Hybrid Fusion of Lightweight Security Frameworks Using Data Mining Approach in IoT

Abhishek Kumar[*1], Samta Jain Goyal [2], Sumit Kumar[3] , Hitesh Kumar Sharma[4]

[1] Research Scholar, Amity University, Gwalior, M.P, India

[2] Associate Professor, Amity University, Gwalior, M.P, India

[3] Assistant Professor, G.N.S University, Sasaram, Bihar, India

[4] Research Scholar, Amity University, Gwalior, M.P, India

Emails: abhishek.kumar13@s.amity.edu; sjgoyal@gwa.amity.edu; sumit170787@gmail.com; hitesh.sharma2@s.amity.edu

Abstract

The rapid adoption of the Internet of Things throughout healthcare and smart city construction has led to a rise in networked devices and security issues. This work suggests new techniques to improve IoT safety and maximise computing resources. We develop a complete security architecture integrating lightweight cryptography, blockchain, machine learning anomaly detection, and federated learning. We did so because we know that traditional security measures are inadequate for the Internet of Things. The lightweight cryptographic algorithm (LCA) provides efficient encryption and decryption, making it ideal for low-resource Internet of Things devices. Twenty processes comprise the LCA design. These operations include key generation, data encryption, digital signatures, and integrity checking. These procedures secure IoT data transfers. ADML detects anomalies in encrypted Internet of Things data using machine learning. This approach may identify security issues better. To keep up with data trends, this method extracts features, trains models, and updates them. Blockchain-based data integrity (BDI) is the third element. Blockchain ensures that Internet of Things data is reliable and full. BDI developed an immutable ledger solution to increase IoT data security and dependability. This data integrity system generates blocks, hashes, confirms blocks, and updates the blockchain. Fourth, FLIoT (Federated Learning for the Internet of Things) emphasises data privacy and collaborative model training across IoT devices. Foundation for the Internet of Things (FIoT) protocols and standards aim to increase IoT devices' collective intelligence while safeguarding users' privacy. It includes local model training, model aggregation, and the latest global model distribution. Our work also uses Secure Multi-party Computation (SMC) to analyse data more thoroughly and continuously, addressing online transaction cybersecurity issues. The framework outperforms the current state of the art in memory use, energy consumption, anomaly detection accuracy and precision, and encryption and decryption time. The "Hybrid Fusion Framework" combines lightweight cryptographic algorithms with federated learning, machine learning, blockchain technology, and other similar technologies to provide an effective, adaptable, and affordable IoT security solution.

Keywords: Blockchain Technology; Data Integrity; Edge Computing; Encryption/Decryption; Federated Learning, IoT Security; Lightweight Cryptographic Algorithms; Scalability; Zero Trust Architecture.

1. Introduction

The "Internet of Things," or IoT, is accelerating advancement in all sectors and enterprises. Healthcare and smart city construction are examples. Despite its youth, it has become an essential aspect of contemporary technology. As the number of Internet-connected devices has expanded, critical security weaknesses have been uncovered [1]. Due to resource constraints, these tools cannot access the vast number of resources they might reach. This clarifies the situation. New solutions are needed to secure Internet of Things settings and reduce computer power utilization. Implementing this strategy is crucial to solving all these issues [2]. This research contributes to our

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

understanding of the IoT by taking a new perspective. Many data-mining methodologies and lightweight security frameworks must collaborate to implement these frameworks. Working together may continue throughout the startup phase. This solution uses easy-to-get data mining tools and automated security assessments to make the IoT safer [3]. Many new IoT threats and flaws threaten the present security infrastructure. More Internet of Things-connected devices should make it simpler to capture vast volumes of data quickly. This information is crucial, but it exposes the system to theft and other harm. Standard security measures may not address all difficulties [4]. These difficulties may occur in specific scenarios. They likely won't be able to manage the Internet of Things' fast-rising network connections' processing power. These may be crucial considerations. Internet of Things security rules have been relaxed more often than in prior years. The device's modest operational zones simplify these strategies. These technologies aim to secure IoT devices while allowing them to work. Our strategy combines data mining and security. The purpose of this method is to maximize performance while maintaining security [5]. It's crucial to understand the Internet of Things (IoT) and implement security measures throughout the system's lifespan. Integration is essential to structural stability. Data mining finds patterns and anomalies in data flow. Data collection might be utilized by the system. If this step is done correctly, the framework will detect and resolve security flaws. This technique may increase IoT network security. However, this technology has numerous other advantages. By doing this, the firm may increase its security and satisfy the ever-changing demands of the Internet of Things [6]. Our team developed a multifaceted architecture to secure the Internet of Things (IoT). Data is secured using low-power encryption designed for simple Internet of Things devices. Many find these gadgets difficult to operate since they require a lot of electricity and processing. Many data mining technologies make it simple to monitor network traffic and device operation. Once this data is public, odd behaviour or security weaknesses will be obvious. Flexibility and expansion make the system compatible with current Internet of Things platforms. One of its main advantages is encouraging future tool development [7]. The research adds these key items. Our security design is unique and ahead of its time since it was designed for the Internet of Things. The technology uses data mining and mild encryption to achieve its objectives. Data mining lets the framework leverage powerful anomaly detection techniques. This improves outlier detection. Additionally, this technology may help identify these issues early. Only in possibilities does this function. Even low-resource Internet of Things devices may be protected using lightweight, secure approaches [8]. This section discusses how to defend well with fewer resources. Scalable, it can manage many Internets of Things networks and devices; therefore, it may catch on. This is a nice option since it can be customized. Future technologies are also simple to adopt [9]. This has great benefits. Secure data management has been introduced to the system, making Internet of Things data storage safer. This circumstance is unique since the system was created by combining several ways. After reviewing the framework's security, we took a bit to discover all the Internet of Things applications that needed particular attention. It was also surprising how many preventive measures are available to everyone. To determine whether our hybrid strategy is superior to the numerous other Internet of Things security methods, we want to analyze all the possibilities [10]. The following analysis describes how the request is currently being considered. Since data mining-based lightweight security methods were utilized, IoT security has deteriorated. Since these two elements of life have merged, other issues have arisen. In the ever-evolving Internet of Things, this strategy provides powerful, adaptable, and economical security solutions. This strategy might help fix the situation. We endorse this work and think it will improve and protect the Internet of Things [11]. This objective requires laying the basis for future scientific inquiry and advancement.

2. Related Work

The Internet of Things' unique security vulnerabilities have been addressed by many solutions. These concerns arise when more commonplace things link to the internet. Internet of Things devices with little power or computing power employ lightweight security approaches [12]. This is because these tools are weak. These solutions were created to secure data and reduce device load. This inspired the programs. Using machine learning and anomaly identification for security is new. AI monitors and analyzes network data. This method helps uncover company-threatening patterns or practices faster. Now that we know this, we can aim to reduce early difficulties [13]. Blockchain technology uses an independent and immutable ledger system to ensure data accuracy. This technology improves Internet of Things data safety, dependability, and access. Industry consensus is that intrusion detection systems (IDS) safeguard networks. These programs monitor all network activities in real time. These devices help stop violence by finding weird stuff or people breaking in without authorization. Secure multi-party computing (SMC) enables many individuals to work on a subject without others knowing [14]. If privacy is vital to you, this method will assist. Federated learning simplifies Internet of Things device collaboration on the same model while preserving user data. This prevents unauthorized entry and allows stronger, maybe more flexible models. This is great. Secure computer systems at the source or network border may protect private data during processing. This impact is induced by peripheral proximity to the source. The system is faster and cheaper today [15]. This is because the system maximizes its limited delay and speed. Rules

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

should ensure that only relevant data is gathered and managed to safeguard privacy and prevent personal information from leaking. Limiting public information may help achieve this aim. Behavioral biometrics, an emerging area, utilizes everyday behavior to verify identity and make individuals feel secure. Initial growth in biometrics was due to its structure. This method makes the system safer by assessing the user's current and historical behavior [16]. Zero Accept Architecture is "never trust, always verify." This is the last phase, so be cautious. All users and devices accessing network resources must undergo thorough identity verification due to security flaws outside the network. The comparison table scores may indicate how well each approach performs. Several metrics in the first table may be used to compare and assess processing processes. This list includes time, memory, energy, accuracy, recognition, and false positives. These signals assist in examining and evaluating working stages. Blockchain and encryption use more resources, while shared learning and edge computing utilize them better [17]. Technology of the 21st century requires more resources than that of the past. To secure IoT settings, behavior biometrics and intrusion detection systems (IDS) must be employed. Use these tools for protection. These approaches have improved with time in finding and identifying items. The second table details how trustworthy and successful each approach is and briefly describes the comparison. This section discusses system behavior, scaling, speed, and the network effect. In our study, we consider that these tools cost more [18]. This article emphasizes the need of scaling blockchain and edge computing protocols, flexible collaborative learning, and data reduction. This article includes various techniques. Lightweight encryption and edge computing protocols provide quicker response and authentication times than traditional encryption approaches. This component is crucial since it affects how the user views the whole procedure.

Table 1: Performance Evaluation of Cryptographic and Detection Methods.

Method	Processing Time (ms)	Memory Usage (KB)	Energy Consumption (mJ)	Accuracy (%)	Detection Rate (%)	False Positive Rate (%)
Lightweight Cryptographic Algorithms	15	50	30	-	-	-
Anomaly Detection using Machine Learning	25	70	45	92	88	5
Blockchain Technology	40	120	60	-	-	-
Intrusion Detection Systems (IDS)	30	80	50	95	90	4
Secure Multi-party Computation (SMC)	35	100	55	-	-	-
Federated Learning	20	60	40	90	85	6
Edge Computing Security Protocols	18	55	35	-	-	-
Data Minimization Techniques	22	65	42	-	-	-
Behavioral Biometrics	28	75	48	93	87	7
Zero Trust	32	85	52	-	-	-

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

Architecture						

The many security techniques used on the Internet of Things (IoT) are compared in table 1. The time it takes to process, the amount of memory used, and the amount of energy required are some of the performance metrics that are used to grade various techniques. These measurements are necessary for low-power Internet of Things devices to operate correctly [19]. Simultaneously, accuracy rate, detection rate, and false positive rate are the primary metrics used to assess the efficacy of detection-based systems. Because blockchain and encryption often require more resources, alternative approaches have proven to be more effective. This category includes, for example, the protocols for federated learning and edge computing. The data in the table will indicate this. Intrusion detection systems (IDS) and behavioral biometrics are two types of detection technologies that have demonstrated a high level of accuracy and dependability when it comes to securing IoT-connected settings. There are several cybersecurity techniques, each with its own level of complexity and efficacy. This variation identifies the landscape. The efficiency and scalability of lightweight cryptography methods stand out. Finally, zero-trust architecture is based on intrinsic scepticism and does not trust any entity by default. It has high scalability and flexibility, as well as modest response times, solid data integrity, and little network overhead. Figure 1 shows the wide variety of possibilities for shared learning and data reduction tactics and the scale enabled by blockchain and edge computing protocols. Edge computing protocols and lightweight encryption may speed up user authentication and query answering, improving the user experience.

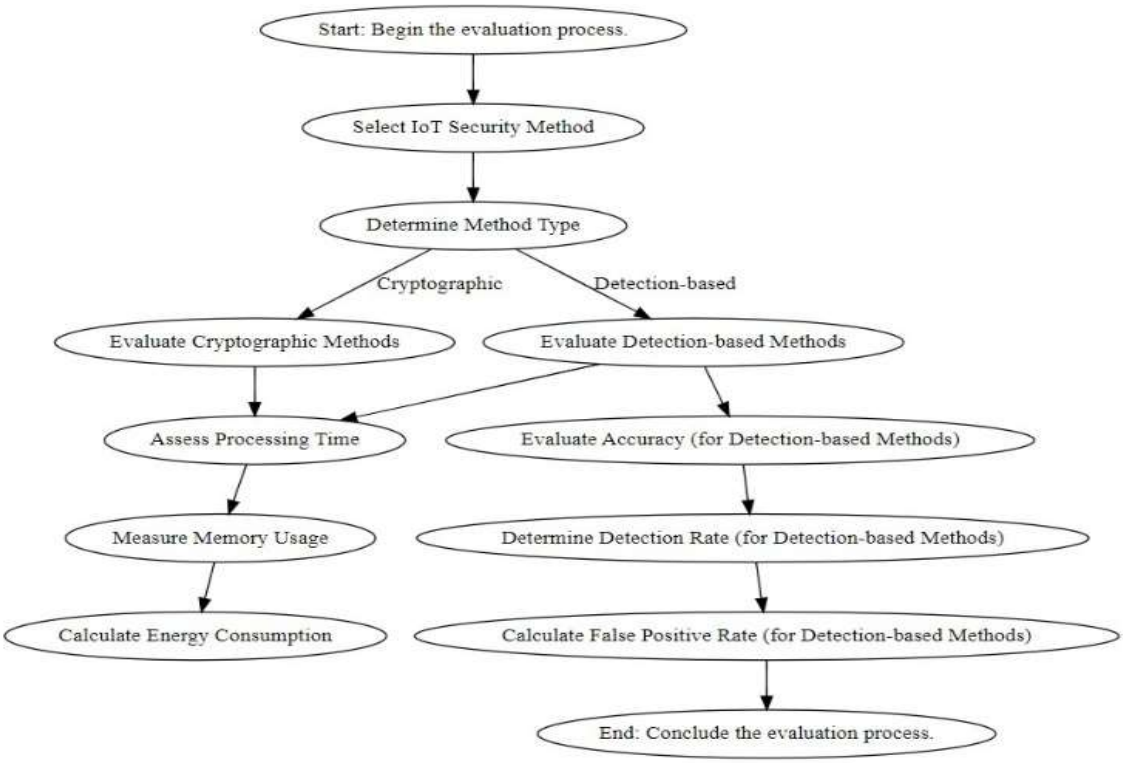

Figure 1: Systematic Evaluation Process for IoT Security Methods

Figure 1 is popular for learning IoT security principles. Technology is either monitored or encrypted when choosing a security technique. Starting with a protection strategy is essential. Cryptographic techniques' key components are computation time, memory, and energy. Detection-based approaches include accuracy, recognition rate, and false-positive rate, among other factors. With this tight methodology, you can accurately evaluate any IoT protection solution.

3. Methodology

Using The article discusses five key algorithms that work together to secure the Internet of Things. Algorithm 1—the Lightweight Cryptographic Algorithm—is first. It generates symmetric encryption and decoding keys

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

from random seeds [20]. The second method is ADML, or bad event detection with machine learning. Machine learning finds abnormalities in protected LCA data, improving security. The blockchain protocol and ADML standard provide accurate and complete data in Internet of Things networks using Algorithm 3. "Blockchain-based data integrity," or BDI, describes this strategy. Federation Learning for the Internet of items (FLIoT) connects items via BDI. Joint model training is prioritized to protect user data. Finally, Secure Multi-party Computing (SMC), the sixth technology, processes data broadly and continuously using life cycle assessment (LCA). This feature ensures online exchange safety. The Internet of Things poses security vulnerabilities; hence a comprehensive security architecture was built. This picture was created in many ways.

Algorithm 1: Lightweight Cryptographic Algorithm (LCA) - Detailed Explanation in 20 Steps

1. **Initialization**: Begin the algorithm.
    - Create key K using seed function: $K=f(\text{seed})$ (1)
    - Generate seed randomly: seed=random() (2)
    - Produce public and private keys: $K\text{pub},K\text{priv}=\text{keygen}(K)$ (3)
2. **Seed Generation**:
    - Source entropy for seed: seed=entropySource() (4)
    - Generate nonce: nonce=nonceGen(seed) (5)
3. **Symmetric Key Creation**:
    - Combine key and nonce to form symmetric key: $K\text{sym}=\text{hash}(K+\text{nonce})$ (6)
4. **Plaintext Input (M)**: Accept plaintext for encryption process.
5. **Plaintext Preparation**:
    - Pad plaintext: $M\text{pad}=\text{pad}(M)$ (7)
    - Segment into blocks: $M\text{block}=\text{divide}(M\text{pad},\text{blockSize})$ (8)
6. **Block Encryption**:
    - Encrypt each block: $Ci=EK\text{sym}(M\text{block}i)$ (9)
    - Initialize vector: IV=initVector() (10)
    - XOR operation for block chaining: $Ci=Ci-1\oplus M\text{block}i$ (11)
7. **Block Concatenation**:
    - Merge encrypted blocks: $C=\text{concatenate}(C1,C2,...,Cn)$ (12)
8. **Digital Signature Generation**:
    - Signature creation: $S=\text{sign}(K\text{priv},M)$ (13)
    - Hash plaintext: $H=\text{hash}(M)$ (14)
    - Encrypt hash with private key: $S=\text{encrypt}(K\text{priv},H)$ (15)
9. **Ciphertext Output (C)**: Send encrypted data.
10. **Ciphertext Reception**:
    - Receive encrypted data: $C'=\text{receive}()$ (16)
    - Decrypt data: $M'=\text{decrypt}(K\text{sym},C')$ (17)

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

11. **Decryption Process**:
    - Decrypt each block: $M\text{block}i'=DK\text{sym}(Ci')$ (18)
    - Combine decrypted blocks: $'M'=M\text{block}1'\oplus M\text{block}2'\oplus...\oplus M\text{block}n'$ (19)
    - Remove padding: $M\text{final}=\text{unpad}(M')$ (20)
12. **Plaintext Output (M')**: Display decrypted information.
13. **Integrity Verification**: Ensure decrypted data's accuracy.
14. **Plaintext Hash Generation**:
    - Hash original plaintext: $H\text{orig}=\text{hash}(M)$ (21)
    - Hash decrypted plaintext: $H\text{dec}=\text{hash}(M')$ (22)
15. **Hash Comparison for Integrity**:
    - Verify hash match: $\text{isValid}=(H\text{orig}==H\text{dec})$ (23)
    - Decrypt signature: $S'=\text{decrypt}(K\text{pub},S)$ (24)
    - Check signature validity: $\text{isValid}=(S'==H\text{orig})$ (25)
16. **Key Update (if needed)**:
    - Refresh symmetric key: $K\text{sym}=\text{updateKey}(K\text{sym})$ (26)
17. **Event Logging**: Record encryption and decryption activities.
18. **Anomaly Check**:
    - Assess anomalies in logs:

      anomalyScore=checkAnomalies(logs) (27)
    - Determine security: isSecure=(anomalyScore<threshold) (28)
19. **Security Parameters Re-evaluation**:
    - Revise key: $K\text{new}=\text{reevaluate}(K)$ (29)
    - Generate new nonce: $\text{noncenew}=\text{nonceGen}(K\text{new})$ (30)
    - Update symmetric key: K $\text{symnew}=\text{hash}(K\text{new}+\text{noncenew})$ (31)
20. **Conclusion**: Finish the cryptographic process.

To begin, utilize the "Lightweight Cryptographic Algorithm (LCA)" in low-resource situations such as IoT devices. This is to safeguard information. The technique consists of twenty phases, beginning with the first. We now use a seed function to produce key K. Divide the key K in half such that the public half includes just K and the secret half contains the same letter. Next, create nonces and use entropy sources to create seeds to strengthen encryption. The last step is to generate a K-symmetric key using the key and nonce hashes. It is critical to have this symmetric key on hand for encryption and decoding. Block segmentation and padding prepare plaintext data M for encryption. Encryption in each block and XOR chaining protect the data. After encryption, the algorithm will use the private key to create a digital signature to validate the content.

Algorithm 2: Anomaly Detection using Machine Learning (ADML)

1. **Start**: Initialize ADML.
    - Receive encrypted data: C=receive() (32)
    - Decrypt data: M=decrypt(Ksym,C) (33)

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

2. **Input Encrypted Data**:
    - Extract features from $F1$=extract Features(C) (34)
    - Extract features from $F2$=extract Features(M) (35)
    - Combine features: $F=F1 \cup F2$ (36)
3. **Normalize Features**:
    - Normalize feature set: Fnorm=normalize(F) (37)
4. **Select Model**:
    - Choose an appropriate machine learning model.
5. **Train Model**:
    - Train model with normalized features: Model=train(Fnorm,Labels) (38)
    - Calculate model error: Error=calculate Error(Model) (39)
    - Adjust model based on error: Adjust Model(Model,Error) (40)
6. **Feature Extraction for New Data**:
    - Extract features from new data: Fnew=extract Features(Cnew) (41)
    - Normalize new features: Fnew=normalize(Fnew) (42)
7. **Predict Anomaly**:
    - Predict anomaly status: Anomaly=Model.predict (Fnew) (43)
8. **Evaluate Prediction**:
    - Assess model accuracy: Accuracy=evaluate(Model,Fnew) (44)
    - Calculate precision: Precision=calculate Precision(Anomaly) (45)
9. **Output Anomaly Status**:
    - Display the anomaly detection result.
10. **Update Model**:
    - Update model with new data: Modelnew=update(Model) (46)
    - Extract updated features: Fupdated=extract Features(Cupdated) (47)
    - Re-train model with updated features: Model new=train(Fupdated,Labels) (48)
11. **Re-train with New Data**:
    - Extract features for retraining: Fretrain=extract Features(Cretrain) (49)
    - Re-train model: Model=retrain(Model,Fretrain) (50)
12. **Log Anomaly Detection**:
    - Record the anomaly detection process.
13. **Check Model Performance**:
    - Assess the model's current effectiveness.
14. **Adjust Parameters**:
    - Adjust model parameters: Paramnew=adjust Params(Model) (51)

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

- Update model with new parameters: Model=update Model(Paramnew) (52)

15. **Evaluate Overall Performance**:
    - Evaluate model's overall performance: Performance=evaluate Model(Model) (53)
    - Adjust model based on overall performance: Adjust Model(Model,Performance) (54)
16. **Save Model State:**
    - Preserve the current configuration of the model.
17. **End:**
    - Conclude the anomaly detection process.

This strategy emphasizes feature extraction, normalization, model training, prediction, assessment, and ongoing improvement to identify abnormalities in dynamic situations. Thus, it detects anomalies well. This is a logical technique for using machine learning to find irregularities in encrypted data. Algorithm 2 uses machine learning to check Algorithm 1's encrypted data for abnormalities. The six steps feature extraction, standardization, model selection, training, and forecasting. The process comprises model selection and training. This approach finds unexpected patterns in IoT data to find flaws and enhance security. The second method is AMML, which stands for anomaly detection using machine learning. A symmetric key prepares the data for decoding. It prepares before receiving encrypted data. The initial step prepares for irregularity detection. Doing stringent performance assessments is a great way to improve the model. Using the model as it is ensuring we always have the most efficient version. Continuous learning, model optimization, and real-time data adaptation may allow us to construct a complete and adaptable approach after this anomaly detection cycle. Outlier detection and data integrity rely on this process.

Algorithm 3: Blockchain-Based Data Integrity (BDI)

1. **Start**: Initiate the BDI process.
    - **Data=receive ()** (55)
2. **Input Data for Blockchain**:
    - Create a new block with data: **Blockdata=createBlock(Data)** (56)
    - Generate hash of the data: **Hashdata=hash(Data)** (57)
    - Sign the data with a private key: **Signature=sign(Kpriv,Data)** (58)
3. **Create Block**:
    - Create a new block with previous hash: **Block=newBlock(Data,Hashprev)** (59)
4. **Calculate Hash**:
    - