

Cloud-Based Intrusion Detection in Fog Computing Using Naïve Bayes Algorithm

Ajibola Oluwatobi Akanni¹, Prince O. Asagba², Chidiebere Ugwu³

^{1,2,3} Department of Computer Science, University of Porthacourt, Rivers State, Nigeria

ARTICLE INFO	ABSTRACT
Published Online: 13 April 2026	The rapid growth of computing, a decentralized paradigm extending cloud computing capabilities closer to end users, has introduced new vulnerabilities and security challenges on file intrusion. Traditional Intrusion Detection Systems (IDS) often fall short in addressing the unique architecture. This study addresses the problem by developing a secured based intrusion detection model tailored for fog computing using the Naïve Bayes Algorithm. The methodology used in this work is object-oriented analysis and design methodology (OOADM) and python programming language is used to implement the model. The system integrates TensorFlow and OpenCV resources for robust data processing and real-time analysis. The model utilizes a policy-driven approach to handle security questions, ensuring compliance with predefined rules and standards. The results demonstrated the model's capabilities to detect various types of intrusions and protection of data, such as unauthorized access attempts, with minimal computational overhead. The integration of Naïve Bayes classifier in a Fog environment is proven to be efficient and reliable, offering a scalable solution for real-time threat detection. The model was trained and tested using an 80/20 data split, achieving high accuracy and low latency in file intrusion detection. The model successfully identified and classified potential threats within Fog network, achieving an impressive accuracy of 97%. Its effectiveness was further evaluated through F1-score, which stood at 0.92, indicating a robust balance between precision and recall. This real-time detection is able to protect data from unauthorized parties from having access to files on the cloud.
Corresponding Author: Prince O. Asagba	
KEYWORDS: File Intrusion, Cloud Computing, Fog Computing, Machine Learning	

INTRODUCTION

The Internet of Things (IoT) has emerged as a transformative technology due to its on-demand service delivery, scalability, and ability to interconnect heterogeneous devices across distributed environments. With the rapid adoption of cloud computing for IoT applications, cloud platforms are now extensively used for data storage, big data analytics, and computation-intensive tasks [1]. However, this integration has significantly amplified concerns related to data security and data privacy, particularly for business-critical and sensitive information. Cloud security fundamentally focuses on ensuring data confidentiality, integrity, and availability, while data privacy emphasizes preventing the unauthorized identification, inference, or misuse of user and organizational data stored in the cloud. Despite advancements in cloud and IoT technologies, existing security mechanisms remain insufficient to fully protect IoT-generated data from evolving

cyber threats. The distributed and resource-constrained nature of IoT devices makes them highly vulnerable to attacks such as data leakage, unauthorized access, ransomware, and data manipulation. [2] highlighted that data security involves protecting corporate data from unauthorized access and preventing data loss caused by cyberattacks that encrypt, destroy, modify, or corrupt data. In cloud-based IoT environments, ensuring consistent security across multiple devices, networks, and platforms remains a major challenge [3]. Anomaly detection in data networks is difficult due to the dynamic behavior of IoT traffic, making it hard to distinguish between legitimate variations and malicious activities. The motivation for strengthening data security and privacy in IoT-cloud environments arises from the increasing dependence of critical sectors on these technologies. Industries such as finance, healthcare, and e-commerce require a high level of data protection to comply with strict regulatory standards. For

instance, organizations processing payment card information must adhere to Payment Card Industry Data Security Standards (PCI-DSS), while healthcare institutions in the United States are required to protect Private Health Information (PHI) in compliance with the Health Insurance Portability and Accountability Act (HIPAA). Failure to ensure robust data security can result in financial loss, reputational damage, legal penalties, and loss of user trust. Therefore, there is a strong need to develop intelligent, scalable, and adaptive security mechanisms capable of safeguarding IoT data against both known and unknown threats. Furthermore, data network security plays a crucial role in protecting IoT-cloud infrastructures from accidental or malicious damage. In this context, anomalies are defined as patterns, behaviors, or events that deviate from established normal network activities. Detecting such anomalies is essential for early identification of security breaches, insider threats, and system misconfigurations. Consequently, improving anomaly detection and privacy-preserving techniques in IoT-enabled cloud systems remains a key research motivation to ensure secure, reliable, and trustworthy digital services.

Related Works

[4] conducted a study titled Data Security and Privacy Challenges in Cloud-Based IoT Systems, which addressed the growing problem of unauthorized access and data leakage in cloud-supported IoT environments. The study identified that traditional security mechanisms were inadequate due to the distributed and dynamic nature of IoT devices. To solve this problem, Ahmed proposed a layered security framework combining encryption, access control, and continuous monitoring. The results demonstrated improved data confidentiality and reduced unauthorized access incidents. However, the study was limited by its reliance on static security rules, which reduced adaptability to evolving attack patterns.

