O. PRIVATE BAG - 20157

As members of Kabarak University family, we purpose at all times and in all places, to set apart in one's heart, Jesus as Lord.
(1 Peter 3:15)
Kabarak University is ISO 9001:2015 Certified

Appendix IV: Publication

International Journal of Computer Applications Technology and Research
Volume 14–Issue 11, 35 – 39, 2025, ISSN:-2319–8656
DOI:10.7753/IJCATR1411.1004

An Investigation of Existing Digital Forensic Models for Internet of Things (IoT) Environments

Elvine Saikwa Satia
Student
Kabarak University
Nakuru, Kenya

Prof. Simon Karume
Lecturer
Kabarak University
Nakuru, Kenya

Dr. Nelson Masese
Lecturer
Kabarak University
Nakuru, Kenya

Abstract: The rapid proliferation of Internet of Things (IoT) devices in smart homes has created new challenges for digital forensic investigations. Traditional forensic models, designed for personal computers and mobile devices, are inadequate for heterogeneous IoT ecosystems characterized by distributed architectures, proprietary protocols, and volatile data. This paper investigates existing digital forensic models for IoT devices, focusing on their applicability to smart home environments. A systematic review of frameworks such as Oriwoh's 1-2-3 Zone model, Perumal's Top-Down approach, Zawoad and Hasan's Forensic-Aware IoT, and Kebande and Ray's DFIF-IoT, Zia et al.'s application-specific model, Goudbeek et al.'s smart home framework, Sathwara et al.'s three-step model, Al-Sadi et al.'s open-source geared approach, and Babun et al.'s IoTDots reveals that most remain conceptual, lack real-world validation, and struggle with scalability, interoperability, and evidentiary admissibility. Comparative analysis highlights deficiencies in event reconstruction, chain of custody, and automated correlation. The study identifies research gaps and proposes opportunities for integrating AI, blockchain, and standardized protocols to strengthen IoT forensic investigations. Findings contribute to the foundation for event reconstruction in smart home forensic models.

Keywords: Digital Forensics, IoT, Smart Home, Forensic Models, Event Reconstruction, Cybersecurity

1.0 INTRODUCTION

Digital forensics (DF) underpins the acquisition, preservation, analysis, and presentation of digital evidence so that it is admissible in judicial processes. The scope and rigor of DF differ from conventional investigations by emphasizing standardized procedures that reduce error and preserve probative value [1], [2]. The IoT paradigm broadened DF's landscape: smart homes interconnect sensors, actuators, and appliances that communicate via heterogeneous protocols and produce short-lived, distributed, and often proprietary data flows [3]. These characteristics complicate traditional forensic workflows designed for stand-alone computers or mobile devices.

IoT devices frequently ship with insufficient security configuration and patching discipline [4], increasing

existing models and processes. Section III provides the research methodology. Section IV provides results of the research. Section V concludes the study and section VI gives future recommendations.

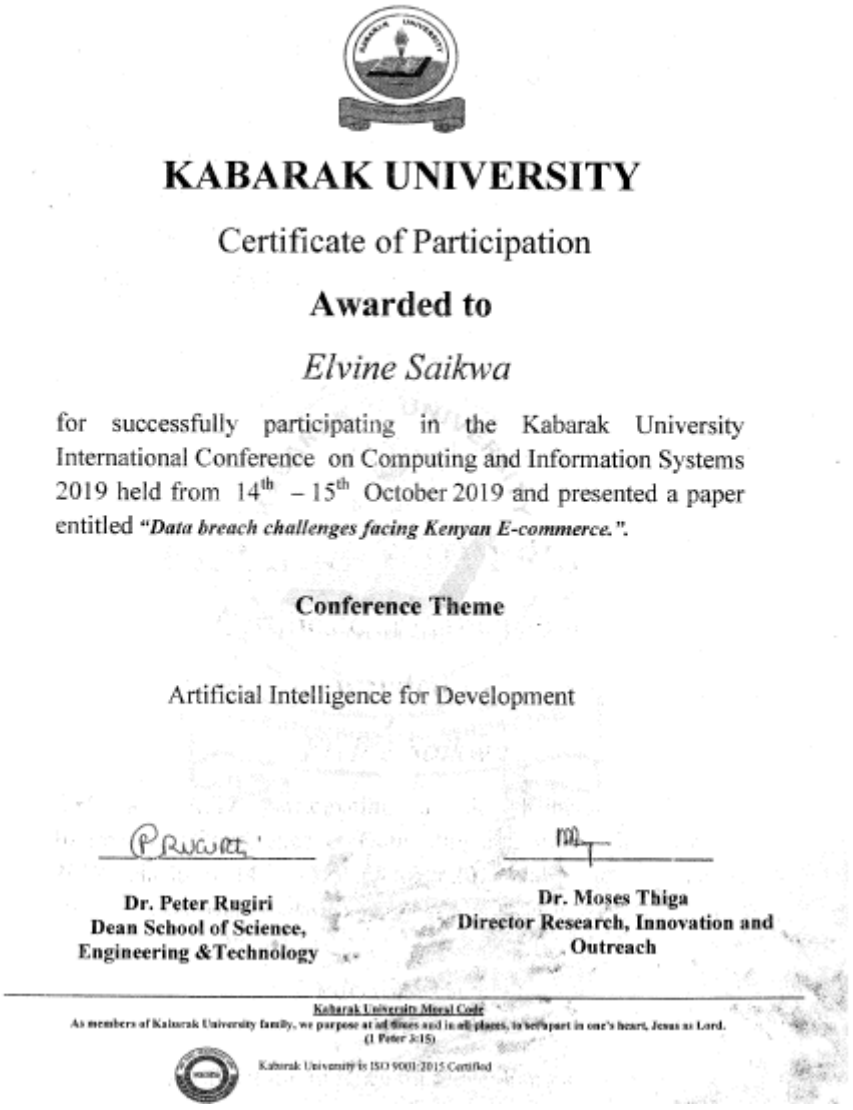
2.0 LITERATURE REVIEW

2.1 Background of Digital Forensics in IoT

IoT forensics extends DF principles—identification, preservation, acquisition, examination, analysis, and reporting—across distributed, multi-stakeholder environments. IoT ecosystems combine device, network, gateway, and cloud tiers, each emitting potential evidence with differing retention, accessibility, and jurisdictional constraints [4]. Evidence volatility, data heterogeneity, resource constraints (power, compute, storage), and proprietary stacks degrade real-time capture and post-hoc reconstruction fidelity.

Event reconstruction is central to DF: it reduces reasoning

Appendix V: Conference



KABARAK UNIVERSITY

Certificate of Participation

Awarded to

Elvine Saikwa

for successfully participating in the Kabarak University International Conference on Computing and Information Systems 2019 held from 14th – 15th October 2019 and presented a paper entitled ***"Data breach challenges facing Kenyan E-commerce."***.

Conference Theme

Artificial Intelligence for Development

Dr. Peter Rugiri
Dean School of Science, Engineering &Technology

Dr. Moses Thiga
Director Research, Innovation and Outreach

Kabarak University Moral Code
As members of Kabarak University family, we purpose at all times and in all places, to set apart in one's heart, Jesus as Lord. (1 Peter 3:15)

Kabarak University is ISO 9001:2015 Certified