INTRUSION DETECTION FRAMEWORK

A MAJOR PROJECT REPORT SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF DEGREE OF

BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

Aadhaar Koul(2020A1R040)

UNDER THE SUPERVISION OF
Mr. Saurabh Sharma
Assistant Professor, CSE Department

SUBMITTED TO

Department of Computer Science & Engineering
Model Institute of Engineering and Technology (Autonomous)
Jammu, India
2024

CANDIDATES DECLARATION

I hereby declare that the work which is being presented in the major project report entitled, **"INTRUSION DETECTION FRAMEWORK"** in the partial fulfillment of requirement for the award of degree of B.Tech. (CSE) and submitted to the Department of Computer Science and Engineering, Model Institute of Engineering and Technology (Autonomous), Jammu, is an authentic record of our own work carried byus under the supervision of **Mr. Saurabh Sharma, Asst. Professor, CSE Department**. The matter presented in this report has not been submitted to any other University / Institute for the award of B.Tech. Degree.

Signature of the Student ***Dated:***

Aadhaar Koul(2020a1R040)

Department of Computer Science & Engineering
Model Institute of Engineering and Technology (Autonomous)
Kot Bhalwal, Jammu, India
(NAAC "A" Grade Accredited)

Ref. No.: MIET/CSE/2024/P01 **Date:**

CERTIFICATE

Certified that this major project report entitled **"INTRUSION DETECTION FRAMEWORK"** is the bonafide work of **"Aadhaar Koul (2020a1r040) , 8th Semester, Computer Science Engineering, Model Institute of Engineering and Technology (Autonomous), Jammu",** who carried out the major project work under my / our supervision during February 2024 - June 2024.

Mr. Saurabh Sharma
Asst. Professor, CSE Department

This is to certify that the above statement is correct to the best of my knowledge.

Mr. Navin Mani Upadhayay
HoD, CSE Department
Model Institute of Engineering & Technology (Autonomous)

ACKNOWLEDGEMENT

This Major Project opportunity was a great chance for learning and professional development. I would also like to express my deepest gratitude to Mr. Saurabh Sharma, Asst. Prof., CSE Department, for hisprecious guidance and knowledge which were extremely valuable for our study both theoretically and practically.

I must record our deep sense of gratitude to Prof. (Dr.) Ankur Gupta (Director, MIET), Prof. (Dr.) Ashok Kumar (Dean Academics, MIET), Prof. Devanand Padha (Senior Professor, CSE Department), Mr. Navin Mani Upadhyay (HoD, CSE Department) for their guidance, constant inspiration, encouragement, and for their keen involvement throughout the course of present work.

I express my sincere gratitude to Model Institute of Engineering and Technology (Autonomous), Jammu for giving me the opportunity to work on the Major Project during our final year of B.Tech.

I would also like to thank my parents who helped us in completion of this Major Project. At the end, thanks to the Almighty for making us fortunate enough to be surrounded by helping and knowledgeable people.

Aadhaar Koul (2020A1R040)

ABSTRACT

The project is an intrusion detection system that leverages the power of honeypot systems, computer vision and various modules integrated into a single framework the system combines different modules including a network gateway honeypot module IoT module blink cloud module computer vision and Android application module these modules work together seamlessly with an Android application acting as the interface to access all the functionality.Whenever any suspicious activity is detected on any of the modules the notification system promptly alerts the administrator who can then take appropriate actions and the notification system sends push notifications to the administrators device prompting them to pay attention to the intrusion and investigate further. The ideas project is based on honeypot systems and computer vision Which are a comprehensive solution for detecting and preventing security breaches in a networked environment the integration of various modules and the Android application interface makes the system easy to use and manage even for non-technical users with its advanced features and notification system the ideas project is an excellent tool for securing networks against potential cyber attackers.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ESP	Electronic Stability Control
HG	Huntington diseases
IOT	Internet of things
IDE	Integrated Development Environment
LCD	Liquid Crystal Display
LED	Light Emitting Diode
PDA	Proportional derivative acceleration
IDS	Intrusion Detection System
GPIO	General Pourpose Input Output
SDK	Software Development Kit
SSL	Secure Sockets Layer

CHAPTER 1

INTRODUCTON

A system called an intrusion detection system (IDS) observes network traffic for malicious transactions and sends immediate alerts when it is observed. It is software that checks a network or system for malicious activities or policy violations. Each illegal activity or violation is often recorded either centrally using a SIEM system or notified to an administration. IDS monitors a network or system for malicious activity and protects a computer network from unauthorized access from users, including perhaps insiders. The intrusion detector learning task is to build a predictive model (i.e. a classifier) capable of distinguishing between 'bad connections' (intrusion/attacks) and 'good (normal) connections'.[2]

1.1 How Does an IDS Work

An IDS (Intrusion Detection System) functions as a vigilant monitor of computer network traffic, diligently analyzing data patterns and anomalies to detect any potentially suspicious activity. It systematically compares the ongoing network activity against a meticulously crafted set of predefined rules and behavioral patterns, designed to identify signs of unauthorized access or malicious intent. Upon detecting a match with these parameters, the IDS promptly triggers an alert, notifying the system administrator of the potential security breach. This swift notification enables the administrator to swiftly investigate the alert, ascertain the nature and severity of the threat, and take decisive action to prevent any

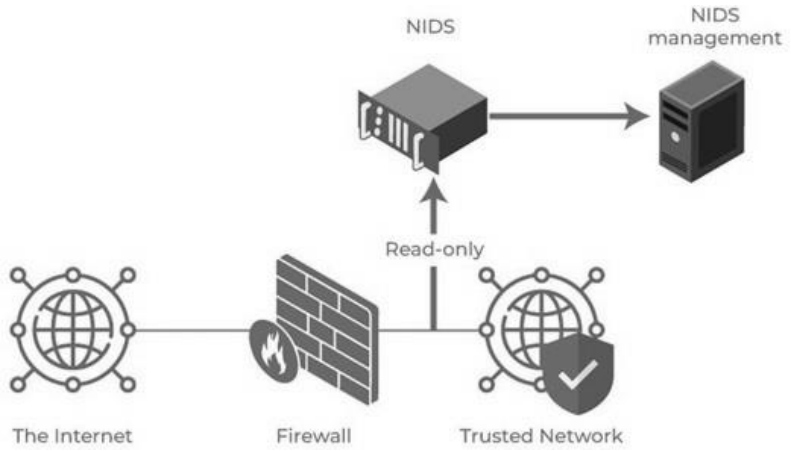

Fig 1.1 : Intrusion Detection Architecture [15]

further damage or compromise to the network infrastructure. The responsiveness of an IDS is crucial in maintaining network security, with reaction times typically measured in the span of tens of seconds, ensuring that any potential threats are swiftly addressed to safeguard the integrity and confidentiality of the network data.

1.2 Classification of Intrusion Detection System

IDS are classified into 5 types:

Host Intrusion Detection System (HIDS): Host intrusion detection systems (HIDS) run on independent hosts or devices on the network. A HIDS monitors the incoming and outgoing packets from the device only and will alert the administrator if suspicious or malicious activity is detected. It takes a snapshot of existing system files and compares it with the previous snapshot. If the analytical system files were edited or deleted, an alert is sent to the administrator to investigate. An example of HIDS usage can be seen on mission-critical machines, which are not expected to change their layout.[14]

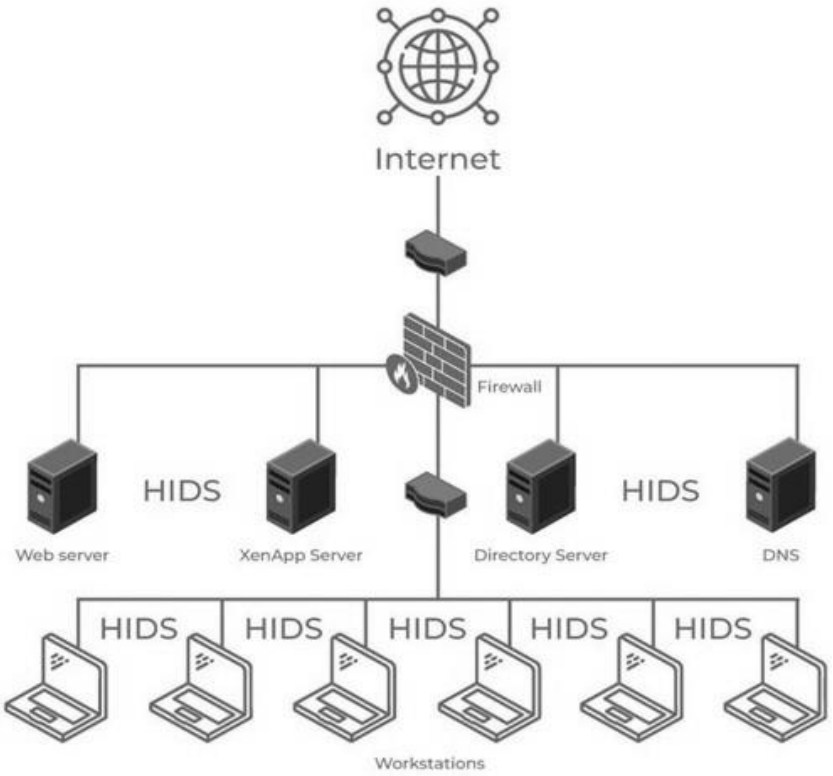

Fig 1.2 : Intrusion Detection Classification [16]

Protocol-based intrusion detection system (PIDS) comprises a system or agent that would consistently reside at the front end of a server, controlling and interpreting the protocol between a user/device and the server. It is trying to secure the web server by regularly monitoring the HTTPS protocol stream and accepting the related HTTP protocol. As HTTPS is unencrypted and before instantly entering its web presentation layer then this system would need to reside in this interface, between to use the HTTPS.[4].

1.3 What is a Honeypot System

Honeypot is a network-attached system used as a trap for cyber-attackers to detect and study the tricks and types of attacks used by hackers. It acts as a potential target on the internet and informs the defenders about any unauthorized attempt to the information system. Honeypots are mostly used by large companies and organizations involved in cybersecurity. It helps cybersecurity researchers to learn about the different type of attacks used by attackers. It is suspected that even the cybercriminals use these honeypots to decoy researchers and spread wrong information.The cost of a honeypot is generally high because it requires specialized skills and resources to implement a system such that it appears to provide an organization's resources still preventing attacks at the backend and access to any production system.A honeynet is a combination of two or more honeypots on a network.

