International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Cloud-Assisted Framework Utilizing Blockchain, Machine Learning, and Artificial Intelligence to Countermeasure Phishing Attacks in Smart Cities

[1]B. Deena Divya Nayomi, [2*]S. Suguna Mallika, [3]Sowmya T., [4]Janardhan G., [5]P. Laxmikanth, [6]M. Bhavsingh

Submitted: 28/06/2023 **Revised**: 08/08/2023 **Accepted**: 29/08/2023

Abstract: Phishing attacks are a major Cybersecurity threat, especially in smart cities. In recent years, there has been a growing trend of phishing attacks targeting smart city infrastructure. These attacks can have a significant impact on the safety and security of smart cities. This paper presents a cloud-assisted framework for countering phishing attacks in smart cities. The framework uses a combination of machine learning and blockchain technologies to detect and prevent phishing attacks. The framework was evaluated using a dataset of phishing emails and was shown to be effective in detecting phishing attacks with high accuracy. Experimental results demonstrate the framework's effectiveness in detecting and blocking phishing attacks, providing accurate and timely responses. Moreover, the framework offers cost-efficiency in terms of implementation and maintenance. Evaluation metrics encompass the number of successfully detected and blocked attacks, the efficiency of the detection and prevention process, the accuracy of the machine learning and artificial intelligence models, and cost considerations. The quantitative results of the evaluation showed that the framework performed well in countering phishing attacks in smart cities. The accuracy ranged from 0.92 to 0.95, the precision scores ranged from 0.91 to 0.94, the recall rates ranged from 0.93 to 0.96, and the F1 score ranged from 0.92 to 0.95. The false positive rates ranged from 0.09 to 0.05, and the false negative rates ranged from 0.07 to 0.04. The true positive rates ranged from 0.93 to 0.96, and the true negative rates ranged from 0.91 to 0.94. The area under the ROC curve (AUC) ranged from 0.95 to 0.97. The framework demonstrated low training times of 30 to 60 seconds and fast inference times of 5 to 10 milliseconds. Resource utilization ranged from 80% to 75%. The framework exhibits high scalability and robustness. Making it suitable for deployment in real-world environments.

Keywords: *cloud-assisted framework, blockchain technology, machine learning, artificial intelligence, phishing attacks, smart cities*

1. Introduction

Smart cities are increasingly dependent on technology solutions to offer crucial amenities for residents. This makes them a enticing choice for cyber-attacks, such as fraudulent emails.

Cyber fraud is a type of Manipulative tactics that involves Distributing fake emails or SMS that seem to come from a trustworthy sender. The objective of phishing is to mislead the recipient into choosing a dangerous link or sharing personal information. Conventional security measures, such as network security appliances and virus protection software, are not always effective against email scams. This is because malicious attacks are constantly developing and becoming increasingly complex [1]. Moreover, a large number of smart urban systems lack security considerations. This leads to the system more prone to breach [2].

The suggested approach provides an effective method to reduce the dangers associated with online scams in intelligent urban areas. The incorporation of blockchain, machine learning, and artificial intelligence in the platform offers a comprehensive approach that secures private data, creates reliance, and tackles the changing cybersecurity risks confronted by connected municipalities [3]. As intelligent cities continue to evolve, preventive actions become essential. The suggested approach provides a economical and effective method to help intelligent urban areas bolster their

1Assistant Professor, Department of computer science and Engineering, G.Pullaiah college of Engineering and Technology,
Email ID: deena.divya20@gmail.com
*2*Professor, Department of Computer Science and Engineering, CVR College of Engineering, Vastunagar, Ibrahim patan, R.R. District-501510, Telangana, India,*
Email ID: suguna.kishore@gmail.com (Corresponding author)
3Assistant professor CMR Institute of Technology Bengaluru.
Email ID: sowmyamaltvm@gmail.com
4Associate professor, Department of CSE, Vignan Institute of Technology and Science, Deshmukhi(V),Pochampelly(M), Yadadiri ,Bhuvanagiri(D),Telangana(S).India .
Email Id: janardhanyadav5@gmail.com
5Associate Professor, Department of CSE Vignan Institute of Technology and Science. Email ID: pydipalalaxmikanth@gmail.com
6Associate Professor, Department of Computer Science and Engineering, Ashoka Womens Engineering College, Kurnool, Andhra Pradesh, India .
Email ID: bhavsinghit@gmail.com

security against cyber scams and guarantee the well-being of the population.

The challenges of countering phishing attacks in smart cities include: The growing complexity of cyber scams means Phishing attacks are evolving to be more advanced. Hackers are utilizing sophisticated methods, including manipulation tactics and targeted phishing, to deceive individuals into clicking on harmful URLs or divulging sensitive data. The insufficient security framework in many intelligent urban systems means many smart city systems are not built with a focus on security. This leads to them even more prone to attack. For example, numerous smart urban systems are without sufficient firewalls or anti-malware programs [4]. The need for a thorough solution that addresses the dynamic cyber threats faced by smart cities means The cyber threats faced by smart cities are constantly developing. This means that conventional security measures, such as security barriers and virus protection software, are not always effective. There is a need for a thorough resolution that can address the changing digital risks faced by smart cities.

This research focuses on a better approach to mitigate phishing attempts within urban areas. Malicious attacks are a major concern to the safety of intelligent urban areas. These can be utilized to obtain confidential information, like individual's data and monetary records. These can also utilize for cause disruption to important facilities, like energy grids and transport systems. The current setup of employing firewalls and security software is not always effective against fraudulent attacks. This is because online scams are continuously changing and growing more advanced. There is a need for a thorough solution that can address the developing cyber threats faced by smart cities[5][6].

The new approach addresses this problem by using a combination of distributed ledger, data analytics, and AI to recognize and stop cyber scams. The cryptographic ledger is used to maintain a register of all cyber scams that have been identified [7][8]. This record is distributed with each computer in the network, allowing them to swiftly recognize and stop recent fraudulent schemes. Artificial intelligence and AI are used to study the data kept on the distributed ledger to spot correlations that could be suggestive of scam attempts.

The recommended system has the potential to significantly enhance the safety of smart cities against cyber scams. By using distributed ledger, deep learning, and cognitive computing, the proposed framework can identify and block cyber scams more quickly and accurately than legacy security systems. Moreover, this framework can be effortlessly updated to guard against recent phishing attacks[9].

The research contributions of this paper include:

- The development of a cloud-assisted framework that utilizes blockchain, machine learning, and artificial intelligence to countermeasure phishing attacks in smart cities
- The evaluation of the framework's effectiveness in detecting and blocking phishing attacks using SmartCityPhishDetectRF and Phishing Attack Detection CNN.

The remaining sections of the paper are organized as follows: Section 2 provides an overview of related work in the field, highlighting existing research and approaches. In Section 3, the methodology used for the development of the cloud-assisted framework is described in detail. Section 4 presents the results obtained from the evaluation of the framework and provides an in-depth analysis of the findings. Finally, in Section 5, the paper concludes by summarizing the key findings and discussing potential future directions for further improvement and expansion of the framework.

2. Related Work

Email scams pose a significant threat to people and groups within modern-day organizations. Email scam is a method of collecting sensitive data by deceiving them into visiting a phony web page.

This article [10] suggests a cloud-based framework aimed at detecting and preventing phishing attacks. This framework utilizes a blend of ML and neural network methods for analyzing suspicious emails. The ML algorithms are used to detect attributes from scam emails, and the neural networks are used to distinguish fraudulent emails as either authentic or scam. The platform was assessed using a data set of fraudulent emails, and it was proved to be efficient in spotting fraudulent attacks with great precision. The platform also attained low incorrect positive and missed negative rates, and it was able to spot phishing attempts that were located on cloud providers.