[5], in their work Privacy Preservation in IoT-Cloud Integrated Architectures, examined the issue of user identity exposure and sensitive data inference in IoT-cloud systems. The authors proposed a privacy-preserving data aggregation technique using anonymization and lightweight encryption. Experimental results showed enhanced privacy protection with minimal computational overhead. Nevertheless, the approach showed reduced efficiency when applied to large-scale real-time IoT data streams.

[6] presented a study titled Anomaly Detection Techniques for Securing IoT Networks, which focused on the difficulty of identifying abnormal network behavior in IoT-based cloud infrastructures. The problem stemmed from high traffic variability and device heterogeneity. The authors proposed a machine learning-based anomaly detection model using ensemble classifiers. The results indicated high detection accuracy and reduced false positives. However, the model

required a large labeled dataset for training, which limited its applicability in resource-constrained environments.

[7], in their research Secure Data Storage Framework for Cloud-Based IoT Applications, addressed the challenge of ensuring data integrity and confidentiality during cloud storage. The study proposed a hybrid cryptographic approach combining symmetric and asymmetric encryption. The evaluation results showed enhanced resistance to data tampering and unauthorized access. Despite these improvements, the solution increased computational complexity and energy consumption on IoT devices.

[8] investigated privacy risks in their study User Privacy Concerns in Cloud-Enabled IoT Systems. The authors highlighted the problem of excessive data collection and insufficient user consent mechanisms. To mitigate this, they proposed a user-centric privacy control model that allowed users to define data-sharing preferences. The results revealed improved transparency and user trust. However, the model depended heavily on user awareness, which limited its effectiveness among non-technical users.

[9] proposed A Secure Access Control Mechanism for IoT-Cloud Environments, focusing on unauthorized privilege escalation in cloud-based IoT systems. The study introduced a role-based access control (RBAC) scheme enhanced with contextual awareness. Experimental analysis showed improved access control accuracy and reduced insider threats. The limitation of the study was that it did not adequately address scalability issues in highly dynamic IoT networks.

[10], in the work titled Data Privacy Risks in Emerging Cloud Technologies, examined the ethical and technical challenges associated with cloud data privacy. The study identified gaps in regulatory compliance and enforcement across cloud platforms. To address this, Marwick proposed a compliance-aware data governance framework. The results showed improved alignment with data protection regulations. However, the framework lacked practical implementation and real-world validation.

[11] conducted a study titled Network Intrusion Detection in IoT-Cloud Systems Using Deep Learning, which addressed the challenge of detecting sophisticated cyberattacks in IoT networks. The authors proposed a deep learning-based intrusion detection system using convolutional neural networks. The system achieved high detection accuracy against multiple attack types. Despite this, the model required high computational resources, making it unsuitable for deployment on low-power IoT devices.

METHOD OF THE PROPOSED SYSTEM

This research adopted the Object-Oriented Analysis and Design Methodology (OOADM) to analyze, model, and implement the proposed system as a set of interacting objects and classes. OOADM was selected because it provided an efficient, reliable, reusable, and structured approach to system development, while also supporting clear visual modeling to

enhance stakeholder communication and overall software quality. The study implemented a File Document Intrusion Detection Model in a fog computing environment using the Naive Bayes algorithm. Fog computing was utilized to place computational resources closer to the network edge, which reduced latency and enabled real-time monitoring of file document activities. The intrusion detection model was integrated into fog nodes to efficiently monitor, detect, and respond to suspicious file-related behaviors. Data required for the model were collected and preprocessed from file access logs, including records of file access, modification, deletion, timestamps, user identities, and access permissions. The preprocessing phase involved labeling the data as either normal or intrusive and extracting relevant features such as unusual access patterns and unauthorized file operations. These processed features were then used to train the Naive Bayes classifier. The Naive Bayes model was trained by computing prior and conditional probabilities from historical data to distinguish between normal and malicious file access behaviors. After training, the model was deployed across

distributed fog nodes, allowing each node to independently analyze file interactions in real time. When a high probability of intrusion was detected, the system generated alerts and supported automated responses such as blocking access, logging events, and notifying system administrators. For system evaluation, a dataset titled *Data Traffic* obtained from Kaggle, consisting of 3,500 records, was used. The dataset contained detailed network traffic attributes, including IP addresses, ports, protocols, packet sizes, and timestamps. The dataset was divided into training and testing subsets to evaluate model performance using metrics such as accuracy, precision, recall, and F1-score. The results showed that the Naive Bayes-based intrusion detection system effectively identified malicious activities with low false-positive and false-negative rates. The system demonstrated robustness, scalability, and suitability for real-time deployment in fog computing environments, thereby enhancing overall network and file document security. Figure 1: Proposed Architecture of the Proposed System

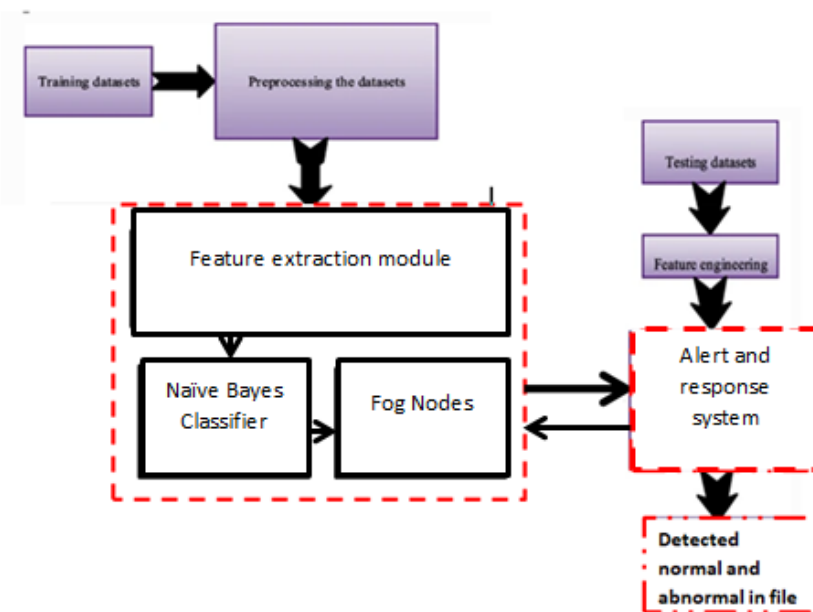


Figure 1: Proposed Architecture of the Proposed System

The first step in the system involves pre-processing, which collects and analyzes user login details. This includes capturing login times, personal security questions and ip

address. By tracking these details, the system establishes a baseline of normal user behavior.

Bayes' Theorem:

Bayes' Theorem is represented as

$$P(H|E) = \frac{P(E|H) \times P(H)}{P(E)}$$

Where:

- $P(H|E)P(H | E)P(H|E)$ is the probability that the hypothesis HHH (e.g., the user is abnormal) is true given evidence EEE (e.g., certain user actions).
- $P(E|H)P(E | H)P(E|H)$ is the probability of observing the evidence given that the hypothesis is true.

- iii. $P(H)P(H)P(H)$ is the prior probability of the hypothesis being true.
- iv. $P(E)P(E)P(E)$ is the overall probability of the evidence.

Probability Calculation:

The classifier calculates the probability that a given set of user activities is either normal or abnormal based on previously labeled data (historical normal and abnormal activity).

For each user action (or set of actions), the system computes:

$$P(\text{Normal}|\text{User Actions}) = \frac{P(\text{User Actions}|\text{Normal}) \times P(\text{Normal})}{P(\text{User Actions})}$$

Similarly, it computes:

$$P(\text{Abnormal}|\text{User Actions}) = \frac{P(\text{User Actions}|\text{Abnormal}) \times P(\text{Abnormal})}{P(\text{User Actions})}$$

Classification Decision:

After computing the probabilities, the classifier compares $P(\text{Normal}|\text{User Actions})$ and $P(\text{Abnormal}|\text{User Actions})$. The higher probability determines whether the system classifies the user as normal or abnormal.

Fog Nodes

Fog computing improves the system's response time, which is essential for detecting intrusions and taking action before critical files are compromised.

Alert and Response System

When an abnormal user behavior is detected, the alert and response system is triggered. This module is responsible for encrypting the file. Making it to be in unreadable format.

Class Diagram of the System

Users created accounts by providing basic credentials, while the system logged registration details such as IP address and time for security monitoring. Fog computing encrypted and decrypted user data, granting access only after key verification. The Naïve Bayes model monitored activities to detect abnormal behavior and potential intrusions. Figure 2: Class diagram of proposed system

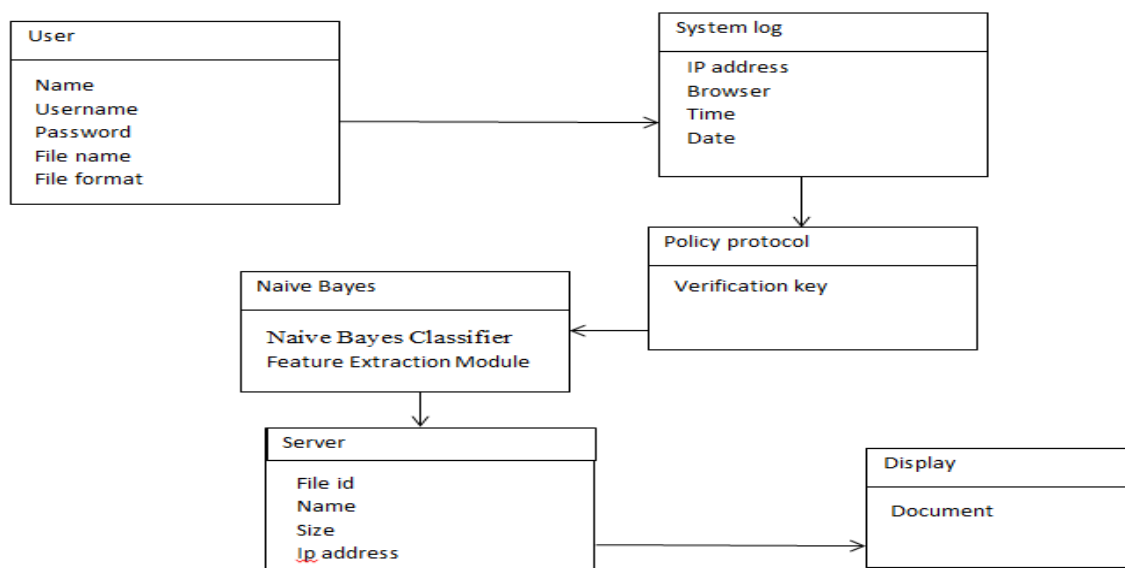


Figure 2: Class diagram of proposed system

Implementation

All the models were built and implemented using Python programming language (version 3.6) utilizing libraries like numpy, pandas, SciPy, PIL, pickle (Python Object serialization library) and matplotlib for plotting graphs, etc. Scientific python development environment (Spyder) by Anaconda was used as the platform to code in python. Spyder is a powerful Python Integrated Development Environment (IDE) with advanced editing, interactive testing, debugging

and introspection features. The implemented Naïve Bayes-based intrusion detection system was evaluated using the testing dataset to assess its effectiveness in identifying malicious file activities within the fog computing environment. The results showed that the model accurately classified file activities into benign and malicious categories, achieving high accuracy with minimal false positives and false negatives. The policy question was trained to detect intrusions in files within fog computing environments using

“Cloud-Based Intrusion Detection in Fog Computing Using Naïve Bayes Algorithm”

the Naive Bayes algorithm. The process began with the collection of a diverse dataset, which included both benign and malicious file activity logs. This dataset was pre-processed to remove any noise and irrelevant information, ensuring that only the essential features for intrusion detection were retained. Feature extraction techniques were applied to identify key characteristics of the files, such as file size, type, access patterns, and modification timestamps. These features were transformed into a suitable format for the Naive Bayes algorithm.

The Naive Bayes model was then trained using a portion of the dataset, where the algorithm learned to associate specific

features with either benign or malicious activity. The training process involved calculating the prior probabilities of each class (benign or malicious) and the conditional probabilities of each feature given the class. Once trained, the model was validated using a separate testing set to evaluate its accuracy and performance. Figure 3 Index page . The result home page served as the central dashboard, displaying the outcomes of the uploaded files and analyzed data. Users could view summarized results and relevant insights derived from the system’s processing. This page acted as the primary interface for navigating between other functionalities such as file uploads, downloads, and detailed analytics.

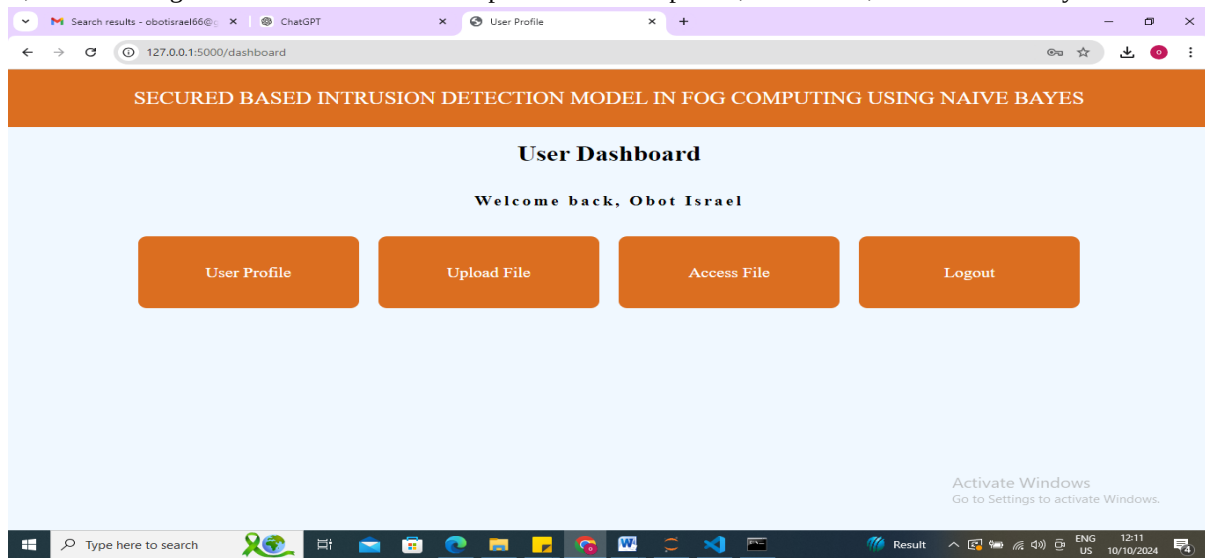


Figure 3 Index Page

Figure 4. File Uploading Page. The file uploading page allowed users to upload their files for analysis. The interface was designed to be intuitive, offering clear instructions on

how to select and upload files. Once a file was successfully uploaded, a confirmation message appeared, and the user was redirected to the result home page for further actions.

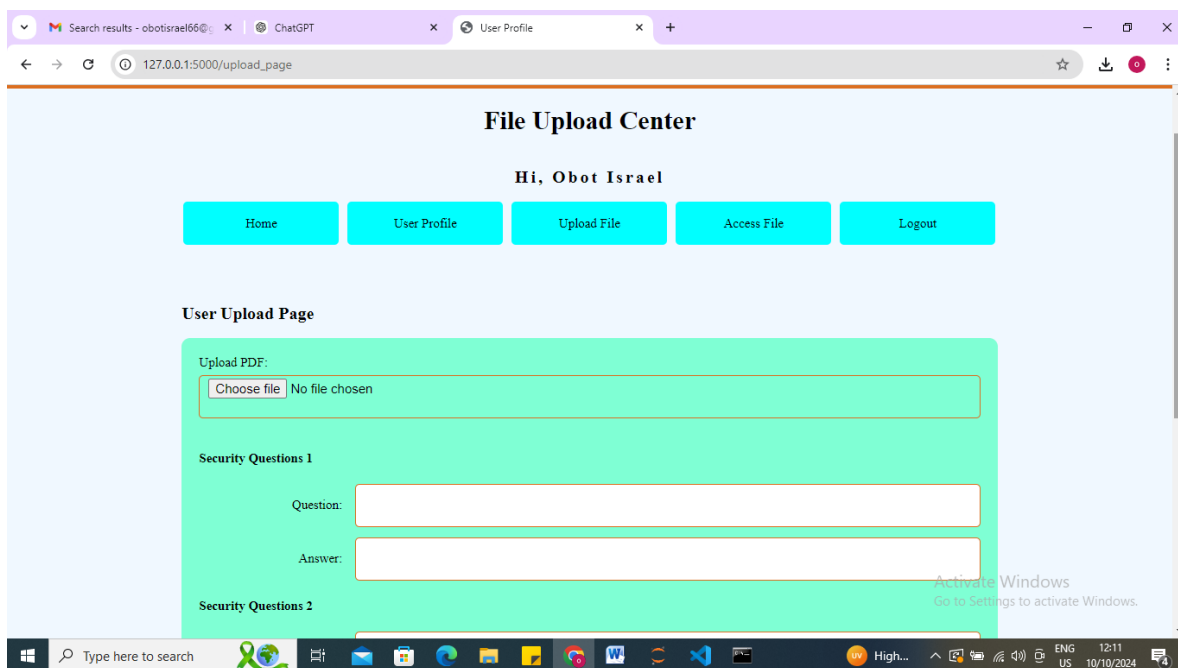


Figure 4 File Uploading Page

“Cloud-Based Intrusion Detection in Fog Computing Using Naïve Bayes Algorithm”

Figure 5 and Figure 6 File Center Page. On the file download page, users could download the processed files or reports generated by the system. A list of previously uploaded and

analyzed files was displayed, along with download links. This page ensured that users could easily retrieve their processed data

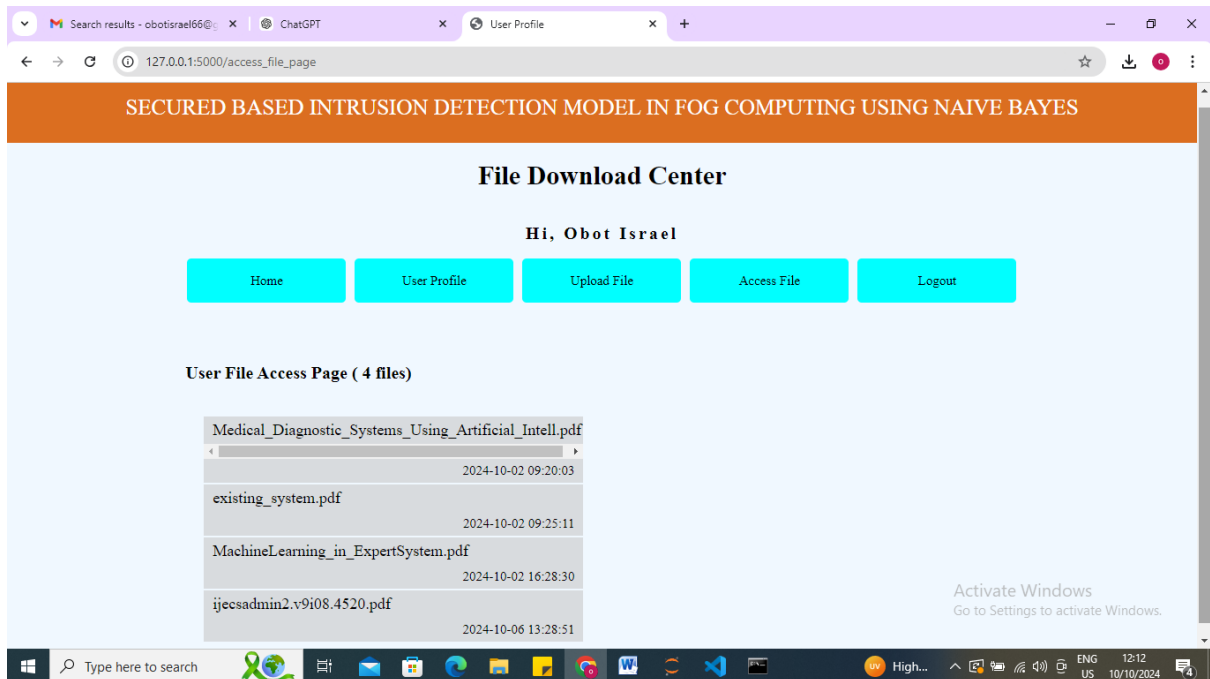


Figure 5 File download page

The User Policy Question Page served as a crucial security layer, requiring users to answer specific policy-related questions to gain access to uploaded files. This step ensured that only authorized individuals with the correct answers could view sensitive or protected documents. The page was designed with a user-friendly interface, presenting the

questions in a clear and structured format. Users could input their responses directly, and the system validated the answers in real-time. This feature reinforced data security while maintaining a seamless user experience. Figure 6 Policy Question Page

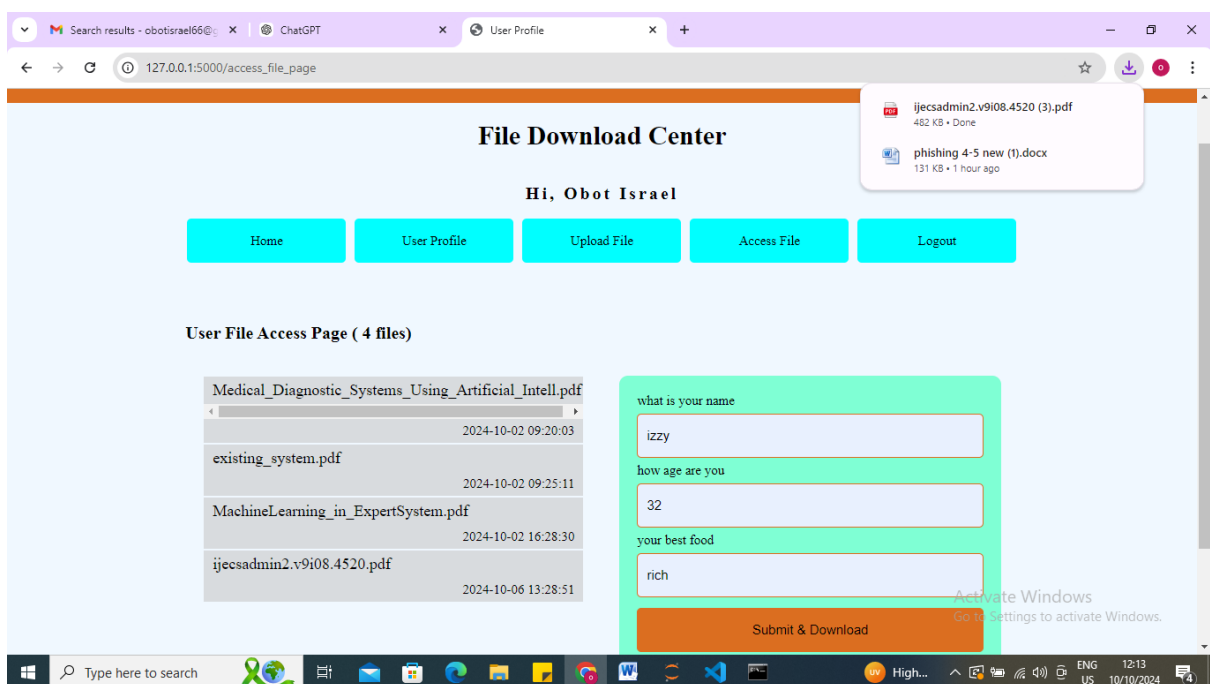


Figure 6 Policy Question Page

“Cloud-Based Intrusion Detection in Fog Computing Using Naïve Bayes Algorithm”

Once the user correctly answered the policy-related questions, the system granted access to the requested document. The content was displayed on a dedicated page, allowing users to interact with the file as intended. This functionality demonstrated the system's efficiency in

combining robust security with ease of access for authorized users. The ability to securely view files without compromising the integrity of the content highlighted the system's user-centric design. Figure 7 File Downing Page

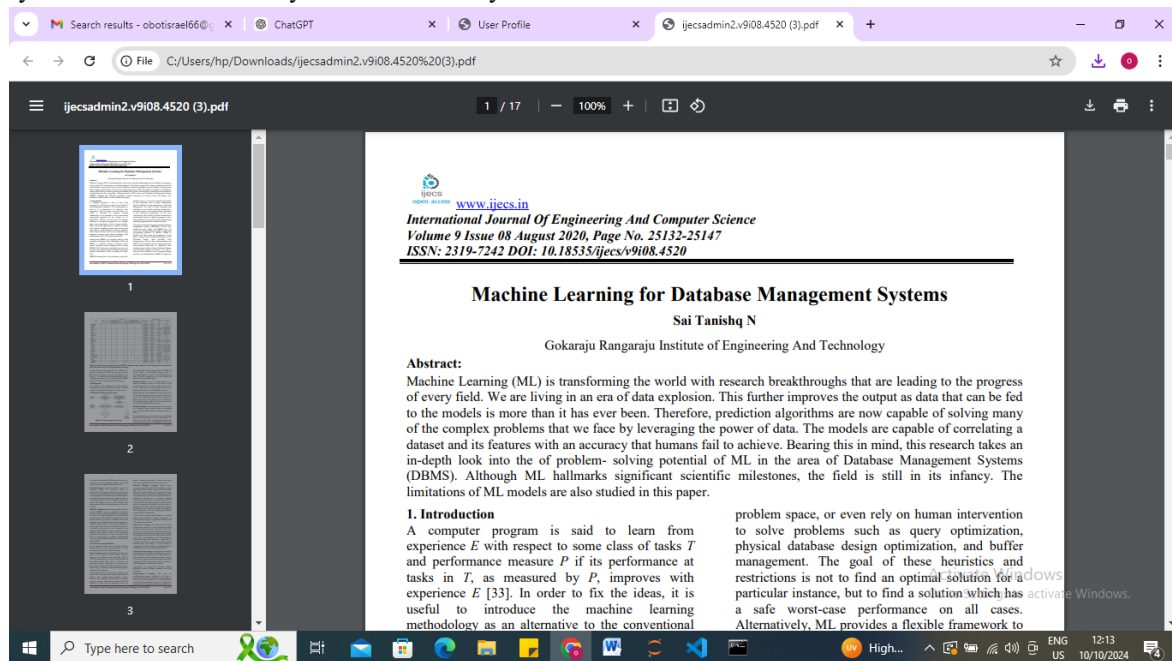


Figure 7 File Downing Page

In instances where users entered incorrect answers to the policy questions, the system denied access to the document. A notification was displayed, informing the user of the error and providing an opportunity to attempt the process again. This access restriction feature ensured that unauthorized users

could not bypass security protocols, emphasizing the importance of accuracy in responding to policy questions. The system's fail-safe mechanism was pivotal in protecting sensitive information from unauthorized access. Figure 8 File protection Page

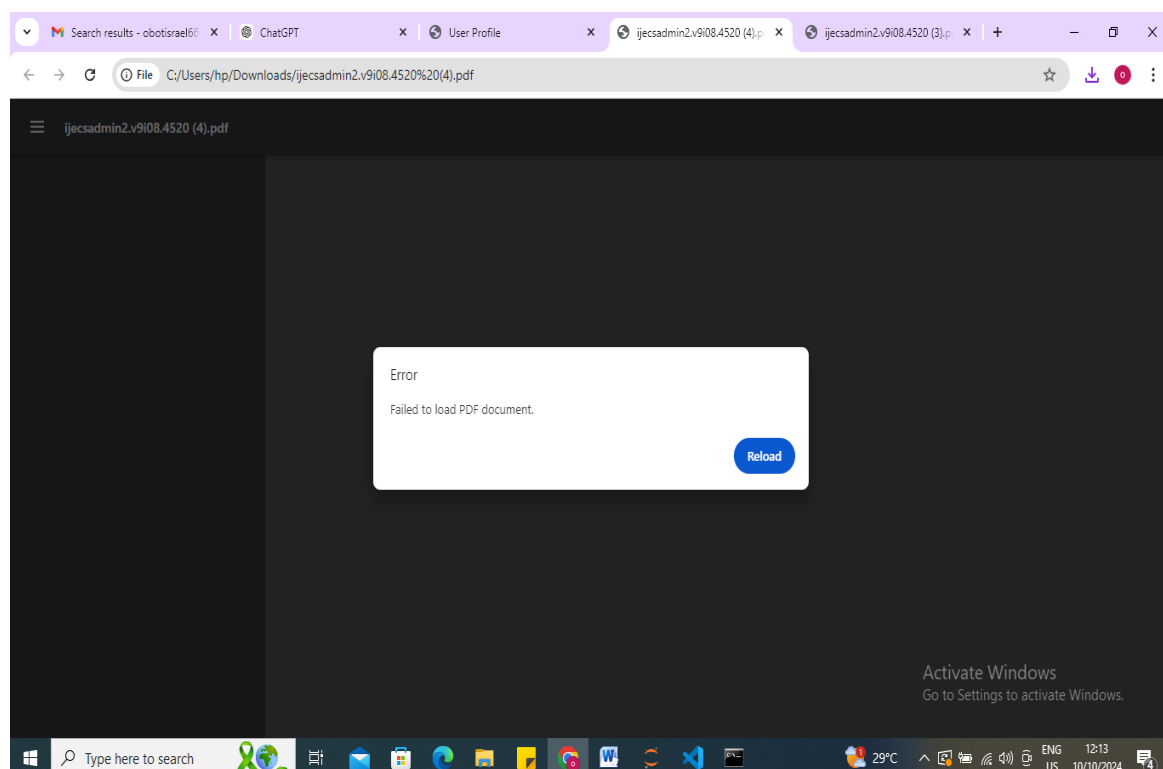


Figure 8 File protection Page

SUMMARY

The study developed a secured-based intrusion detection model for fog computing using the Naive Bayes algorithm. The primary goal was to improve the detection of malicious activities within fog computing environments, where data is processed closer to the source. Throughout the research, the model effectively classified network traffic into benign or malicious categories by analyzing patterns in the data. The dataset used was split into training and testing sets, where the Naive Bayes algorithm demonstrated strong predictive capabilities. After extensive testing, the model achieved an accuracy rate of 97%, which indicated its efficiency in distinguishing between normal and malicious activities. Additionally, the model was able to minimize false positives, thus reducing the number of incorrect alerts generated.

REFERENCES

1. Ahmed, A. (2023). *Data Security and Privacy Challenges in Cloud-Based Systems*. *Journal of Cloud Computing*, 12(4), 1–15.
2. Zhang, Y., Chen, M., & Li, X. (2022). *Security and Privacy in IoT-Cloud Integrated Environments*. *IEEE Internet of Things Journal*, 9(10), 7501–7513.
3. Alazab, M., Khan, S., & Gadekallu, T. R. (2023). *Anomaly Detection for Securing IoT Networks*. *Future Generation Computer Systems*, 138, 98–110.
4. Ahmed, A. (2023). Data security and privacy challenges in cloud-based IoT systems. *Journal of Cloud Computing*, 12(4), 1–15.
5. Zhang, Y., Chen, M., & Li, X. (2022). Privacy preservation in IoT-cloud integrated architectures. *IEEE Internet of Things Journal*, 9(10), 7501–7513.
6. Alazab, M., Khan, S., Gadekallu, T. R., & Pham, Q. V. (2023). Anomaly detection techniques for securing IoT networks. *Future Generation Computer Systems*, 138, 98–110.
7. Kumar, R., & Singh, P. (2021). Secure data storage framework for cloud-based IoT applications. *International Journal of Information Security*, 20(3), 341–355.
8. Norberg, P. A., Horne, D. R., & Horne, D. A. (2022). User privacy concerns in cloud-enabled IoT systems. *Computers & Security*, 115, 102607.
9. Fu, X., Wang, J., & Zhang, H. (2023). A secure access control mechanism for IoT-cloud environments. *Journal of Network and Computer Applications*, 206, 103442.
10. Marwick, A. E. (2023). Data privacy risks in emerging cloud technologies. *Information, Communication & Society*, 26(9), 1821–1837.
11. Lohachab, A., & Karambir, S. (2022). Network intrusion detection in IoT-cloud systems using deep learning. *Computer Communications*, 181, 124–135.