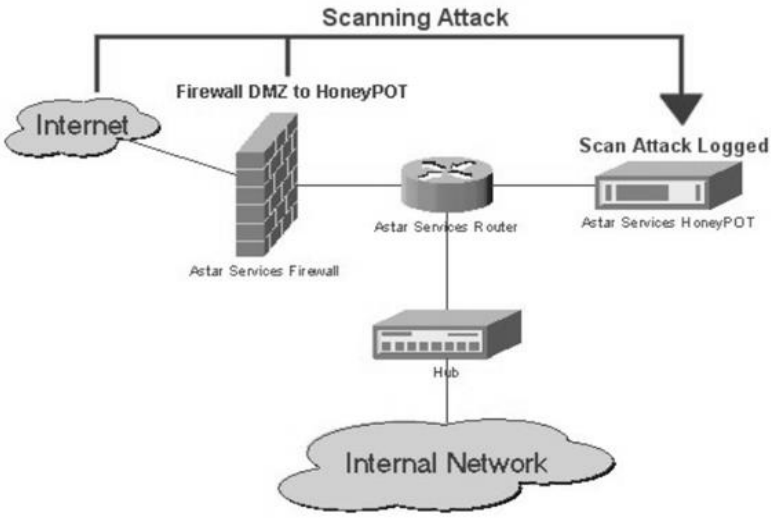

Fig 1.3 : Honeypot System Architecture

1.4 Types of Honeypots

Honeypots, critical components in the field of cybersecurity, are classified based on their deployment and the level of interaction with intruders. This classification helps in understanding their specific roles and optimizing their deployment for effective network security. Based on deployment, honeypots are divided into research and production honeypots. Research honeypots are designed primarily for academic and experimental purposes, where cybersecurity researchers use them to study hacker behaviors, analyze attack patterns, and develop new methods to prevent these attacks. They are instrumental in understanding the evolving tactics of cybercriminals and devising robust security measures. On the other hand, production honeypots are integrated into production networks alongside servers. These honeypots function as decoys, presenting false information to attackers and acting as a frontline defense mechanism. By misleading attackers, production honeypots buy valuable time for system administrators to identify and rectify vulnerabilities in the actual systems, thereby enhancing overall network security.

Furthermore, honeypots are categorized based on the level of interaction they provide to intruders, which includes low, medium, and high interaction honeypots. Low interaction honeypots simulate only the most commonly requested services by attackers, offering minimal interaction. This limited engagement ensures that the main operating system remains uninvolved, reducing the risk associated with potential breaches. These honeypots are relatively easy to deploy, require fewer resources, and provide basic insights into attack methods. However, their simplicity also means that experienced hackers can often identify and bypass them, limiting their effectiveness in high-stakes environments. Medium interaction honeypots offer a higher level of engagement than low interaction honeypots, allowing hackers to perform more activities. These honeypots are designed to mimic real systems more closely and provide responses that go beyond the basic interactions of low interaction honeypots, thus yielding more detailed information about attack methods and strategies.High interaction honeypots represent the most sophisticated type of honeypot, offering a broad range of services and activities to hackers. These honeypots engage hackers extensively, simulating real operating systems and environments. The primary objective of high interaction honeypots is to waste hackers' time and gather comprehensive data on their behavior and techniques. Due to their high level of engagement, they are significantly riskier because they involve real-time operating systems. If a hacker identifies a high interaction honeypot, they could potentially exploit it to gain deeper insights into the system's defenses. Despite these risks, high

interaction honeypots are invaluable for in-depth cybersecurity research. They provide extensive information that can be used to enhance security protocols and develop advanced defensive strategies. However, their complexity and the need for substantial resources make them expensive to implement and maintain. Thus, while they offer rich insights and robust defense mechanisms, their deployment is often limited to environments where the benefits outweigh the associated costs and risks.

1.5 Advantages and Disadvantages of Honeypots

ADVANTAGES	DISADVANTAGES
Acts as a rich source of information and helps collect real-time data.	Being distinguishable from production systems, it can be easily identified by experienced attackers.
Identifies malicious activity even if encryption is used.	Having a narrow field of view, it can only identify direct attacks.
Improves security.	Fingerprinting (an attacker can identify the true identity of a honeypot).

Table 1.1 : Advantages and Disadvantages of a Honeypot

1.6 What is IOT ?

IoT stands for Internet of Things. It refers to the interconnectedness of physical devices, such as appliances and vehicles, that are embedded with software, sensors, and connectivity which enables these objects to connect and exchange data. This technology allows for the collection and sharing of data from a vast network of devices, creating opportunities for more efficient and automated systems.

Internet of Things (IoT) is the networking of physical objects that contain electronics embedded within their architecture in order to communicate and sense interactions amongst each other or with respect to the external environment. In the upcoming years, IoT-based technology will offer advanced levels of services and practically change the way people lead their daily lives. Advancements in medicine, power, gene therapies, agriculture, smart cities, and smart homes are

just a very few of the categorical examples where IoT is strongly established. [6]

IoT is network of interconnected computing devices which are embedded in everyday objects, enabling them to send and receive data.Over 9 billion ‘Things’ (physical objects) are currently connected to the Internet, as of now. In the near future, this number is expected to rise to a whopping 20 billion.

1.6.1 Main Components Used in IOT

Low-power embedded systems are a cornerstone of IoT technology, where the dual imperatives of low battery consumption and high performance drive the design of electronic systems. These systems must balance energy efficiency with robust functionality to enable long-term, reliable operation in various applications, from smart homes to industrial automation. An Intrusion Detection System (IDS) plays a crucial role in this landscape, acting as a vigilant monitor of computer network traffic. By continuously analyzing data patterns and anomalies, an IDS detects suspicious activities by comparing ongoing network activity against predefined rules and behavioral patterns aimed at identifying unauthorized access or malicious intent. When a match is detected, the IDS promptly alerts the system administrator, enabling swift investigation and decisive action to mitigate potential security breaches. This proactive approach ensures that threats are swiftly addressed, maintaining the integrity and confidentiality of the network data. The responsiveness of an IDS, with reaction times typically measured in seconds, underscores its essential role in upholding robust network security protocols, safeguarding the intricate web of interconnected devices that comprise IoT systems.Building IoT systems can be approached in various ways, each with its unique set of challenges and advantages. One method involves forming a separate internetwork that includes only physical objects, effectively creating a distinct and isolated network environment dedicated to IoT devices. This approach simplifies security and management but may limit the scalability and integration capabilities of the IoT system. Alternatively, expanding the Internet itself to accommodate IoT devices necessitates advanced technologies such as rigorous cloud computing and rapid big data storage. This method allows for greater scalability and more seamless integration with existing internet infrastructure but comes at a higher cost due to the need for robust data management and processing capabilities.

CHAPTER 2

PROBLEM STATEMENT AND JUSTIFICATION

2.1 Problem Statement and Justification

Small-scale industries, general public, and unaware users are particularly vulnerable to cyber attacks due to their limited resources, lack of awareness, and reliance on technology. Small-scale industries, in particular, often lack the cybersecurity infrastructure and resources of larger organizations, making them more susceptible to attacks. According to a recent survey, 43% of cyber attacks target small businesses.General public and unaware users are also at risk, as they may not be familiar with basic cybersecurity practices and may unknowingly expose themselves to cyber threats. This includes using weak passwords, clicking on suspicious links, and downloading malicious software. Furthermore, many people are unaware of the importance of regularly updating their software, antivirus, and firewalls. This leaves them vulnerable to attacks that exploit known vulnerabilities in outdated software and systems. The consequences of a cyber attack on small-scale industries, general public, and unaware users can be devastating, including financial losses, reputational damage, and theft of sensitive data. It is therefore essential that everyone takes proactive steps to protect themselves against cyber attacks, such as educating themselves on basic cybersecurity practices, using strong passwords, keeping their software up to date, and being vigilant against suspicious activity.

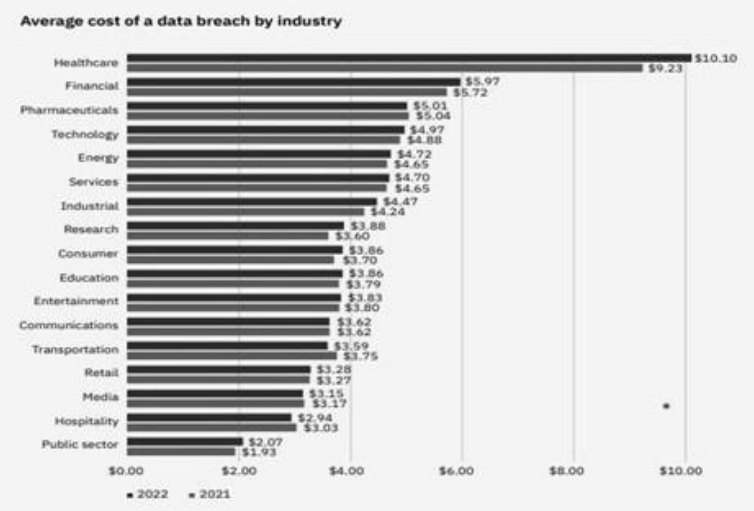

Fig 2.1 : Average data breach cost by industry [17]

According to a comprehensive report by Cybersecurity Ventures, the financial impact of cybercrime is expected to reach an astronomical $10.5 trillion annually by 2025, a steep climb from $3 trillion recorded in 2015. This staggering projection highlights the exponential growth of cyber threats and the significant economic burden they impose globally. The same report provides a sobering prediction regarding ransomware attacks, estimating that they will occur every 11 seconds by 2021, a troubling increase from every 14 seconds in 2019. The frequency and sophistication of these attacks are a clear indication of the evolving tactics employed by cybercriminals. In a related finding, the FBI reported a dramatic 400% surge in cybercrime complaints in 2020, driven largely by the COVID-19 pandemic. This surge resulted in losses exceeding $4.2 billion, underscoring the heightened vulnerability of digital infrastructures during periods of global crisis. Further emphasizing the escalation in cyber threats, the 2021 SonicWall Cyber Threat Report documented a 62% global increase in ransomware attacks in 2020. The report also noted that the average ransom demand had soared to $170,404, reflecting the growing audacity and financial motivation of cyber attackers. A detailed study by Verizon provided additional insights, revealing that 70% of security breaches in 2020 were perpetrated by external actors, with hacking being a factor in 52% of these incidents. This data underscores the external origins of most cyber threats and the persistent challenge of safeguarding against sophisticated hacking attempts. Furthermore, the 2021 Cost of a Data Breach Report by IBM Security and Ponemon Institute highlighted the financial repercussions of such incidents, indicating that the