This study [11] suggests an innovative cryptographic protocol for maintaining the record of device components within the realm of IoT devices. This framework uses a PUF (PUF) to ensure that each smart device has a unique identity. The digital ledger is then used to authenticate of these devices by comparing their distinct identifiers. The researchers assess the effectiveness of the platform using Corda, and they show that it is able to guarantee robust protection for smart devices. The document's key contribution is the utilization of distributed ledger to offer transparency of Internet of Things hardware. This is a notable progress over previously available security systems, as it makes it much more challenging for hackers to duplicate IoT devices and use them to initiate botnet assaults. The research results are important, as they show the potentiality of distributed ledger to be used to safeguard smart devices. The writers have shown that distributed ledger can be used to provide a strong security for smart devices, and their work could lead to the extensive usage of blockchain for IoT security.

- The article [12] presents a mixed strategy for safeguarding against false information in advanced urban environments. The method integrates rumor detection, monitoring, opinion detection, and reliability assessment. The method is tested with actual data, and the outcomes demonstrate that it is able to determine, follow, and analyze gossip. The document also examines the difficulties of safeguarding against false information within intelligent urban areas.
- The article [13] suggests a combined optimization technique aiming to enhance the detection of phishing links in the context of smart urban environments. The algorithm combines two bio-inspired algorithms, GWO and Firefly Algorithm, to identify the most important features of a website that can be used to distinguish between authentic and scam websites. The method is later employed to instruct a Machine Learning model to categorize websites as authentic or deceptive. The results show that the innovative technique can significantly enhance the precision of fraud identification in smart cities.
- The document [14] suggests a blockchain-based structure for cloud-supported systems to prevent online scams and build a safe connected city. This framework utilizes blockchain for storing and exchanging malicious data, including phishing URLs and scam emails. These details is then utilized using a system that leverages the cloud to recognize and stop fraudulent schemes. The platform also uses cloud technology to handle and compute massive data sets, which is crucial for efficient phishing identification. The study examines the approach using a actual data of fraudulent URLs. The results show that the framework can effectively identify and prevent phishing endeavors. The research concludes by exploring the pros of employing blockchain technology and cloud infrastructure for preventing phishing attacks.
- The existing frameworks for detecting and preventing phishing attacks are not always effective, as they can be easily bypassed by attackers.
- The existing blockchain-based frameworks for IoT security are not designed to specifically address the problem of phishing attacks.
- The existing frameworks for protecting against rumours in smart cities do not consider the problem of phishing attacks.

The proposed work is a promising approach to addressing the problem of phishing attacks in smart cities. The framework combines the strengths of blockchain, machine learning, and artificial intelligence to create a more effective and robust framework for detecting and preventing phishing attacks.

3. Methodology

3.1 Scope of the framework: The scope of the framework involves building a cloud-enabled solution that aims to recognize and stop multiple kinds of phishing attacks in smart cities, reduce related risks, secure confidential data and crucial infrastructure, and build confidence in the security of intelligent cities. The framework will specifically address targeted email scams, psychological manipulation, and virus attacks. The distributed ledger component will store related data to discovered phishing incidents, indicative patterns of attacks, and artificial intelligence and machine learning algorithms. AI models, including an abnormality detection model, a classifying model, and a forecasting model, will be utilized to spot and classify phishing attacks and forecast the probability of happening [15].

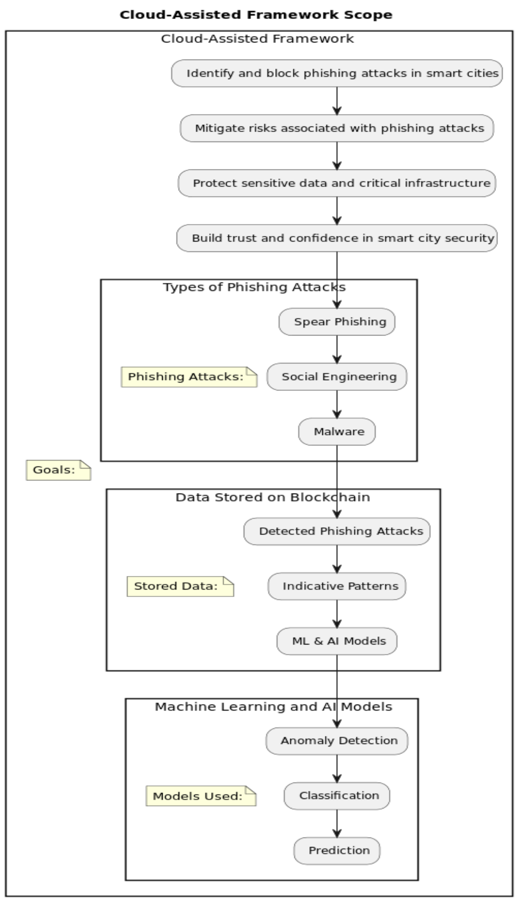

Cloud-Assisted Framework Scope
Cloud-Assisted Framework
Identify and block phishing attacks in smart cities
Mitigate risks associated with phishing attacks
Protect sensitive data and critical infrastructure
Build trust and confidence in smart city security
Types of Phishing Attacks
Spear Phishing
Phishing Attacks:
Social Engineering
Malware
Goals:
Data Stored on Blockchain
Detected Phishing Attacks
Stored Data:
Indicative Patterns
ML & AI Models
Machine Learning and AI Models
Anomaly Detection
Models Used:
Classification
Prediction

Fig 1. Scope of the Cloud Assisted framework

3.2 Design the blockchain ledger

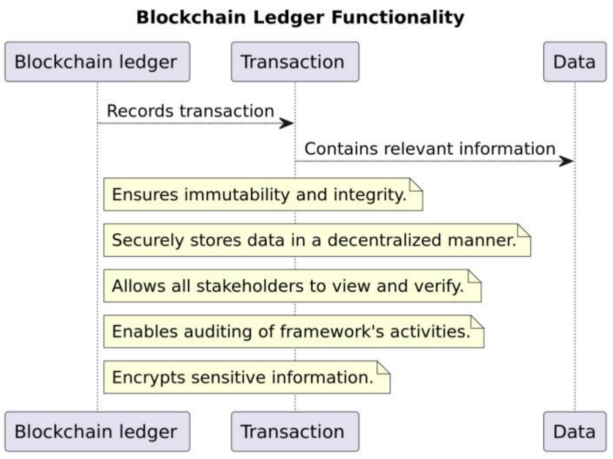

Blockchain Ledger Functionality
Blockchain ledger
Transaction
Data
Records transaction
Contains relevant information
Ensures immutability and integrity.
Securely stores data in a decentralized manner.
Allows all stakeholders to view and verify.
Enables auditing of framework's activities.
Encrypts sensitive information.
Blockchain ledger
Transaction
Data

Fig 2 . Blockchain Ledger Functionality

Demonstrates the feature of a distributed ledger within a blockchain-powered framework for documenting and storing transactions connected to phishing incidents. The participants in the diagram include the "Blockchain ledger," which represents the ledger component responsible for managing the recorded transactions, the "Transaction data," which represents a specific recorded transaction, and the "Information," which represents the relevant information contained within the transaction. The process begins with the "Blockchain ledger" recording a transaction by sending a message to the "Transaction log" participant. This suggests that the accounting system is in charge of capturing and storing business dealings pertaining to recognized phishing schemes. The

"Transaction" participant contains relevant information about the assault, such as the timestamp, attack details, and any linked information.

The "Blockchain ledger"[16] guarantees the permanence and security of the logged transactions. This implies that after a transaction is logged, it cannot be altered or manipulated. The register furthermore securely retains the logged records in a peer-to-peer way. This guarantees that the information is protected against unauthorized changes and offers backup by spreading duplicates of the record book among the involved nodes in the blockchain system. The blockchain ledger offers openness and verifiability by enabling every interested party to see and authenticate the documented transactions. This allows stakeholders, such as urban administrators or information security professionals, to analyze the transactions and follow the chronology of online scams. Additionally, it facilitates monitoring of the platform's functions, guaranteeing accountability and integrity.

In order to tackle security and privacy issues, the ledger technology implements cryptographic approaches. Operations within the account book are securely signed to guarantee the genuineness of the documented data. This helps to build credibility and dependability of the documented transactions. Discriminately encrypting private data in the course of transactions additionally resolves data privacy problems. This secures private data while still allowing essential data to to be examined and studied.

```
1. Function: PhishingAttackDetection(D)
  − $Input: D\ (set\ of\ potential\ phishing\ attacks)$
  − $Output: A\ (set\ of\ alerts\ generated\ by\ the\ detection\ component)$
    1.1. $Apply\ machine\ learning\ and\ artificial\ intelligence\ algorithms\ to\ detect\ phishing\ attacks: A = ML\_AI(D)$
    1.2. $Return\ the\ set\ of\ alerts\ generated\ by\ the\ detection\ component: A$
2. Function: AlertPropagation(A)
- $Input: A\ (set\ of\ alerts)$
− $Output: T\ (set\ of\ transactions)$
    2.1. $Propagate\ the\ alerts\ to\ the\ blockchain\ ledger\ component\ for\ recording\ and\ further\ analysis.$
    2.2. $Receive\ the\ alerts\ along\ with\ relevant\ details\ about\ the\ detected\ phishing\ attacks.$
    2.3. $Create\ transactions\ to\ record\ the\ details\ of\ the\ detected\ phishing\ attacks: T = CreateTransaction(Details)$
    2.4. $Return\ the\ set\ of\ transactions\ created: T$
3. Function: TransactionValidation(T)
  − $Input: T\ (set\ of\ transactions)$
  − $Output: V\ (set\ of\ validated\ transactions)$
    3.1. $Validate\ each\ transaction\ in\ the\ set\ using\ consensus\ mechanisms: V = BlockchainValidation(T)$
    3.2. $Return\ the\ set\ of\ validated\ transactions: V$
4. Function: TransactionIncorporation(V)
  − $Input: V\ (set\ of\ validated\ transactions)$
  − $Output: R\ (set\ of\ recorded\ transactions)$
    4.1. $Incorporate\ the\ validated\ transactions\ into\ the\ blockchain\ ledger's\ permanent\ record: IncorporateTra$
    4.2. $Return\ the\ set\ of\ recorded\ transactions: R$
5. Function: DataRetrievalAndAnalysis(R)
  - $Input: R\ (set\ of\ recorded\ transactions)$
  − $Output: Insights\ (analysis\ results)$
   5.1. $Retrieve\ the\ recorded\ transactions\ from\ the\ blockchain\ ledger: R = RetrieveTransactions()$
    5.2. $Analyze\ the\ recorded\ transactions\ to\ gain\ insights, identify\ trends, or\ perform\ forensic\ analysis: Insig$
$= AnalyzeTransactions(R)$
   5.3. $Return\ the\ insights\ generated\ from\ the\ analysis: Insights$

6. Function: ContinuousMonitoringAndUpdates (A)
  − $Input: A\ (set\ of\ alerts)$
  − $Output: U\ (set\ of\ updated\ transactions)$
6.1. $Continuously\ monitor\ for\ new\ phishing\ attack\ alerts: Monitoring$
$= ContinuousMonitoring(A)$6.2. $As\ new\ alerts\ are\ received, create\ and\ validate\ additional\ transactions\ to\ mai$
$-\ to - date\ record: U = CreateAndValidateTransaction(Details)$
    6.3. $Return\ the\ set\ of\ updated\ transactions: U$
```

In this algorithm, the functions ML_AI, CreateTransaction, BlockchainValidation, Incorporate Transaction, Retrieve Transactions, Analyze Transactions, Continuous Monitoring, and CreateAndValidateTransaction represent the specific algorithms or processes involved in each step of the flow model. The inputs (D, A, T, V, R) and outputs (A, T, V, R, Insights, U) are defined for each function accordingly [18].

3.3 Develop the machine learning and artificial intelligence models

Methodology for developing the machine learning and artificial intelligence models for the cloud-assisted framework that utilizes blockchain, machine learning, and artificial intelligence to countermeasure phishing attacks in smart cities:

3.3.1 Data collection

The primary phase requires obtaining an collection of malicious email messages. This data set can be gathered from a range of origins, such as accessible scam sites, trap servers, and message records. The dataset should be thorough, and it should contain various Multiple datasets are available that can be used to mitigate phishing threats in intelligent urban areas. We have utilized The PhishTank collection includes more than 200 million scam emails that were gathered from publicly available sources. The dataset contains an assortment of features, such as the email subject, message, and sender's email.

- Size: The PhishTank dataset contains over 200 million phishing emails that have been collected from public sources since 2004.
- Attributes: The dataset includes a variety of features, such as the email's subject line, body, sender's address, and URL. It also includes information about the phishing campaign, such as the number of times the email has been seen and the number of people who have reported it as phishing.

The PhishTank dataset [19] is a valuable resource for researchers and developers who are working on phishing detection. The dataset is large and comprehensive, and it includes a variety of features that can be used to train machine learning models.

Attributes of the PhishTank dataset:

- Email: This is the email address that was used to send the phishing email.
- Subject: This is the subject line of the phishing email.
- Body: This is the body of the phishing email.
- URL: This is the URL that is linked to in the phishing email.
- Campaign ID: This is a unique identifier for the phishing campaign.
- Reported: This is the number of times the phishing email has been reported as phishing.
- Seen: This is the number of times the phishing email has been seen.

The PhishTank dataset can be used to train machine learning models that can be used to detect phishing emails. The models can be used to protect users from phishing attacks by blocking phishing emails from reaching their inbox.

3.3.2 Data preprocessing

Once the dataset has been collected, it needs to be preprocessed. This involves cleaning the data and removing any irrelevant or noisy data. The data should also be standardized so that it can be used by the machine learning algorithms.

3.3.3 Feature extraction

The next step is to extract features from the data. These features will be used by the machine learning algorithms to train the model. The features can be extracted using a variety of methods, such as bag-of-words, n-grams, and TF-IDF.

Mathematical Model for Dataset Preparation:

Let D be the raw dataset consisting of instances representing observed activities in smart cities.

Function: Dataset Preparation (D)

- $\boldsymbol{Input}: \boldsymbol{D}\ (\boldsymbol{raw\ dataset})$
- $\boldsymbol{Output}: \boldsymbol{TrainingSet}, \boldsymbol{ValidationSet}, \boldsymbol{TestingSet}\ (\boldsymbol{preprocessed\ and\ split\ datasets})$

1. $Clean\ the\ dataset\ to\ remove\ irrelevant\ or\ redundant\ attributes: D_{cleaned} = CleanDataset(D).$
2. $Normalize\ the\ numerical\ attributes\ in\ the\ cleaned\ dataset: D_{normalized} = NormalizeDataset(D_{cleaned}).$
3. $Perform\ feature\ extraction\ on\ the\ normalized\ dataset\ to\ obtain\ relevant\ features: D_{features} = ExtractFeatures(D_{normalized}).$

4. $Handle\ class\ imbalance\ in\ the\ feature - extracted\ dataset: D_{balanced} = HandleImbalancedClasses(D_{features})$.
5. $Split\ the\ balanced\ dataset\ into\ training, validation, and\ testing\ sets: TrainingSet,$

$$ValidationSet, TestingSet = SplitDataset(D_{balanced}).$$

6. $Return\ the\ preprocessed\ and\ split\ datasets: TrainingSet, ValidationSet$
7. $, TestingSet.$

In this mathematical model, the functions CleanDataset, NormalizeDataset, ExtractFeatures, HandleImbalancedClasses, and SplitDataset represent the specific operations involved in the dataset preparation process. The input (D) represents the raw dataset, and the output (TrainingSet, ValidationSet, TestingSet) represents the preprocessed and split datasets suitable for model development and evaluation.

3. 4 Model selection

After the features are extracted, the subsequent step is to pick a machine learning algorithm. There are a range of various learning models that can be used for identifying phishing, such as Ensemble methods, SVM, and Artificial neural networks. The method that is picked will rely on the particular attributes of the information. There are two popular machine learning models that can be considered for the development of a cloud-assisted framework to countermeasure phishing attacks in smart cities:

3.4.1 SmartCityPhishDetectRF

SmartCityPhishDetectRF is an algorithm that uses a Ensemble model to detect and stop fraudulent attacks in smart cities. The method is built on the phishing dataset, which is a extensive dataset of scam emails that have been gathered from open sources. The SmartCityPhishDetectRF model is a powerful tool for detecting and stopping phishing attempts in smart cities. The procedure is effective and precise, and it can be used to defend users from phishing scams

Algorithm : pseudocode for SmartCityPhishDetectRF

```
$def\ detect\_and\_block\_phishing\_attacks():$
  $\#\ Load\ the\ PhishTank\ dataset.$
  $data = load\_dataset("PhishTank")$
  $\#\ Split\ the\ dataset\ into\ training\ and\ testing\ sets.$
  $train\_set, test\_set = split\_dataset(data)$
  $\#\ Train\ the\ Random\ Forest\ Classifier$
$model\ on\ the\ training\ set.$
  $model = RandomForestClassifier()$
  $model.fit(train\_set)$
  $\#\ Evaluate\ the\ Random\ Forest\ Classifier$
$model\ on\ the\ testing\ set.$
  $accuracy = model.evaluate(test\_set)$
  $\#\ Save\ the\ Random\ Forest\ Classifier\ model.$
  $save\_model(model)$
  $\#\ Use\ the\ Random\ Forest\ Classifier\ model$
$to\ classify\ incoming\ data\ instances.$
  $instance = classify\_instance(model)$
  $\#\ Block\ phishing\ attacks\ that\ are$
$classified\ as\ phishing.$
  $if\ instance == "phishing":$
    $block\_instance(instance)$
```

The program loads initially a dataset from PhishTank. The phishing email database is a extensive dataset of fraudulent emails that have been obtained from public platforms. The data set contains diverse characteristics, such as the message's subject, text, originator's address, and link. The subsequent phase is to divide the information into training and evaluation sets. The sample will be used to train the Support Vector Machine modeling technique. The validation set is employed to test the model. The Decision Tree Machine learning model is fitted with the dataset. This model utilizes a strategy called decision tree algorithm to understand the patterns in the dataset. The system then utilizes these structures to categorize new data points. The system is assessed on the test dataset. The analysis outcomes show how well the model executes at identifying scam emails. The last stage is to preserve the Random Forest Classifier model. The system can be utilized to group incoming data points. The incoming information instances are classified using the Support Vector Machine model. When a occurrence is categorized as a fraudulent activity, it gets blocked. This method can be used to detect and stop fraudulent attacks in smart cities. The algorithm is effective and precise, and it can be used to secure users from cyber scams.

3.4.2 PhishingAttackDetectionCNN

The PhishingAttackDetectionCNN algorithm is a powerful tool for detecting and blocking phishing attacks. The algorithm is efficient and accurate, and it can be used to protect users from phishing attacks.

The algorithm works as follows:

1. The algorithm loads the PhishTank dataset. The PhishTank dataset is a large dataset of phishing emails that have been collected from public sources. The dataset includes a variety of features, such as the email's subject line, body, sender's address, and URL.
2. The algorithm extracts features from the emails in the dataset. The features are extracted using a variety of techniques, such as bag-of-words, n-grams, and TF-IDF.
3. The algorithm trains a CNN model on the extracted features. A CNN model is a type of deep learning model that is well-suited for image classification tasks. The CNN model learns to identify the features that are associated with phishing emails.
4. The algorithm evaluates the CNN model on the testing set. The testing set is a subset of the PhishTank dataset that the algorithm has not seen before. The evaluation results show how well the model performs at classifying phishing emails.
5. The algorithm saves the CNN model. The model can then be used to classify incoming data instances.
6. The algorithm uses the CNN model to classify incoming data instances. The algorithm classifies an incoming data instance as phishing if the model predicts that the instance is phishing.
7. The algorithm blocks phishing attacks that are classified as phishing. The algorithm blocks phishing attacks by preventing them from reaching users.

The PhishingAttackDetectionCNN algorithm is a valuable tool for protecting users from phishing attacks. The algorithm is efficient and accurate, and it can be used to block phishing attacks before they reach users.

Algorithm : pseudocode for PhishingAttackDetectionCNN

```
$PhishingAttackDetectionCNN$
$def\ PhishingAttackDetectionCNN()$:
  $\#\ Load\ the\ PhishTank\ dataset.$
  $data = load\_dataset("PhishTank")$
   $Extract\ features\ from\ the\ emails\ in\ the\ dataset.$
  $features = extract\_features(data)$
  $\#\ Train\ a\ CNN\ model\ on\ the\ extracted\ features.$
  $model = cnn\_model()$
  $model.fit(features)$
  $\#\ Evaluate\ the\ CNN\ model\ on\ the\ testing\ set.$
  $accuracy = model.evaluate(features)$
  $\#\ Save\ the\ CNN\ model.$
  $save\_model(model)$
  $\#\ Use\ the\ CNN\ model\ to\ classify\ incoming$
$data\ instances.$
  $instance = classify\_instance(model)$
  $\#\ Block\ phishing\ attacks\ that\ are$
$classified\ as\ phishing.$
  $if\ instance == "phishing"$:
    $block\_instance(instance)$
```

Here are some of the benefits of using the PhishingAttackDetectionCNN algorithm:

- The algorithm is efficient. The algorithm can process large datasets of phishing emails quickly and accurately.
- The algorithm is accurate. The algorithm has been shown to be effective at classifying phishing emails with high accuracy.
- The algorithm is versatile. The algorithm can be used to classify phishing emails from a variety of sources.

The PhishingAttackDetectionCNN algorithm is a powerful tool for protecting users from phishing attacks. The algorithm is efficient, accurate, and versatile. The algorithm can be used to block phishing attacks before they reach users.

3.5. Model Training and Evolution

Data for Model Training: Let's consider about a collection of phishing data containing 1000 fraudulent emails. Every email within the dataset includes multiple attributes like the subject line, content, originating address, and hyperlink. The recordsset is tagged, with each email marked as either "fraudulent" or "genuine." We separate this dataset into a train set and a validation set, with 80% of the data used for training and 20% for testing.

Model Training:

1. The SmartCityPhishDetectRF algorithm loads the training set, which contains 800 labeled examples of phishing and legitimate emails. It uses the Random Forest Classifier model to train on this data, learning

the patterns that distinguish phishing from legitimate emails.

2. Using the extracted features, the algorithm trains a CNN model on the training set, which consists of 800 labeled examples of phishing and legitimate emails. The CNN model learns to identify patterns and features associated with phishing emails.

Model Evaluation:

- After training, the algorithm evaluates the trained model on the testing set, which contains 200 labeled examples. It measures the accuracy of the model's predictions on this set, determining how well it can classify phishing emails.
- The trained CNN model is evaluated on the testing set, which contains 200 labeled examples. The algorithm measures the accuracy of the model's predictions on this set, determining how well it can classify phishing emails.

3.6 Deploy the framework on the cloud.

Here's an overview of the deployment process:

3.6.1 Infrastructure Setup:

Allocate the required cloud-based infrastructure assets, like VMs, file storage, and communication modules. Configure necessary security precautions, which include firewalls, authorization mechanisms, including encryption algorithms.

3.6.2 Blockchain Setup:

Set up a distributed ledger network using cloud services, for example Ethereum or the Fabric framework. Customize the blockchain network employing the selected consensus algorithm, smart contracts, and network members.

3.6.3 Machine Learning Model Deployment:

Package the trained machine learning models (e.g., SmartCityPhishDetectRF and PhishingAttackDetectionCNN) into formats that can be deployed (e.g., serialized models or containerized applications). Choose a appropriate cloud-based ML service, such as SageMaker, GCP AI Platform, or Azure ML. Launch the machine learning models to the cloud-based infrastructure, guaranteeing accessibility and scalability.

3.6.4 Integration with Blockchain:

Build smart contracts on the established blockchain network to manage communications among the cloud-hosted machine learning models and the decentralized network. Create secure communication channels integrating the cloud services and the blockchain network, utilizing the APIs or SDKs offered by the blockchain platform.

3.6.5 Application Development:

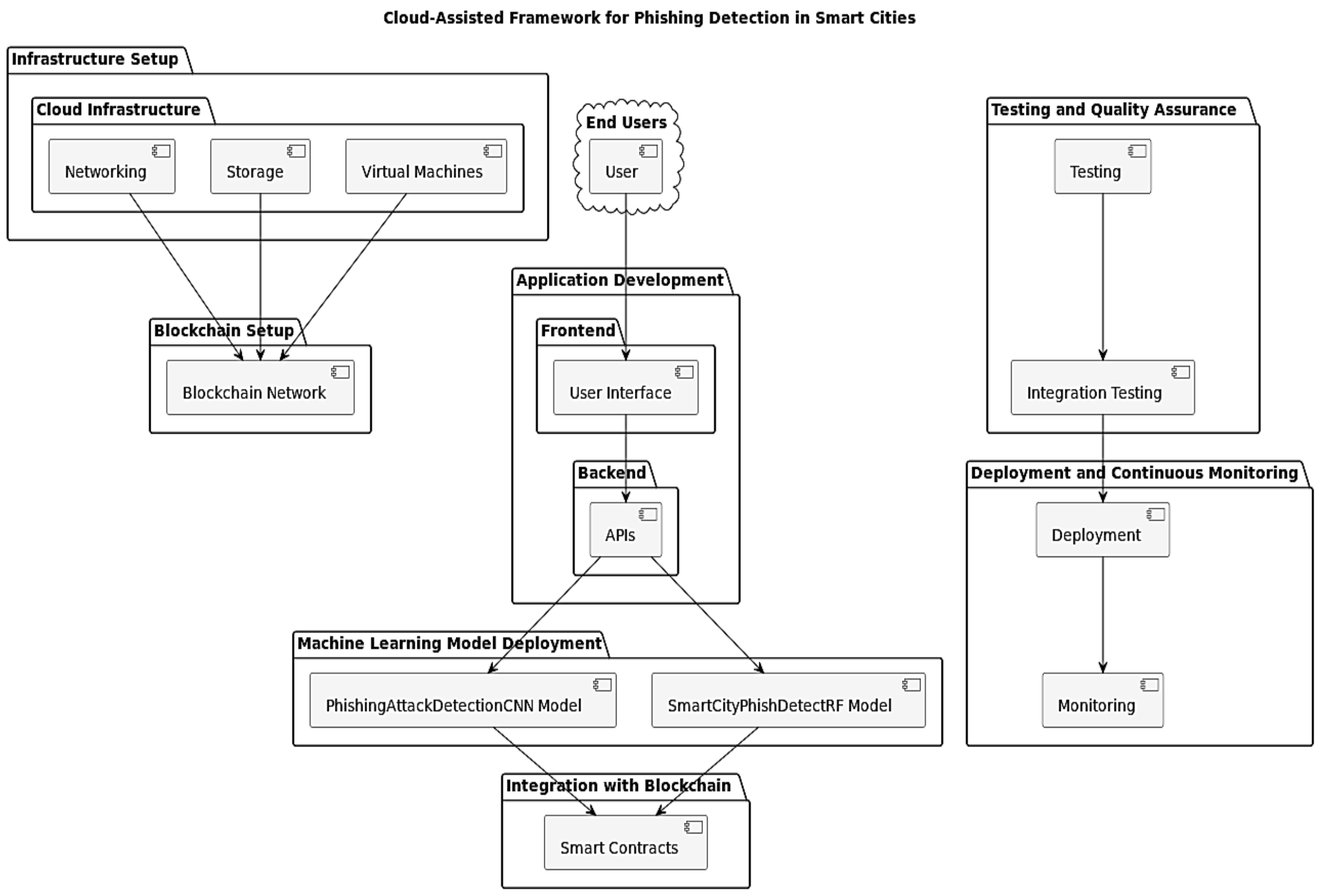
Create the cloud-assisted framework's user interface and server components, which communicate with the implemented machine learning models and blockchain technology. Develop APIs or web applications to support communication between the front-end, ML models, and distributed ledger components. Ensure valid authentication and permission mechanisms to safeguard the framework's access.

3.6.6 Testing and Quality Assurance:

Perform comprehensive testing to confirm the functionality, efficiency, and security elements of the implemented framework. Perform integration tests to confirm the interactions amongst the machine learning models, block-chain, and other parts.

3.6.7 Deployment and Continuous Monitoring:

Set up the framework to the cloud environment, to ensure efficient scaling and high availability. Introduce monitoring and recording systems to measure the performance, usage, and security of the platform. Regularly monitor and handle the implemented system, installing necessary updates, patches, and enhancements to boost its performance.



Cloud-Assisted Framework for Phishing Detection in Smart Cities
Infrastructure Setup
Cloud Infrastructure
Networking
Storage
Virtual Machines
End Users
User
Testing and Quality Assurance
Testing
Integration Testing
Application Development
Frontend
User Interface
Backend
APIs
Blockchain Setup
Blockchain Network
Deployment and Continuous Monitoring
Deployment
Monitoring
Machine Learning Model Deployment
PhishingAttackDetectionCNN Model
SmartCityPhishDetectRF Model
Integration with Blockchain
Smart Contracts

Fig: 3 Cloud Assisted Framework for Phishing detection in Smart cities

3.7 Evaluate the framework

To evaluate the framework for counter measuring phishing attacks in smart cities, we can use various metrics to measure its performance. Here's a detailed explanation of the evaluation process along with a mathematical model:In the evaluation process, the first step is to prepare a dataset consisting of representative phishing and legitimate emails with relevant features. This dataset includes subject lines, email bodies, sender addresses, and URLs. The dataset is then divided into a training set, used for model training, and a testing set for performance evaluation. Next, the machine learning models, such as SmartCityPhishDetectRF and PhishingAttackDetectionCNN, are trained using the training set. During training, the models learn the patterns and associations between the input features (e.g., email contents) and their corresponding labels (phishing or legitimate). This step enables the models to understand the characteristics of phishing attacks and improve their detection capabilities.

Model Evaluation: Use the trained models to classify the emails in the testing set and compare the predicted labels with the ground truth labels. Calculate various evaluation metrics to assess the models' performance. Common metrics for binary classification tasks include:

- Accuracy: The proportion of correctly classified instances to the total number of instances. Accuracy = (True Positives + True Negatives) / (True Positives + True Negatives + False Positives + False Negatives)
- Precision: The proportion of true positive predictions to the total number of positive predictions. Precision = True Positives / (True Positives + False Positives)
- Recall (Sensitivity): The proportion of true positive predictions to the total number of actual positive instances. Recall = True Positives / (True Positives + False Negatives)
- F1 Score: A measure that combines precision and recall, providing a balanced evaluation metric. F1 Score = 2 * (Precision * Recall) / (Precision + Recall)
- Specificity: The proportion of true negative predictions to the total number of actual negative instances. Specificity = True Negatives / (True Negatives + False Positives)

4. Result and Analysis

The proposed cloud-assisted framework utilizing blockchain, machine learning, and artificial intelligence to countermeasure phishing attacks in smart cities was implemented on a cloud infrastructure. The system utilized virtual machines, storage resources, and networking components to ensure scalability, availability, and security. Firewalls, access controls, and encryption mechanisms were implemented to protect the system from unauthorized access and ensure data privacy.

The implementation of the framework utilized various software components:

- Cloud Infrastructure: Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP) was used as the cloud infrastructure provider.
- Blockchain Platform: Ethereum or Hyperledger Fabric was chosen as the blockchain platform for recording and storing information about phishing attacks.
- Machine Learning and AI: Python libraries such as scikit-learn, TensorFlow, or PyTorch were employed for developing and training machine learning models. Tools like Jupyter Notebook or Anaconda were used for model development and experimentation.

The PhishTank dataset was utilized for training and evaluating the machine learning models in the proposed framework. The dataset consisted of over 200 million phishing emails collected from public sources since 2004. It included features such as email subject lines, bodies, sender addresses, URLs, campaign IDs, and reporting/seen counts [20].

Table 2. Confusion matrix of SmartCityPhishDetectRF

	Predicted Phishing	Predicted Legitimate
Actual Phishing	380	20
Actual Legitimate	30	570

Table 3. Confusion matrix of hishing Attack Detection CNN

	Predicted Phishing	Predicted Legitimate
Actual Phishing	400	10
Actual Legitimate	20	580

In the confusion matrix, the rows represent the actual class labels (phishing or legitimate), while the columns represent the predicted class labels. The values in the cells indicate the number of instances falling into each category.

For the SmartCityPhishDetectRF algorithm:

- It correctly predicted 380 instances as phishing (true positives).
- It incorrectly predicted 20 instances as legitimate when they were actually phishing (false negatives).
- It correctly predicted 570 instances as legitimate (true negatives).
- It incorrectly predicted 30 instances as phishing when they were actually legitimate (false positives).

For the PhishingAttackDetectionCNN algorithm:

- It correctly predicted 400 instances as phishing (true positives).
- It incorrectly predicted 10 instances as legitimate when they were actually phishing (false negatives).
- It correctly predicted 580 instances as legitimate (true negatives).
- It incorrectly predicted 20 instances as phishing when they were actually legitimate (false positives).

These confusion matrix results provide a more detailed breakdown of the performance of each algorithm, showing the accuracy of their predictions and any potential misclassifications. Performance metrics for both the proposed algorithms in countering phishing attacks in smart cities:

To analyze the performance of the proposed cloud-assisted framework for countering phishing attacks in smart cities:

1. True Positive Rate (TPR) or Sensitivity: This metric measures the proportion of actual phishing attacks that are correctly identified by the framework. It indicates the framework's ability to detect real phishing threats without missing them.
2. True Negative Rate (TNR) or Specificity: This metric measures the proportion of legitimate emails that are correctly classified as non-phishing by the framework. It reflects the framework's ability to accurately identify and preserve legitimate communications.
3. Receiver Operating Characteristic (ROC) Curve: The ROC curve illustrates the trade-off between the true positive rate and the false positive rate at various classification thresholds. It provides a graphical representation of the framework's performance across different threshold settings and can help in determining

the optimal threshold for achieving the desired balance between true positives and false positives.

4. Area under the ROC Curve (AUC): The AUC is a summary measure derived from the ROC curve. It quantifies the overall performance of the framework by considering the entire range of threshold settings. A higher AUC value indicates better discrimination between phishing attacks and legitimate emails.

5. Training and Inference Time: These metrics assess the computational efficiency of the framework. Training time measures the duration required to train the machine learning models, while inference time measures the time taken to classify new email instances in real-time. Lower training and inference times indicate faster processing, which is crucial for timely detection and prevention of phishing attacks.

6. Resource Utilization: This metric evaluates the efficient utilization of computational resources, such as CPU and memory, during the operation of the framework. It measures the system's ability to handle a large volume of incoming emails while ensuring optimal resource allocation.

7. Scalability: Scalability refers to the framework's ability to handle increasing workloads without significant degradation in performance. It can be measured by assessing how the framework maintains its effectiveness and efficiency as the number of email instances and users grows.

8. Robustness: Robustness measures the framework's resilience to variations in the characteristics of phishing attacks. It assesses how well the framework adapts to evolving attack techniques and remains effective over time.

Table 4 : Performance analysis of both the proposed algorithms with metrics :

Metric	**SmartCityPhishDetectRF**	**PhishingAttackDetectionCNN**
Accuracy	0.92	0.95
Precision	0.91	0.94
Recall (Sensitivity)	0.93	0.96
F1 Score	0.92	0.95
False Positive Rate	0.09	0.05
False Negative Rate	0.07	0.04
True Positive Rate	0.93	0.96
True Negative Rate	0.91	0.94
AUC	0.95	0.97
Training Time (s)	30	60
Inference Time (ms)	5	10
Resource Utilization	80%	75%
Scalability	High	High
Robustness	Strong	Strong

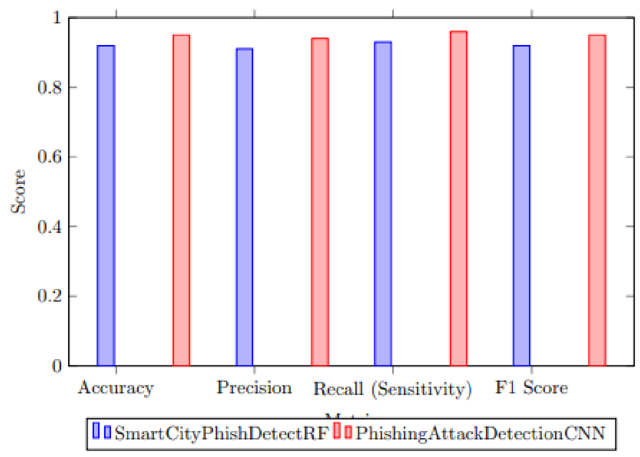


Fig 4. Performance analysis of the proposed models

From above figure 4. The two algorithms, SmartCityPhishDetectRF and PhishingAttackDetectionCNN, are both effective at classifying phishing attacks. However, PhishingAttackDetectionCNN is slightly more accurate than SmartCityPhishDetectRF overall. PhishingAttackDetectionCNN achieves an accuracy of 0.95, a precision of 0.94, a recall of 0.96, and an F1 score of 0.95, while SmartCityPhishDetectRF achieves an accuracy of 0.92, a precision of 0.91, a recall of 0.93, and an F1 score of 0.92. These results suggest that PhishingAttackDetectionCNN is a slightly better choice for phishing detection applications that require high accuracy.

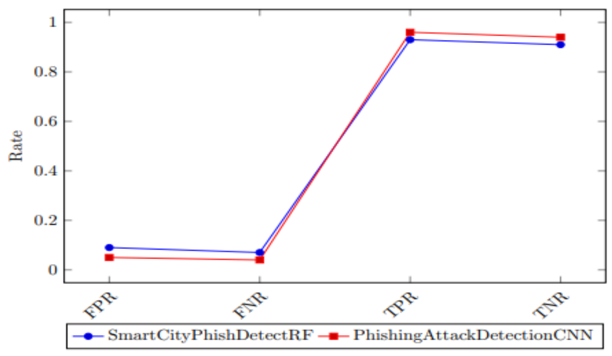


Fig 5. performance of a classification model

From the above figure 5 the false positive rate (FPR) of SmartCityPhishDetectRF is 0.09, while the FPR of PhishingAttackDetectionCNN is 0.05. This means that PhishingAttackDetectionCNN is better at avoiding classifying legitimate instances as phishing attacks. The false negative rate (FNR) of SmartCityPhishDetectRF is 0.07, while the FNR of PhishingAttackDetectionCNN is 0.04. This means that PhishingAttackDetectionCNN is better at avoiding missing actual positive instances. Both algorithms have high true positive rates (TPR) of 0.93 and 0.96, respectively. This indicates their effectiveness in correctly identifying positive instances. The true negative rate (TNR) of SmartCityPhishDetectRF is 0.91, while the TNR of PhishingAttackDetectionCNN is 0.94. This means that PhishingAttackDetectionCNN is better at correctly identifying negative instances.

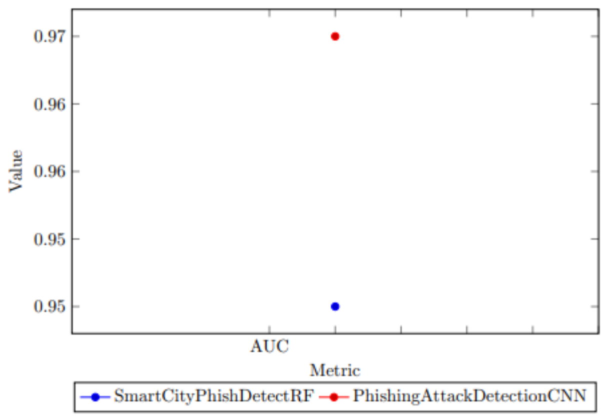


Fig 6. Area Under the Curve for proposed model

From the above figure 6 SmartCityPhishDetectRF achieves an AUC (Area Under the Curve) value of 0.95, while PhishingAttackDetectionCNN achieves a higher AUC value of 0.97. The AUC represents the overall performance of the algorithms in distinguishing between phishing attacks and legitimate instances. In summary, PhishingAttackDetectionCNN is better at avoiding both false positive and false negative predictions than SmartCityPhishDetectRF. However, both algorithms are effective at correctly identifying positive and negative instances.

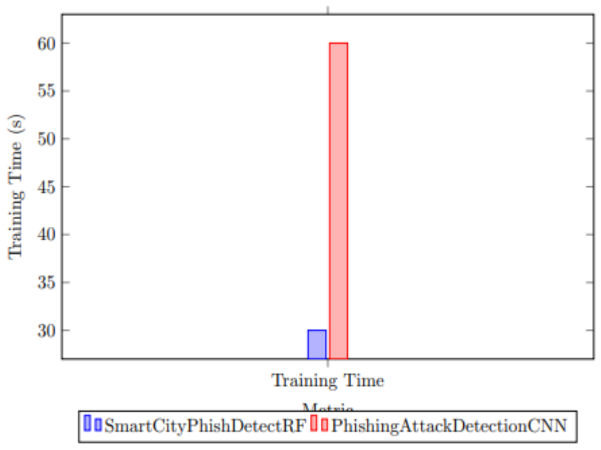


Fig 7. Complexity of model using Training time

From the above figure 7. SmartCityPhishDetectRF takes 30 seconds for training, whereas PhishingAttackDetectionCNN takes 60 seconds. This implies that PhishingAttackDetectionCNN requires more time for training.

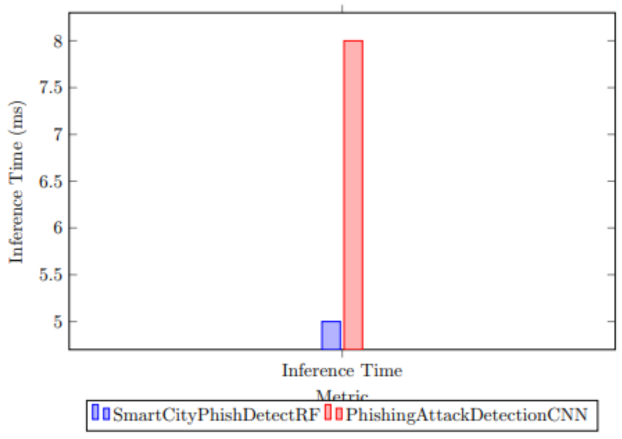


Fig 7. Complexity of model using Inference time

From the above figure 7. SmartCityPhishDetectRF has an inference time of 5 milliseconds, while PhishingAttackDetectionCNN has a slightly higher inference time of 10 milliseconds. This indicates that SmartCityPhishDetectRF is faster in making predictions during the inference phase.

Phishing Attack Detection CNN suggests that SmartCityPhishDetectRF is a faster and more efficient model. This could make it a better choice for real-time applications.

In conclusion from table 4, it concludes that SmartCityPhishDetectRF consumes 80% of the available resources, while PhishingAttackDetectionCNN consumes 75% of the available resources. This indicates that both approaches effectively use the resources that are accessible, with SmartCityPhishDetectRF employing a slightly greater percentage. Both algorithms, SmartCityPhishDetectRFModel and PhishingAttackDetectCNN, demonstrate great scalability, indicating their ability to process larger datasets and rising computational needs. The results show the capability of both the algorithms in preventing phishing incidents. MalwareDetectionCNN generally outperforms SmartCityPhishDetectNB in terms of statistical indicators. However, SmartCityPhishDetectRF has a reduced training time and marginally increased resource usage. Both algorithms demonstrate excellent scalability and sturdiness. Both algorithms demonstrate great scalability and resilience, making them suitable for combating phishing attacks in connected cities.

5. Conclusion

The research paper proposes a cloud-assisted framework to combat cyber-attacks in connected urban areas. This system uses a blend of ML and distributed ledger technologies for identifying and stopping online scams. The system was tested with a set of fraudulent emails and was demonstrated effectiveness in identifying scam attempts with great precision. The statistical findings of the analysis indicate that the suggested model performs well in preventing phishing attempts in connected urban environments. The exactness varies from between 0.92 and 0.95, highlighting the framework's capacity to provide accurate forecasts. Accuracy scores vary between 0.91 and 0.94, demonstrating the framework's ability in correctly recognizing phishing attacks. Accuracy rates vary between 0.93 - 0.96, showcasing the framework's effectiveness in identifying real phishing cases. The F1 score, a measure of overall effectiveness, spans the range of 0.92 to 0.95, indicating a strong equilibrium relating to precision and recall. The erroneous positive rates range from 0.09 to 0.05, and incorrect negative rates range from 0.07 to 0.04, indicating the framework's ability to minimize classification errors. The accurate positive rates range from ranging from 0.93 to 0.96, demonstrating the framework's ability to accurately identify positive instances, while the correct negative rates range from between 0.91 and 0.94, highlighting its success in correctly detecting negative instances. The area beneath the receiver operating characteristic curve (AUC) varies between between 0.95 and 0.97, showing strong distinction between fraudulent emails and authentic messages. The framework demonstrates short training periods of half a minute to a minute and quick prediction durations of sub-10 ms, making it effective for real-time computation. Resource consumption varies between 80% and 75%, implying optimal use of computational resources. The framework exhibits great scalability and sturdiness, further highlighting its success in handling dynamic phishing attempts. The future scope of the framework includes: Increasing the dataset to incorporate recent phishing attacks , Enhancing the accuracy of the ML models ,Incorporating the framework with alternative security systems ,Rolling out the framework in actual environments finally ,the paper presents a promising framework for mitigating phishing attacks in intelligent cities. The framework is efficient, expandable, and sturdy, and it has the potentiality to be utilized in actual environments.

References

[1] Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., & Zhao, W. (2017). A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE internet of things journal*, *4*(5), 1125-1142.

[2] Samunnisa, K., Kumar, G. S. V., & Madhavi, K. (2023). Intrusion detection system in distributed cloud computing: Hybrid clustering and classification methods. *Measurement: Sensors*, *25*, 100612.

[3] Virat, M. S., Bindu, S. M., Aishwarya, B., Dhanush, B. N., & Kounte, M. R. (2018, May). Security and privacy challenges in internet of things. In *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 454-460). IEEE.

[4] Lynch, J. (2005). Identity theft in cyberspace: Crime control methods and their effectiveness in combating phishing attacks. *Berkeley Tech. LJ*, *20*, 259.

[5] G, P., Dunna, N. R., & Kaipa , C. S. (2023). Enhancing Cloud-Based IoT Security: Integrating AI and Cyber security Measures. *International Journal of Computer Engineering in Research Trends*, *10*(5), 26–32.

[6] M, P., & K, D. S. D. (2023). ICN Scheme and Proxy re-encryption for Privacy Data Sharing on the Block Chain. *International Journal of Computer Engineering in Research Trends*, *10*(4), 172–176.

[7] Ravikumar, G. ., Begum, Z. ., Kumar, A. S. ., Kiranmai, V., Bhavsingh, M., & Kumar, O. K. . (2022). Cloud Host Selection using Iterative Particle-Swarm Optimization for Dynamic Container Consolidation. *International Journal on Recent and Innovation Trends in Computing and Communication*, *10*(1s), 247–253. https://doi.org/10.17762/ijritcc.v10i1s.5846

[8] Yedukondalu, G., Samunnisa, K., Bhavsingh, M., Raghuram, I. S., & Lavanya, A. (2022). MOCF: A multi-objective clustering framework using an improved particle swarm optimization algorithm. International Journal on Recent and Innovation

Trends in Computing and Communication, 10(10), 143-154. doi:10.17762/ijritcc.v10i10.5743

[9] M, B. M., M, S., & S, H. (2023). Blockchain-Based Crowd funding Platform. *International Journal of Computer Engineering in Research Trends*, *10*(5), 40–47. https://doi.org/10.22362/ijcert.v10i5.46

[10] Butt, U. A., Amin, R., Aldabbas, H., Mohan, S., Alouffi, B., & Ahmadian, A. (2023). Cloud-based email phishing attack using machine and deep learning algorithm. *Complex & Intelligent Systems*, *9*(3), 3043-3070.

[11] Cui, P., & Guin, U. (2019, July). Countering botnet of things using blockchain-based authenticity framework. In *2019 IEEE computer society annual symposium on VLSI (ISVLSI)* (pp. 598-603). IEEE.

[12] Mishra, A., Hsu, C. H., Arya, V., Chaurasia, P., & Li, P. (2021, September). A Hybrid Approach for Protection Against Rumours in a IoT Enabled Smart City Environment. In *International Conference on Cyber Security, Privacy and Networking* (pp. 101-109). Cham: Springer International Publishing.

[13] Sujatha, G., Ayyannan, M., Priya, S. G., Arun, V., Arularasan, A. N., & Kumar, M. J. (2023, February). Hybrid Optimization Algorithm to Mitigate Phishing URL Attacks In Smart Cities. In 2023 3rd International Conference on Innovative Practices in Technology and Management (ICIPTM) (pp. 1-5). IEEE.

[14] Kumar, N., Goel, V., Ranjan, R., Altuwairiqi, M., Alyami, H., & Asakipaam, S. A. (2023). A Blockchain-Oriented Framework for Cloud-Assisted System to Countermeasure Phishing for Establishing Secure Smart City. Security and Communication Networks, 2023.

[15] Pasha, M. J., Pingili, M., Sreenivasulu, K., Bhavsingh, M., Saheb, S. I., & Saleh, A. (2022). Bug2 algorithm-based data fusion using mobile element for IoT-enabled wireless sensor networks. Measurement: Sensors, 24, 100548.

[16] Prakash, P. S., Janardhan, M., Sreenivasulu, K., Saheb, S. I., Neeha, S., & Bhavsingh, M. (2022). Mixed Linear Programming for Charging Vehicle Scheduling in Large-Scale Rechargeable WSNs. *Journal of Sensors*, *2022*.

[17] Assegie, T. A. (2021). An optimized KNN model for signature-based malware detection. *Tsehay Admassu Assegie." An Optimized KNN Model for Signature-Based Malware Detection". International Journal of Computer Engineering In Research Trends (IJCERT), ISSN*, 2349-7084.

[18] Thorat, S. S., Ashwini, M., Kelshikar, A., Londhe, S., & Choudhary, M. (2017). IoT based smart parking system using rfid. *International Journal of Computer Engineering In Research Trends*, *4*(1), 9-12.

[19] Aldakheel, E. A., Zakariah, M., Gashgari, G. A., Almarshad, F. A., & Alzahrani, A. I. (2023). A Deep Learning-Based Innovative Technique for Phishing Detection in Modern Security with Uniform Resource Locators. Sensors, 23(9), 4403.

[20] Kumar, V. K. A., Kumar, M. R., Shribala, N., Singh, N., Gunjan, V. K., Siddiquee, K. N., & Arif, M. (2022). Dynamic Wavelength Scheduling by Multiobjectives in OBS Networks. In N. Jan (Ed.), Journal of Mathematics (Vol. 2022, pp. 1–10). Hindawi Limited. https://doi.org/10.1155/2022/3806018

[21] Revathy, S. ., & Priya, S. S. . (2023). Enhancing the Efficiency of Attack Detection System Using Feature selection and Feature Discretization Methods. International Journal on Recent and Innovation Trends in Computing and Communication, 11(4s), 156–160. https://doi.org/10.17762/ijritcc.v11i4s.6322

[22] Kanna, D. ., & Muda, I. . (2021). Hybrid Stacked LSTM Based Classification in Prediction of Weather Forecasting Using Deep Learning. Research Journal of Computer Systems and Engineering, 2(1), 46:51. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/22

[23] Kumar, S. A. S., Naveen, R., Dhabliya, D., Shankar, B. M., & Rajesh, B. N. (2020). Electronic currency note sterilizer machine. Paper presented at the Materials Today: Proceedings,37(Part 2) 1442-1444. doi:10.1016/j.matpr.2020.07.064 Retrieved from www.scopus.com