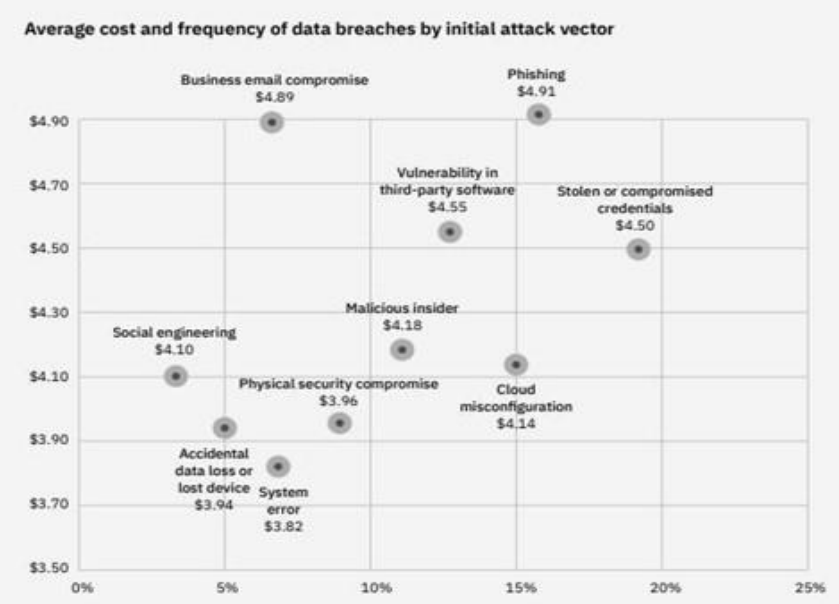

Fig 2.2 : Average cost and frequency of data breaches [17]

average cost of a data breach rose to $4.24 million in 2021, marking a 10% increase from the previous year. This upward trend in breach costs reflects the escalating complexity and severity of cyber incidents, as well as the substantial financial and reputational damage they inflict on affected organizations In the second figure we can see that the phishing attacks are most frequent types of attacks to which the naive users are the most susceptible to. There by the general public is most vulnerable to these types of attacks.

2.2 Solution

There needs to be a hybrid intruder detection system that works on both the network and premises level and handles all the security parameter using the state-of-the-art artificial intelligent model specifically designed for the maximum output ad results in the cyber world. A home automation system needs to be devised that would possess the IOT modules like the Laser sensors, Ultrasonic Sensors, PIR Sensors etc which would work in coordinated manner around a framework which tracks each and every sensor and the network activity and generates the flag based triggered notification for the user.The Solution need to be cheap and efficient that would simulate the real world IDS's. In order to achieve these cheap microcomputers and microcontroller need to be integrated into the framework such as the Raspberry pi 4 Model, Arduino pro micro, ESP 32, 8266 etc. These Components would work in a coordinated manner to generate flags whenever an IOT module triggers a signal.A live feed for the IOT reading need to be rendered to a cloud module which would give an ability to render the reading to an android app or a website using an integratable API key

Solution Objectives

- Cut down costs for maximum affordability.
- Make the product seamless and easy to use.
- Cover most of the security concerns to deliver the best product

Expected Results

- Deliver all the modules as planned
- Deliver a high quality and efficient product
- Ability to scale up as per the market requirement.
- The consumer should be totally satisfied with the ethics and the quality as promised.

2.3 Theory

Programming languages play a pivotal role in the implementation and functionality of various aspects of a project, each serving a specific purpose. JavaScript is the primary language used for the frontend, encompassing the user interface, interactivity, and dynamic rendering of network information. Its widespread support by web browsers enables seamless client-side scripting, allowing for responsive and engaging user experiences. HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are fundamental for structuring and styling web pages. HTML defines the page structure and content, providing a semantic framework for the information presented. In contrast, CSS focuses on the visual presentation, including layout, colors, and typography, ensuring a visually appealing and user-friendly interface. On the hardware side, Arduino, which uses a simplified version of C++, is employed for programming the ESP32 CAM module. This allows for low-level control of the camera module's functions, configuration, and interactions with other components, making it essential for precise hardware manipulation and integration. Python is utilized for the backend server implementation, handling tasks such as the Blynk cloud module and data transmission between the network gateway and the admin control application. Python's versatility and extensive libraries make it an ideal choice for server-side programming, facilitating robust and efficient backend operations. Java is used for developing the admin control application, providing an interface for managing and controlling the network gateway. Its object-oriented programming paradigm and rich ecosystem make it well-suited for building robust desktop applications that require reliable performance and maintainability. Additionally, XML (eXtensible Markup Language) is employed for data representation and configuration across various parts of the project. It provides a standardized format for storing and exchanging structured data, ensuring compatibility and ease of integration between different components. This combination of languages, each chosen for their specific strengths, creates a cohesive and efficient system capable of handling the diverse requirements of the project.

2.4 Computer Vision Module

YOLOv8 , an advanced version of the YOLO (You Only Look Once) Object detection model , introduces significant improvements over its predecessor ,, such as new backbone network , a refined loss function, and an anchor-free detection head. Developed by Ultralytics, YOLOv8 is designed for high-speed and accurate image segmentation tasks, making it a state-of-the-art tool in computer vision.

2.4.1 Image Segmentation with YOLOv8:

YOLOv8 supports instance segmentation, which differentiates individual instances of the same object class. Pre-trained models like "yolov8n-seg.pt" are trained on the COCO128-seg dataset, enabling detailed analysis of objects in images.

Training Details: The model is trained for 100 epochs on 640-pixel images, enhancing its ability to accurately segment objects.[9]

Advantages of YOLOv8:

- **Speed:** Capable of processing 81 frames per second, YOLOv8 is significantly faster than other models like Mask R-CNN, making it ideal for real-time applications such as self-driving cars and security systems. This high processing speed ensures that the system can respond quickly to dynamic environments, which is critical for applications where timing is essential.
- **Flexibility:** YOLOv8's unified framework handles multiple tasks efficiently, including object detection, instance segmentation, and image classification. This versatility reduces the need for deploying multiple models for different tasks, thereby saving computational resources and simplifying the development and deployment process. The ability to manage diverse tasks within a single framework makes YOLOv8 a powerful tool for developers working on complex, multi-faceted projects.
- **Pre-trained Models:** YOLOv8 offers pre-trained models for various tasks, which can save developers a significant amount of time and resources. These pre-trained models provide a strong starting point and can be fine-tuned for specific applications, allowing for rapid development and deployment. The availability of these models means that developers do not need to build and train models from scratch, thus speeding up the project timeline.
- **Practical Applications:** YOLOv8's speed and accuracy make it suitable for numerous real-time applications. Its ability to process images and video frames quickly and accurately is beneficial in fields such as autonomous driving, where the vehicle needs to recognize and react to its surroundings instantaneously. Additionally, YOLOv8 can be used in security systems to monitor and detect suspicious activities in real-time, enhancing safety and security measures. Its efficiency and versatility extend to various

other domains, making it a highly practical choice for developers looking to implement advanced image and video analysis capabilities in their projects.

2.4.2 YOLOv8 in Real-Time Intruder Detection:

The model processes live camera feeds to detect and classify individuals, comparing them to a database of known persons.Upon detecting an unknown individual, it triggers security measures like locking mechanisms and alarms, and sends real-time alerts to neighbors, enhancing overall security.

Fig 2.3 : Computer Vision Implementation

2.5 Surveillance Module

Implementing a full-fledged camera surveillance system to cover most of the premises can be an effective security measure, but it often comes with significant costs and maintenance requirements. This is where the $10 ESP-32 CAM module becomes a highly attractive alternative. The ESP32-CAM offers several advantages over traditional commercial surveillance cameras, making it a compelling choice for various applications. Firstly, its cost-effectiveness cannot be overstated. Compared to commercial surveillance cameras, the ESP32-CAM is a much more affordable option, making it accessible for individuals or organizations

operating on a limited budget. This affordability does not come at the expense of functionality, as the ESP32-CAM is a capable device with a range of features suitable for surveillance purposes.The ESP32-CAM is also compact and lightweight, which allows for discreet placement in locations where traditional cameras might be too bulky or conspicuous. This makes it ideal for a variety of surveillance scenarios, from monitoring small indoor spaces to keeping an eye on larger outdoor areas without drawing attention. Another significant advantage of the ESP32-CAM is its versatility. Unlike traditional surveillance cameras that come with fixed functionalities, the ESP32-CAM is a programmable device that can be customized and integrated into existing systems or projects. This provides users with a high degree of flexibility in terms of functionality and application, allowing the camera to be tailored to specific surveillance needs. Moreover, the ESP32-CAM's built-in Wi-Fi capabilities are a game-changer. It can easily connect to a network, stream video footage, and support real-time monitoring and remote access via mobile devices or web interfaces. This feature is particularly valuable for users who need to keep an eye on their premises from remote locations, providing peace of mind and enhanced security. The convenience of easy setup and configuration further adds to the appeal of the ESP32-CAM. Users with programming experience can quickly install and configure the device using the Arduino IDE, making the development process straightforward. The module's compatibility with various libraries and APIs allows for seamless integration with other systems, enhancing its utility and making it a versatile component in any surveillance setup.

In summary, the ESP32-CAM module offers a cost-effective, versatile, and easy-to-use alternative to traditional commercial surveillance cameras. Its affordability makes it accessible for budget-conscious individuals and organizations, while its compact size allows for discreet placement in a variety of settings. The module's programmability and built-in Wi-Fi capabilities provide flexibility and convenience, enabling real-time monitoring and remote access. The ease of setup and integration with other systems further enhance its value, making the ESP32-CAM an excellent choice for anyone looking to implement a comprehensive and efficient surveillance system without the hefty price tag and maintenance burden associated with traditional solutions. The pictorial representation of the ESP-32 Cam is depicted below, showcasing its compact design and highlighting its potential for a wide range of surveillance applications.

2.5.1 Features

The ESP32-CAM module The ESP32-CAM module boasts a variety of impressive features that make it a highly versatile and efficient option for numerous applications. Firstly, it is recognized as the smallest 802.11b/g/n Wi-Fi BT SoC module, which enhances its applicability in compact and discreet setups. Despite its small size, the module is equipped with a low-power 32-bit CPU that can also function as an application processor, delivering substantial computing power. With a clock speed of up to 160MHz and a computing capacity of up to 600 DMIPS, it ensures high performance in processing tasks. Additionally, the ESP32-CAM includes built-in 520 KB SRAM and an external 4MPSRAM, providing ample memory for various operations.

The module supports multiple communication interfaces such as UART, SPI, I2C, PWM, ADC, and DAC, making it highly adaptable to different hardware requirements. This wide range of interfaces ensures that the ESP32-CAM can be integrated into various projects, from simple data logging tasks to complex sensor networks and control systems. It is compatible with OV2640 and OV7670 cameras and comes with a built-in flash lamp, which is particularly useful for capturing images in low-light conditions. The ESP32-CAM also supports image WiFi upload, enabling seamless transmission of captured images over a network. This feature is particularly beneficial for applications such as remote monitoring, security systems, and IoT devices that require real-time image processing and transmission.

Moreover, the ESP32-CAM supports TF cards, which can be used for additional storage, allowing for extensive data logging and image storage capabilities. It offers multiple sleep modes, making it energy-efficient for prolonged use. This is especially important for battery-powered applications where power conservation is crucial. The module has embedded Lwip and FreeRTOS, supporting a variety of network protocols and real-time operations. It can operate in various modes, including STA, AP, and STA+AP, providing flexibility in network connectivity. The module also supports Smart Config and AirKiss technologies, facilitating easy network configuration. These features make it straightforward to set up and deploy the ESP32-CAM in different network environments, whether it's for home automation, industrial IoT, or remote sensor networks.

2.5.2 Circuit

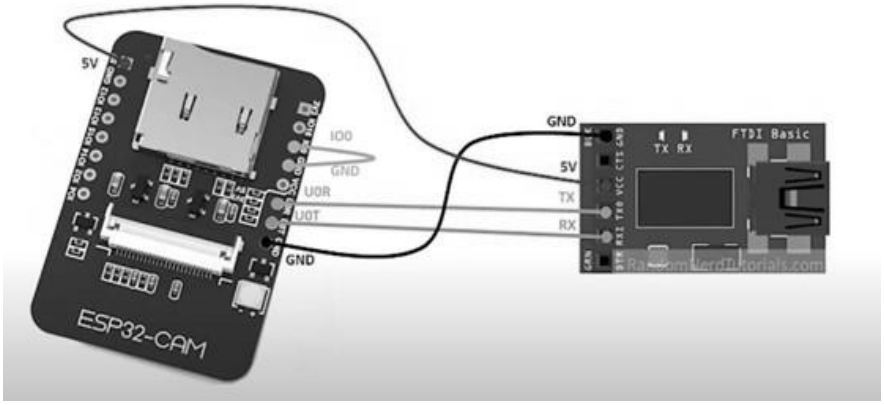

Fig 2.4 : ESP 32 cam and power module circuit diagram

2.5.3 Video Streaming Server

The code begins by selecting the camera model by uncommenting the corresponding #define statement. The available camera models include WROVER_KIT, ESP_EYE, M5STACK_PSRAM, M5STACK_V2_PSRAM, M5STACK_WIDE, M5STACK_ESP32CAM, AI_THINKER, and TTGO_T_JOURNAL.The SSID and password for the Wi-Fi network are defined as variables. These credentials will be used to connect the ESP32-CAM module to the desired network.[6] The setup() function is then defined. It starts the serial communication, configures debug output, and initializes the camera configuration structure, "config," with the appropriate pin mappings and settings. The "config" structure specifies various GPIO pins for different camera functions, such as data lines, clock signals, synchronization signals, and power-related pins. The pixel format is set to PIXFORMAT_JPEG, indicating that the captured images will be in JPEG format.If the PSRAM (pseudo-static random access memory) is detected, the configuration is adjusted to support higher resolution (UXGA) and lower JPEG quality with a larger frame buffer. If PSRAM is not present, the configuration uses SVGA resolution and higher JPEG quality with a single frame buffer.Additional pin configurations and settings specific to certain camera models are included within conditional compilation blocks.The esp_camera_init(&config) function is called to initialize the camera with the provided configuration. If the initialization fails, an error message is printed. Further adjustments are made

to the sensor settings to account for specific camera models, such as flipping the image vertically, adjusting brightness, and saturation. The frame size is set to QVGA for a higher initial frame rate.

The serial console prints the IP address of the ESP32-CAM module, indicating that it is ready for use.In the loop() function, there is a delay of 10 seconds, allowing for the execution of any additional code that may be added in the future.With our Computer vision module we were able to get the live ai feed and implement esp 32 features on top of that in order to make the detections while in transit.

2.6 Blynk Cloud Module

Blynk is a comprehensive software suite that enables the prototyping, deployment, and remote management of connected electronic devices at any scale.

Whether it's personal IoT projects or commercial connected products in the millions, Blynk empowers users to connect their hardware to the cloud and create iOS, Android, and web applications, analyze real-time and historical data from devices, remotely control them from anywhere, receive important notifications, and much more.

Fig 2.5 : Blynk cloud module Dashboard

Blynk Library is a user-friendly and portable C++ library, that comes pre-configured to work with hundreds of development boards. It implements a streaming connection protocol, allowing for low- latency and bi-directional communication.

2.6.1 Data Flow From Blynk to Platform

With Blynk you can send raw or processed data from any sensor or actuator connected to the MCU board When you send data to Blynk it flows through a Data stream using Blynk protocol. Then every value is automatically timstamped and stored in the Blynk .Cloud database (you can also send batches of timestamped data if needed).Datastream is a channel that tells Blynk what type of data is flowing through it .With Blynk you can send any raw or processed data from any sensor or actuator.

Virtual Pins in Blynk allow for data exchange between hardware and the Blynk app. Unlike physical pins, they provide hardware-independent functionality, making it easier to transition between different hardware platforms. With Virtual Pins, you can send data from the app, process it on the microcontroller, and send it back to the smartphone. This abstraction enables various interactions such as triggering functions, reading I2C devices, controlling motors, and interfacing with external libraries. Additionally, Virtual Pins offer more control over widget behavior and stability compared to manipulating digital pins. When using Virtual Pins, there is no direct correlation between them and the physical GPIO pins on your hardware; you need to implement the code to link them. The data sent through Virtual Pins can be stored as raw data or averaged based on the chosen plan, and the Chart Widget in Blynk.Console can be used to visualize and store the data.

2.7 Network Gateway Module

The network gateway module serves as a crucial component of the project, providing transparent network information and facilitating seamless administration through the admin control application. This section highlights the key features and functionalities of the network gateway module, which utilizes JavaScript to render and display essential network details.

I. **DNS Name:** The module retrieves and displays the DNS name associated with the

network. The DNS name provides a human-readable identifier for the network, making it easier for administrators to recognize and manage multiple networks if applicable.[6]

II. **IP Address:** The network gateway module retrieves the IP address assigned to the network. This information is vital for identifying and accessing devices within the network and allows administrators to establish connections and perform network-related tasks.

III. **ISP Provider:** By querying the network, the module retrieves and presents the name of the Internet Service Provider (ISP). This information helps administrators identify the company responsible for providing internet connectivity to the network and aids in troubleshooting network issues if required.

IV. **Network Latency:** The module measures and displays the network latency, which refers to the time it takes for data packets to travel from the source to the destination and back. Network latency is crucial for assessing network performance and identifying potential bottlenecks or connectivity issues.

V. **Network Speed**: The module measures and showcases the network speed, indicating the data transfer rate between devices within the network. This information helps administrators monitor network performance and optimize data transmission for efficient communication.[7]

VI. **Router Name:** The module retrieves and presents the name of the router used in the network. The router name is helpful for identifying and distinguishing between different network configurations or access points, particularly in complex network setups.

The provided code snippet demonstrates several JavaScript functions that perform network-related operations and retrieve network information.The pingHost() function allows the user to ping a specific host or URL. It utilizes the XMLHttpRequest object to send a GET request to the specified URL. Upon receiving a response, it calculates the ping time by subtracting the start time from the current time. The result is then displayed in the designated HTML element.[8]

In the measureNetworkSpeed() function, the network speed is measured by fetching a file from a given URL. Using the fetch API, a GET request is sent to the URL, and the duration of the request is calculated by subtracting the start time from the end time. Based on the file size and duration, the network speed is calculated in Mbps and displayed in the corresponding HTML element.The checkNetworkStatus() function determines the network status by checking the value of the navigator.onLine property. If the property evaluates to true, indicating an online connection, a checkmark icon is displayed, and the text is set to "Online." Otherwise, a cross icon is displayed, and the text is set to "Offline."[12] Additionally, the code includes functions for measuring network latency and initializing the widgets on page load.

Fig 2.6: Network Gateway Module

The network gateway section in the project documentation elucidates how the network information is retrieved and how JavaScript is utilized to render and display the data in a user-friendly manner. This transparency and accessibility foster efficient network administration and facilitate informed decision-making for maintaining optimal network performance.

2.7.1 IOT Enablers

Fig 2.7 : ESP 8266 Module

Instead of relying on the complex frameworks like django and flask which would have brought a boat to cost not only in the development sector but in the maintainence sector as well. This is why the team decided to us the ESP8266 from the IOT module to build a network interfared honeypot system which would render the network the information onto the admin's device as well. The pictorial representation of which can be seen below in the given figure.

The ESP8266 microcontroller is a versatile device that can be utilized in various applications.The ESP8266 module enables microcontrollers to connect to 2.4 GHz Wi-Fi, using IEEE 802.11 bgn. It can be used with ESP-AT firmware to provide Wi-Fi connectivity to external host MCUs, or it can be used as a self-sufficient MCU by running an RTOS-based SDk One of its intriguing uses is setting up a seemingly vulnerable WiFi network with an open service server to detect any unauthorized attempts to connect. [13]

2.7.2 ESP8266 Features

The ESP8266 is a feature-rich microcontroller module renowned for its versatility and efficiency in wireless communication applications, making it a popular choice among developers and engineers. This compact device supports the 802.11 b/g/n protocols, enabling seamless connectivity across various network environments. Its capability to function within these protocols ensures broad compatibility and robust performance in diverse Wi-Fi setups.

Furthermore, the ESP8266 offers advanced networking features such as Wi-Fi Direct (P2P) and soft-AP modes, which enhance the flexibility of network configurations, allowing for direct device-to-device communication and the creation of custom access points, respectively One of the standout features of the ESP8266 is its support for antenna diversity, which significantly improves signal reliability by enabling the module to switch between multiple antennas to select the best signal path. This feature is particularly beneficial in environments with variable signal strength or interference. In terms of power consumption, the ESP8266 is designed to be highly efficient. It boasts an extremely low power down leakage current of less than 10 microamperes, making it ideal for battery-powered applications. Additionally, its standby power consumption is remarkably low, at less than 1.0 milliwatt (DTIM3), ensuring that the device can remain operational for extended periods without frequent recharging.

2.8 Admin Control Application

The requirement of an application was a must in order to bring out the best out of the retrieved abundance of data. For the project to work a complex platform application was not a requirement as the whole motoo of developing the application was just to render the data reterieved from the IDS sensors ,the gateways and the cloud platforms.[9] Developing an application with such a motive could easily be achieved by traditional application development methods and programming languages. In this case the developers decided to go with Java and XML. Java presenting as the backede and the logic for the application and XMl presentin as the interface. Many reasons for choosing to develop the application in java were . Application development became even easier on IDE as much as it could have been but the application structure and user experience still needed to be developed at the developers end which required a stable yet apealing activity / directory flow.As per the requirement a well planned activity flow was created , the traces of which and the pictorial diagram can be viewed as below.

2.8.1 Implementation

The code snippet provided includes several import statements that import different classes and libraries in Android development. Here is an explanation of each library mentioned: **android.content.Intent**: This library allows you to work with intents, which are used for inter-

component communication within an Android application. Intents can be used to start activities, services, or broadcast messages between different components. **android.support.v7.app.AppCompatActivity**: This library provides support for backward compatibility with older versions of Android. It allows you to extend the AppCompatActivity class, which is a base class for activities in Android. It provides additional features and compatibility support for various UI elements and themes. Each library mentioned above serves a specific purpose in Android development, providing ready- made classes and methods to simplify various tasks related to UI, intents, notifications, networking, and more. The provided code represents an Android activity class named dashboard in the package com.example.chakravyuh. This class extends the AppCompatActivity class and is responsible for managing the dashboard functionality of the application Within the onCreate method, the layout for the activity is set using setContentView to associate it with the XML layout file activity_dashboard.xml. Several ImageView and TextView widgets are retrieved using findViewById and assigned to corresponding member variables. Several click listeners are set on the ImageViews, and when clicked, they trigger specific actions. For example, clicking on the image ImageView starts a new activity named aiLivefeed, while clicking on image15 starts the networkGateway activity and shows a notification. Similarly, other ImageViews start different activities upon being clicked. The dashboard class also registers a network callback to monitor the internet connectivity status. This is done using the ConnectivityManager class, which checks if the device is connected to the internet. Based on the connectivity status, the visibility of the noInternetImage ImageView is toggled, and a corresponding toast message is displayed. The TextView1 is also updated with the status information The showNotification method creates and displays a notification using the NotificationManager and NotificationCompat.Builder classes. The method checks the Android version and creates a notification channel if the version is Oreo or higher. Depending on the internet connectivity status, the image in image3 ImageView is either removed or set to display a specific image. The notification is then built and displayed. Additionally, the isConnectedToInternet method checks if the device is connected to the internet using the ConnectivityManager and returns a boolean value indicating the connectivity status. Overall, this code implements the functionality of the dashboard activity, allowing users to interact with various ImageViews, monitor internet connectivity, and receive notifications based on certain conditions. The similar code snippets have been used to achieve the other aspects of the application as well.All the integrations and the cloud data rendering allowed us to present the

admin all the relevant data in realtime

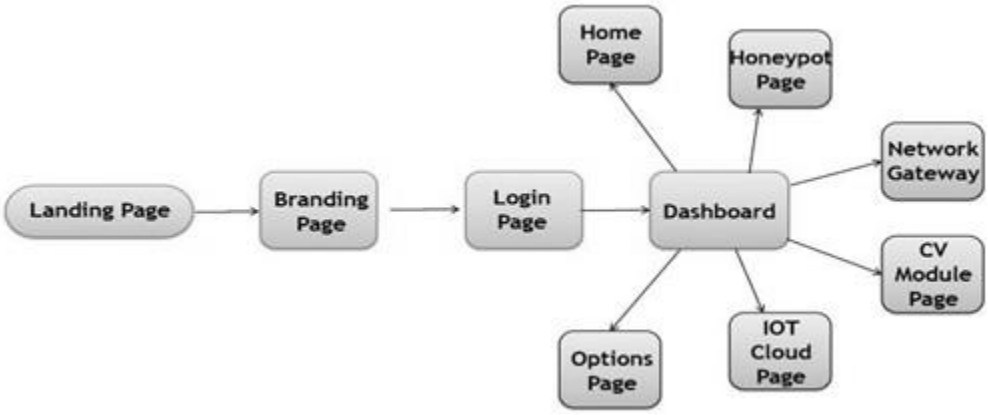

Fig 2.8 : Admin Control Application architecture

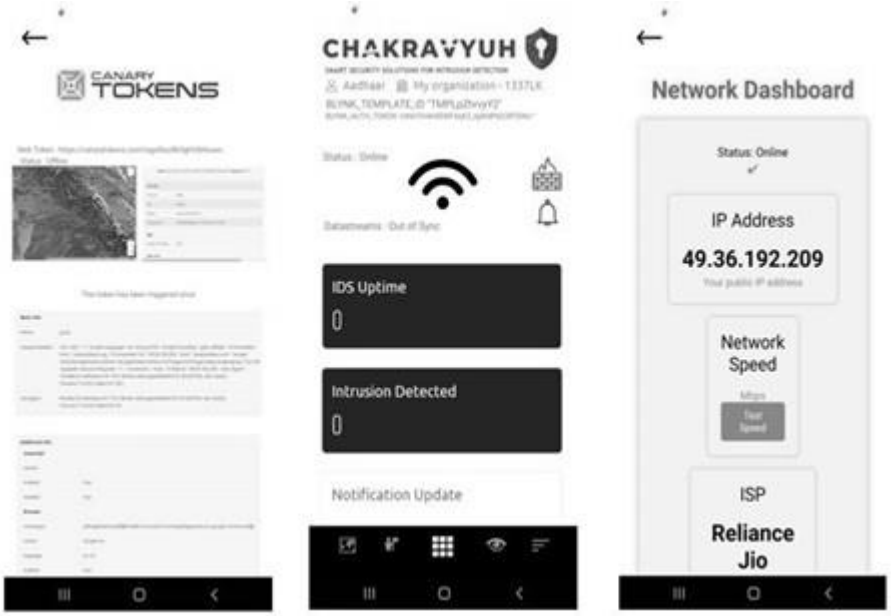

Fig 2.9: Admin control application dashboard

CHAPTER 3

PROPOSED FRAMEWORK AND WORKFLOW

3.1 Proposed Framework

A Framework in which the modules computer vision , surveillance , Blynk cloud , IOT , network gateway , Honeypot , android application work in a coordinated manner . the framework would comprises of a Network and home automation control panel that is receiving real-time responses from different components like wifi , detection sensors , arduino , home gateway , media server . The wifi render the information from a centralized home security system that comprises of a VPN , Network Sniffer , IP surveillance and a honeypot system. The detection sensors receive information from the ultrasonic sensors , laser sensors , motion sensors , PIR sensors in real-time. The arduino renders the real-time information from electricity appliances. the home gateway renders the information from the home gateway dashboard and a media server the is connected to the network storage whenever at

any instant a sensor or an activity happens on any of the modules of the framework a Smart notification system triggers a push notification in the admin control application telling them that there has been an intrusion on a particular sensor.

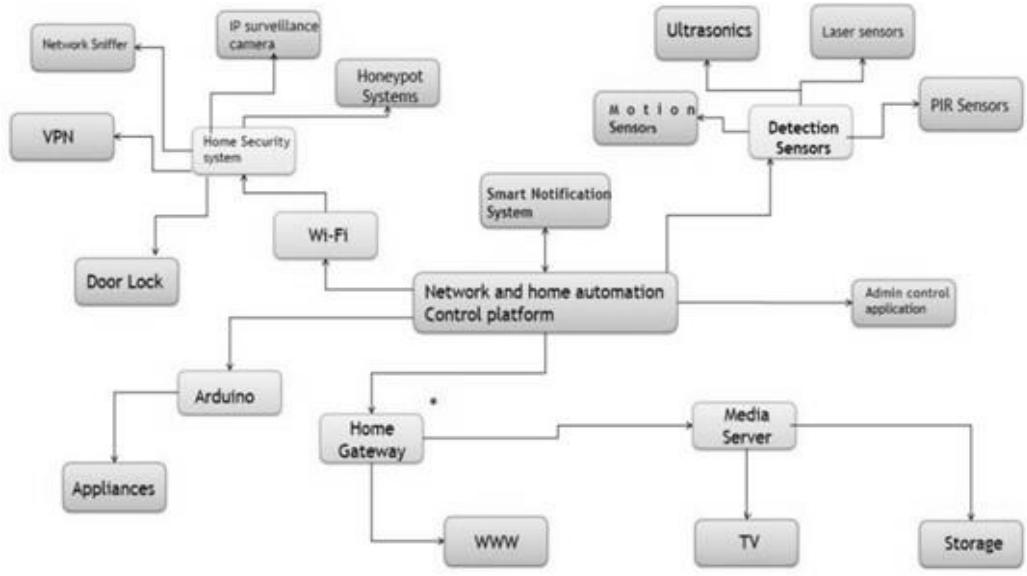

Fig 3.1 : Proposed framework for Pingfence

The framework we have developed integrates various modules, including computer vision, surveillance, Blynk cloud, IoT, network gateway, honeypot, and an Android application, to create a cohesive system. At the core of this framework is a network and home automation control panel that receives real-time responses from different components such as WiFi, detection sensors, Arduino, home gateway, and a media server. The WiFi component serves as a centralized home security system, incorporating a VPN, network sniffer, IP surveillance, and a honeypot system. These elements work together to monitor and protect the network against potential threats and unauthorized access.The detection sensors play a crucial role in the framework, as they receive real-time information from various sources such as ultrasonic sensors, laser sensors, motion sensors, and PIR (Passive Infrared) sensors. This enables the system to detect any unusual activity or intrusion accurately. The Arduino component of the framework gathers real-time data from electricity appliances, providing valuable insights into their usage and status. This information can be utilized for monitoring and controlling energy consumption. The home gateway serves as a dashboard, presenting information from various aspects of the home automation system. It provides a centralized view of the network, security status, and connected devices.Additionally, the media server connected to the network storage control capabilities while ensuring prompt notification and response to any potential threats or intrusions.

3.2 Proposed Workflow

The project operates around a below mentioned proposed pictorial work-flow which comprises of 2 decision making diamonds , 3 dialog boxes and 2 trapeziums. At first the user turns on the wifi , simultaneously turning on the IDS , which configures all the Honeypot , detection , network sniffer and the web app portal systems and then the IDS goes into a keep true state. If there in activity detected on any module the IDS stays in the Keep true state. If the IDS detects an intrusion the IDS updates all the tables and databases with the time stamps and the additional information like snapshots etc. after that the smart notification system sends an alert notification to the admin via an android application. and if the honeypot is triggered the hackers information saved onto a secret database with his / her IP address and time-stamp , DNS , ISP , location etc. after this the user is asked to take necessary actions , if the user does not take the necessary actions then the IDS would keep itself into the keep true state , but if they do take necessary actions then the application would allow the admin to turn off certain network port or a service so that the intruder

is completely shut down from the system. after this the IDS goes into the final state that is keep true after implementing the necessary protocols

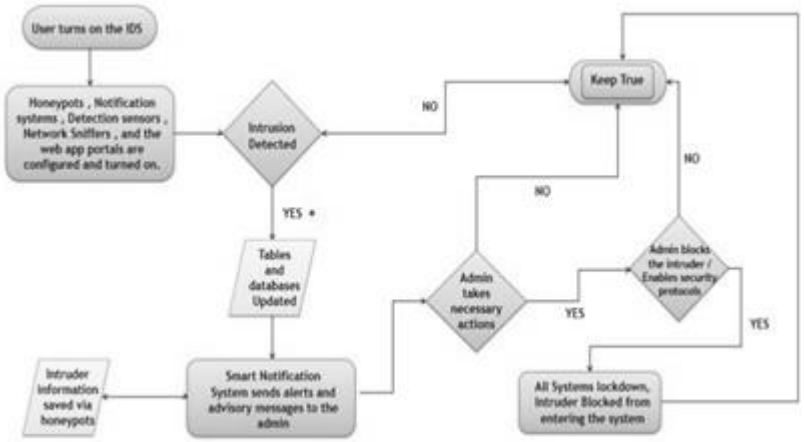

Fig 3.2 : Proposed workflow for the project

The operational sequence starts with the user initiating the process by activating the WiFi connection, a pivotal step that concurrently initializes the Intrusion Detection System (IDS). The comprehensive IDS configuration entails setting up key components such as the Honeypot, robust detection mechanisms, a network sniffer for real-time data analysis, and a web application portal system for remote monitoring and management. Once meticulously configured, the IDS seamlessly transitions into its proactive "Keep True" state, meticulously monitoring the network's every node and communication channel for any aberrant activity or potential security breaches.In the event of detected activity within its surveillance framework, the IDS remains steadfast in its "Keep True" state, ensuring continuous and vigilant surveillance. Upon detecting an intrusion, the IDS swiftly springs into action, updating pertinent tables and databases with critical details such as timestamps and comprehensive snapshots of the suspicious behavior. This triggers an intelligent notification system designed to promptly alert the designated administrator via an Android application, empowering them with real-time awareness and the ability to swiftly initiate responsive actions to mitigate risks and fortify defenses.Furthermore, the system incorporates sophisticated Honeypot technology strategically positioned to simulate vulnerable areas, effectively enticing potential hackers and capturing their actions in a secure, confidential database. The captured data includes crucial information like the hacker's IP address, precise

timestamp of the incident, DNS details, ISP information, and geographical location, facilitating in-depth forensic analysis and proactive threat intelligence gathering. Upon detection, the administrator receives immediate prompts to execute necessary response actions. Should these actions not be promptly addressed, the IDS remains in its proactive "Keep True" stance, ensuring continuous monitoring and readiness to defend against evolving threats.

once the administrator takes decisive actions to address the intrusion, the system empowers them with granular control options, allowing for targeted measures such as disabling specific network ports or services to sever the intruder's access effectively. This strategic implementation of security protocols and swift administrative responses ensures a robust defense posture, reinforcing the system's resilience against potential cyber threats. Ultimately, following the implementation of these strategic security measures, the IDS transitions into its final "Keep True" state, maintaining unwavering vigilance and proactive monitoring to safeguard the integrity and continuity of the network infrastructure.This detailed operational workflow exemplifies a methodical and integrated approach to network security, commencing with the meticulous initialization and configuration of the IDS and network environment. It systematically progresses through the detection and decisive response to intrusions or suspicious activities, culminating in the continuous assurance of system security through proactive measures and ongoing threat monitoring.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Estimations

We begin by outlining the comprehensive estimations and detailed expectations for the project, ensuring a thorough understanding of the anticipated outcomes and precise goals to be achieved. The primary objective of this ambitious initiative is to develop an integrated and sophisticated framework that seamlessly combines multiple modules essential for modern home automation and security. These modules encompass advanced technologies including computer vision for intelligent monitoring, robust surveillance systems, integration with Blynk cloud services for remote access and control, IoT devices for sensor data integration, a network gateway for secure data transmission, a honeypot system to detect and deter potential cyber threats, and a dedicated Android application for user interaction and control. The framework is designed not only to enhance convenience but also to establish a highly efficient and coordinated system ensuring unparalleled levels of home security. The development timeline for this project is meticulously planned, aiming for completion within a specified timeframe. However, given the complex nature of integrating these technologies, potential challenges and unforeseen complexities could impact the schedule. To preemptively address such risks, a comprehensive project plan will be devised, detailing specific milestones, deliverables, and resource allocation strategies. In terms of functionality, the framework will boast an array of advanced features designed to meet the diverse needs of modern homeowners. Key functionalities include real-time response capabilities for immediate action upon detection of security breaches or anomalies, a centralized home security system that integrates all components seamlessly, detection sensors employing state-of-the-art technologies, Arduino microcontroller integration for flexible device control, a home gateway dashboard offering comprehensive system oversight, media server connectivity for seamless multimedia sharing and access, and a smart notification system providing timely alerts and updates to homeowners. Moreover, to fortify the security measures, the framework will incorporate cutting-edge technologies and protocols. These include the deployment of Virtual Private Network (VPN) solutions to ensure secure communication channels, network sniffers for real-time monitoring and analysis of network traffic patterns, IP surveillance cameras equipped

with advanced analytics to track and identify suspicious activities, and a honeypot system strategically placed to attract and isolate potential attackers, thereby safeguarding the entire network from external threats. From a user experience perspective, paramount emphasis will be placed on delivering a highly intuitive and user-friendly interface through the dedicated Android application. This application will serve as the primary interface for homeowners to interact with and manage the entire system effortlessly. It will facilitate real-time monitoring, enable prompt responses to security alerts, and empower users to take necessary actions to mitigate risks or address any detected intrusions effectively. The design philosophy behind the application focuses on maximizing usability, responsiveness, and ease of navigation, ensuring a seamless and satisfying user experience.

In the success of this ambitious project hinges on several critical factors: adherence to the established timeline, seamless integration of advanced technologies, cost-effectiveness in deployment and maintenance, the delivery of a highly intuitive and user-friendly interface, and rigorous testing to validate the system's performance and reliability under various conditions. By meticulously addressing these aspects, the project aims not only to meet but exceed expectations, delivering a robust and efficient solution that significantly enhances both the security and convenience of home environments for its users.

4.2 Results

I. One key outcomes of our project is the successful integration of different modules into a coordinated framework. Using computer vision, and IoT technologies, we have created a comprehensive system that can monitor and detect intrusions in real-time. Image 4.1 Depicts the implementation of the computer vision module in the IDS. It Uses the YOLO v8 Engine libraries to classify and detect intrusions.

Fig : 4.1 Computer Vision Module Implementation

II. The implementation of our IDS, supported by various modules and integrated into a cohesive framework, has yielded positive results. Our system demonstrated its effectiveness in detecting and mitigating intrusions, offering advanced security measures to protect businesses and individuals. Its seamless integration and user-friendly interface, our IDS stands out as a reliable and comprehensive solution, surpassing other players in the market. Image 4.2 Depicts the implementation of the Admin Control Application module in the IDS. It Uses the Java and XML to make use of the functionalities of the cloud platforms and computer visions to smartly send the data to the processing centre and generate smart notifications based on the level of threat.

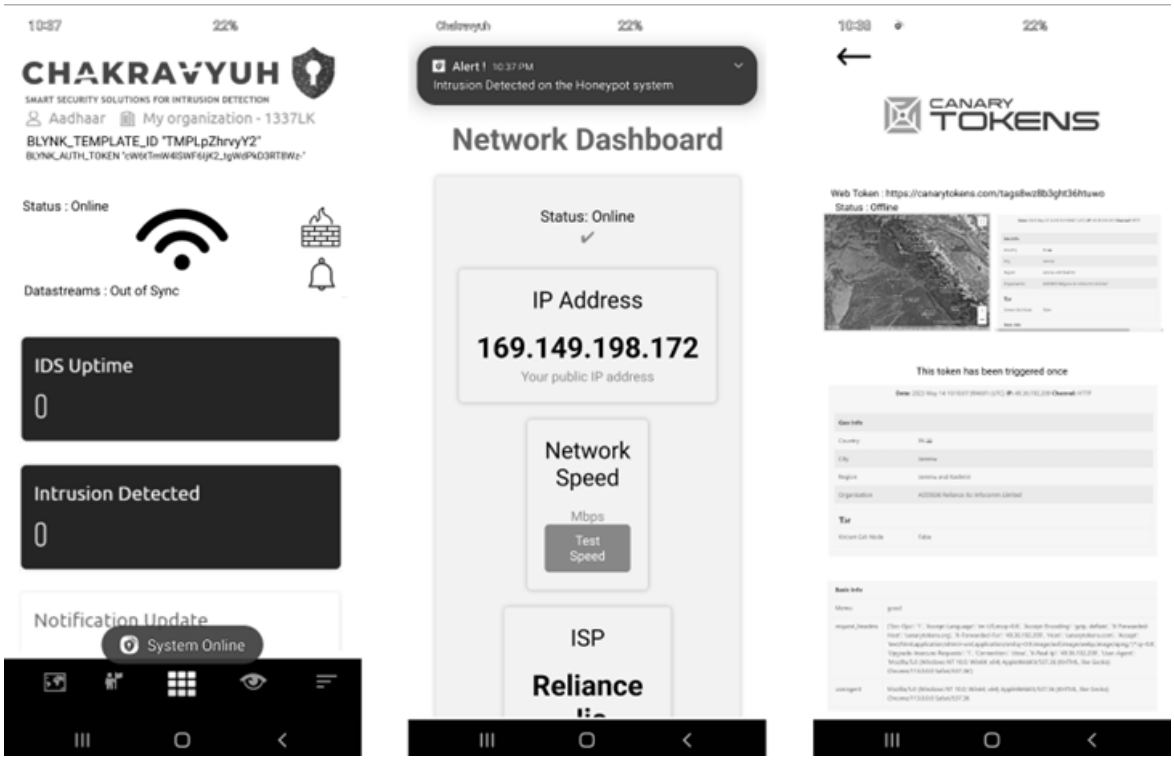

Fig : 4.2 Admin Control Application Module Implementation

III. The implementation of the smart notification system, as shown in Image 4.4, underscores the project's focus on proactive threat detection and immediate response. This system is designed to provide real-time alerts to administrators about potential security breaches, enabling them

to take swift and necessary actions to protect the network. The smart notification system ensures that any detected threats are promptly communicated, allowing for timely intervention and mitigation of risks.

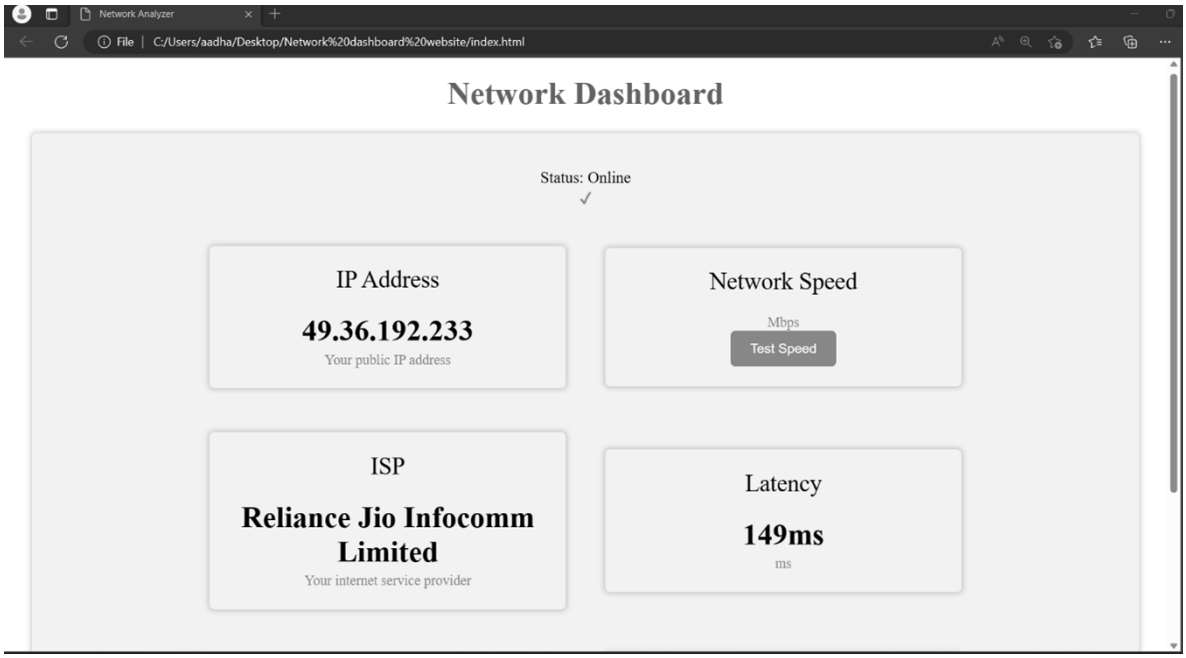

Fig : 4.3 Network Dashboard Module Implementation

IV. The results of this project highlight the successful integration of advanced technologies into a cohesive and functional framework. The combination of real-time monitoring, effective intrusion detection, comprehensive network management, and proactive notification systems collectively enhances the security and efficiency of home automation systems. This integrated approach not only provides robust protection against potential threats but also ensures a user-friendly and responsive interface for administrators, ultimately delivering a reliable and sophisticated solution for modern home security.

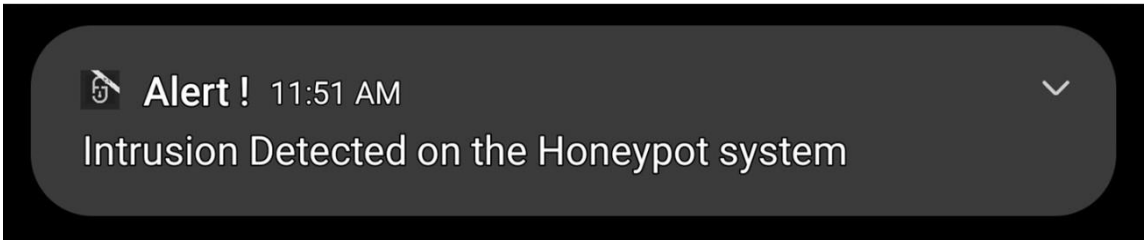

Fig : 4.4 Smart Notification Implementation

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

Our Intrusion Detection System (IDS) project has successfully developed and implemented a robust framework for enhancing network and premises security. By integrating modules such as computer vision, surveillance, Blynk cloud, IoT, network gateway, honeypot, and an Android application, our IDS offers a comprehensive and coordinated approach to real-time threat detection and response.Using Java, XML, and the Android Studio IDE, we have created a user-friendly application that empowers users with efficient monitoring and control capabilities. Our IDS addresses the security gaps prevalent among general users, providing an accessible and intuitive solution to minimize the risk of cyber attacks.Compared to other players in the market, our IDS stands out due to its comprehensive feature set, scalability, and ease of integration. It covers a wide range of security aspects, including network monitoring, intrusion detection, honeypot systems, and real-time notifications. Our IDS can be customized to meet specific industry requirements.Deployment and testing have shown promising results, with high accuracy in detecting intrusions and minimizing false positives. Our IDS enhances network security, protects sensitive information, and ensures the integrity of critical infrastructure.

5.2 Future Scope

The implemented Intrusion Detection System (IDS) and its individual modules offer significant potential for extensibility and enhancement. New methods can be implemented to further improve the accuracy and effectiveness of the IDS. For example, advanced machine learning algorithms, anomaly detection techniques, or behavioral analysis approaches can be integrated into the existing system to enhance threat detection capabilities. Additionally, we have plans to incorporate augmented reality (AR) functionality into the application using the Unity engine. This expansion will enable users to visualize network information and security alerts in a more immersive and interactive manner, enhancing their understanding and response to potential threats. With ongoing advancements in technology and cybersecurity, the future scope of the IDS is promising.

REFFERENCES

[1] S. E. Hejres and M. Hammad, "Image Recognition System Using Neural Network Techniques: an Overview," 2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Zallaq, Bahrain, 2021, pp. 706-713, doi: 10.1109/3ICT53449.2021.9581829.

[2] M. H. Ali, F. Kamaruzaman, M. A. Rahman and A. A. Shafie, "Automated secure room system," 2015 4th International Conference on Software Engineering and Computer Systems (ICSECS), Kuantan, Malaysia, 2015, pp. 73-78, doi: 10.1109/ICSECS.2015.7333086. keywords: {Security;Streaming media;Geometry;Computers;Image edge detection;Software engineering;Image color analysis;Room alarming system;motion detection;optical flow;human verification;hand geometry verification},

[3] M. -X. Chen and B. -Y. Lin, "Design remote monitor system based on OSGi platform in vehicle gateway for vehicle network," The 2nd International Conference on Next Generation Information Technology, Gyeongju, Korea (South), 2011, pp. 12-17. keywords: {Vehicles;Security;Logic gates;Bridges;Computer architecture;Monitoring;Communication system security;Vehicle Security System;Vehicle Network Gateway;Service Location Protocol;Session Initiation Protocol;Open Service Gateway initiative},

[4] M. B. Guldogan, F. Gustafsson, U. Orguner, S. Björklund, H. Petersson and A. Nezirovic, "Human gait parameter estimation based on micro-doppler signatures using particle filters," 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 2011, pp. 5940- 5943, doi: 10.1109/ICASSP.2011.5947714. keywords: {Humans;Leg;Torso;Legged locomotion;Feature extraction;Radar tracking;human gait analysis;micro-Doppler;particle filters (PF);radar},

[5] M. B. Guldogan, F. Gustafsson, U. Orguner, S. Björklund, H. Petersson and A. Nezirovic, "Radar micro-Doppler parameter estimation of human motion using particle filters," 2011 IEEE 19th Signal Processing and Communications Applications Conference (SIU), Antalya, Turkey, 2011, pp. 395-398, doi: 10.1109/SIU.2011.5929670. keywords: {Humans;Radar tracking;Doppler radar;Conferences;Signal processing;Time frequency analysis},

[6] K. Ishiguro and R. Huang, "Implementation of a wireless communication technologies based home security system," 2011 3rd International Conference on Computer Research and Development, Shanghai, China,2011,pp.394-398,doi:10.1109/ICCRD.2011.5764043.

[7] M. -X. Chen and Y. -C. Chuang, "Supporting Home Security System in OSGi Platform," 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, Perth, WA, Australia, 2010, pp. 280-285, doi: 10.1109/WAINA.2010.115. keywords: {Home automation;Information security;Protocols;National security;Communication system security;Home appliances;Computer security;Computer architecture;Automatic control;Control systems;Home Security System;Open Service Gateway initiative;Service Location Protocol;Session Initiation Protocol;Universal Plug and Play},

[8] D. Mitrović, M. Zeppelzauer and H. Eidenberger, "On feature selection in environmental sound recognition," 2009 International Symposium ELMAR, Zadar, Croatia, 2009, pp. 201-204. keywords: {Data analysis;Frequency;Principal component analysis;Music information retrieval;Spatial databases;Linear predictive coding;Cepstral analysis;Information retrieval;Speech recognition;Fourier transforms;Feature Selection;Statistical Data Analysis;Environtmental Sound Recognition},

[9] M. Kawamoto, F. Asano, K. Kurumatani and Y. Hua, "A System for Detecting Unusual Sounds from Sound Environment Observed by Microphone Arrays," 2009 Fifth International Conference on Information Assurance and Security, Xi'an, China, 2009, pp. 729-732, doi: 10.1109/IAS.2009.200. keywords: {Microphone arrays;Safety;Cameras;Surveillance;Training data;Feature extraction;Data security;Information security;National security;Signal processing;Security and safety system;Unusual sounds;Sound direciton estimation;Sound classification;Microphone array},

[10] J. See and S. -W. Lee, "An integrated vision-based architecture for home security system," in IEEE Transactions on Consumer Electronics, vol. 53, no. 2, pp. 489-498, May 2007, doi: 10.1109/TCE.2007.381720. keywords: {Face recognition;Motion detection;Authentication;Computer security;Communication system security;Information security;Cameras;Lighting control;Control systems;Multimedia systems},

[10] M. Padavala, M. S. S and J. Valarmathi, "A Novel Intruder Alert System Using LoRa and FMCW Radar," 2023 Innovations in Power and Advanced Computing Technologies (i-PACT), Kuala Lumpur, Malaysia, 2023, pp. 1-5, doi: 10.1109/i-PACT58649.2023.10434731.

[12] R. Islam, M. I. Hossain, M. S. Rahman, S. Kabir, M. S. R. Sohan and A. Shufian, "Smart IoT System for Automatic Detection and Protection from Indoor Hazards: An Experimental Study," 2022 IEEE 10th Region 10 Humanitarian Technology Conference (R10-HTC), Hyderabad,India,2022,pp.112-117,doi:10.1109/R10-HTC54060.2022.9929677.

[13] M. S. R. Sohan, S. Kabir, M. J. -A. -M. Hoque and A. Shufian, "Automatic Protection of Electrical and Gas Transmission System on Earthquake," 2022 IEEE Region 10 Symposium (TENSYMP), Mumbai, India,2022,pp.1-6,doi:10.1109/TENSYMP54529.2022.9864518

[14] R. Akter, A. Shufian, M. M. Rahman, R. Islam, S. Kabir and M. J. -A. -M. Hoque, "IoT and Solar Based Smart Farming Technique," 2021 IEEE International Conference on Power, Electrical, Electronic and Industrial Applications (PEEIACON), Dhaka, Bangladesh, 2021, pp. 1-4, doi: 10.1109/PEEIACON54708.2021.9929706.

[15] "Image reference 1," [Online]. Available: https://images.app.goo.gl/PebG9eZQjoM1JBuV8. [Accessed: 07-Jul-2024].

[16]"Image reference 2," [Online]. Available: https://images.app.goo.gl/Dz2zNVY5a9FemShV7. [Accessed: 07-Jul-2024].

[17]"Image reference ," [Online]. Available: https://images.app.goo.gl/2za9ebG92FFemShV72z. [Accessed: 07-Jul-2024].

APPENDIX – A

3.2.1 Libraries Used

import android.support.v7.app.AppCompatActivity; import android.os.Bundle;
import android.widget.Toast; import java.util.Timer; import java.util.TimerTask;
import android.annotation.SuppressLint; import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity; import android.view.View;
import android.widget.Button; import android.widget.EditText; import android.widget.Toast;
import android.app.Notification;
import android.app.NotificationChannel; import android.app.NotificationManager; import android.content.Context;
import android.content.Intent; import android.graphics.Color;
import android.net.ConnectivityManager; import android.net.Network;
import android.net.NetworkCapabilities; import android.net.NetworkInfo;
import android.os.Build; import android.os.Bundle;
import android.support.v4.app.NotificationCompat; import android.support.v7.app.AppCompatActivity; import android.view.View;
import android.widget.ImageView; import android.widget.TextView; import android.widget.Toast;
import android.annotation.SuppressLint;
import android.support.v7.app.AppCompatActivity; import android.os.Bundle;
import android.webkit.WebSettings; import android.webkit.WebView;

5.2.1 ESP-32 CAM Script

```
#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
//         Ensure ESP32 Wrover Module or other board with PSRAM is selected
```

```
//          Partial images will be transmitted if image exceeds buffer size
//

// Select camera model
#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
//#define CAMERA_MODEL_ESP_EYE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
//#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
//#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
//#define CAMERA_MODEL_AI_THINKER // Has PSRAM
//#define  CAMERA_MODEL_TTGO_T_JOURNAL  //  No  PSRAM
      #include "camera_pins.h"
      const  char*  ssid  =  "**********";  const
      char* password = "**********";

      void startCameraServer();

      void loop() {
      // put your main code here, to run repeatedly:
      delay(10000);
      }



      xhr3.open('GET',  'https://ipwhois.app/json/'  +  ip,  true);
      xhr3.onload = function() {
      if (this.status === 200) {
      var data2 = JSON.parse(this.responseText); document.getElementById('isp').innerHTML =
      data2.isp;
      }
      };
      xhr3.send();
```

```
    }
    };
    xhr2.send();


// Router
    document.getElementById('router').innerHTML = location.hostname;


// DNS
    var dns = 'example.com';
    var xhr = new XMLHttpRequest();
    xhr.open('GET', 'https://dns.google/resolve?name=' + dns, true);
    xhr.onload = function() {
    if (this.status === 200) {
    var data = JSON.parse(this.responseText); var
    ipAddress = data.Answer[0].data;
    console.log('DNS: ' + ipAddress);
    document.getElementById('dns').innerHTML = ipAddress;
    }
    };
    xhr.send();


//speed
// Speed test
    function measureNetworkSpeed() {
    const startTime = Date.now();
    const fileSize = 1024 * 1024 * 10; // 10 MB
    const url = "https://speed.hetzner.de/10GB.bin"; // replace with your preferred file URL const
    xhr = new XMLHttpRequest();
    xhr.open("GET", url + "?r=" + Math.random(), true);
```

```
xhr.responseType = "arraybuffer";

xhr.onload = function(e) { const
endTime = Date.now();
const duration = (endTime - startTime) / 1000;
const bitsLoaded = fileSize * 8;
const speedBps = bitsLoaded / duration;
const speedMbps = (speedBps / (1024 * 1024)).toFixed(2);
console.log("Download speed: " + speedMbps + " Mbps");
document.getElementById("speed").innerHTML = speedMbps;
};

xhr.onerror = function(e) { console.error("Error
fetching file", e);
};


xhr.send();
}
```

5.2.2 ESP8266 Honeypot Code

Code :

```
// NAPT example released to public domain #if
LWIP_FEATURES && !LWIP_IPV6
    #define HAVE_NETDUMP 0

    #ifndef STASSID
    #define STASSID "Your_Wifi_Network_Name" // set the SSID (name) of the Wi-Fi network the
ESP8266 will connect to for internet
```

```
#define STAPSK  "Your_Wifi_Network_Password" // set the password of the Wi-Fi network the ESP8266 will connect to for internet
#define NEWSSID  "honeypot_Wifi_Name" // set the name (SSID) of the Wi-Fi network the ESP8266 will create
#define NEWPASS  "honeypot_Wifi_Password" // set the password of the Wi-Fi network the ESP8266 will create
#endif

#include <ESP8266WiFi.h>
#include <lwip/napt.h>
#include <lwip/dns.h> #include
<dhcpserver.h> #include
<ESPCanary.h>

#define NAPT 1000
#define NAPT_PORT 10 #if
HAVE_NETDUMP
#include <NetDump.h>

void setup() {
Serial.begin(115200);
Serial.printf("\n\nNAPT Range extender\n");
Serial.printf("Heap on start: %d\n", ESP.getFreeHeap());

#if HAVE_NETDUMP
phy_capture = dump; #endif

// first, connect to STA so we can get a proper local DNS server
WiFi.mode(WIFI_STA);
WiFi.begin(STASSID, STAPSK);
while (WiFi.status() != WL_CONNECTED) {
Serial.print('.');
```

```
delay(500);
}
Serial.printf("\nSTA: %s (dns: %s / %s)\n",
WiFi.localIP().toString().c_str(),
WiFi.dnsIP(0).toString().c_str(),
WiFi.dnsIP(1).toString().c_str());

// give DNS servers to AP side
dhcps_set_dns(0, WiFi.dnsIP(0));
dhcps_set_dns(1, WiFi.dnsIP(1));

WiFi.softAPConfig( // enable AP, with android-compatible google domain IPAddress(172,
217, 28, 254),
IPAddress(172, 217, 28, 254),
IPAddress(255, 255, 255, 0));
WiFi.softAP(NEWSSID, NEWPASS);
Serial.printf("AP: %s\n", WiFi.softAPIP().toString().c_str());

Serial.printf("Heap before: %d\n", ESP.getFreeHeap()); err_t
ret = ip_napt_init(NAPT, NAPT_PORT);
Serial.printf("ip_napt_init(%d,%d): ret=%d (OK=%d)\n", NAPT, NAPT_PORT, (int)ret,
(int)ERR_OK);
if (ret == ERR_OK) {
ret = ip_napt_enable_no(SOFTAP_IF, 1);
Serial.printf("ip_napt_enable_no(SOFTAP_IF): ret=%d (OK=%d)\n", (int)ret, (int)ERR_OK); if
(ret == ERR_OK) {


Serial.printf("WiFi Network '%s' with password '%s' is now NATed behind '%s'\n", NEWSSID,
NEWPASS, STASSID);
}
}
```

```
Serial.printf("Heap after napt init: %d\n", ESP.getFreeHeap()); if
(ret != ERR_OK) {
Serial.printf("NAPT initialization failed\n");
}

/////FTP Setup, ensure SPIFFS is started before ftp;  ///////// if
(SPIFFS.begin()) {
Serial.println("SPIFFS   opened!");     ftpSrv.begin(ftp_user,ftp_pass,canary,append_ip,append_char);
                                                                    //username, password for ftp.
set ports in ESPCanary.h  (default 21, 50009 for PASV)
}
}

#else
#error "NAPT not supported in this configuration" void
setup() {
Serial.begin(115200);
Serial.printf("\n\nNAPT not supported in this configuration\n");
```

APPENDIX – B

SPECIFICATIONS

Specification of Arduino UNO

Microcontroller	ATmega168
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14
Analog Input Pins	6
DC Current per I/O Pin	40mA
DC Current for 3.3V Pin	50mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Table 1.2 : Specifications of Arduino UNO

Specification of Raspberry Pi

Microcontroller	ATMEGA328P/ATMEGA158
Operating Voltage	5V
Input Voltage	7-12 V
Digital I/O Pins	40
PWM	6 out of 14 digital pins
Max Current Rating	40mA
USB	A
Analog Pins	8
Flash Memory	1Gb
SRAM	1Gb - 2Gb
Crystal Oscillator	16 MHz
EEPROM	512bytes or 1KB
USART	Yes

Table 1.3 : Specifications of Raspberry Pi4 Model B

Specification of ESP -32

Operating Voltage	2.2V to 3.6V
GPIO	36 Ports
ADC	14 Ports
DAC	2 Ports
Flash Memory	16 Mbytes
SRAM	250 Kbytes
Clock Speed	Up to 240 MHz
Wi-Fi	2.4 GHz
Sleep Current	2.5 Micro Ampere

Table 1.4 : Specifications of ESP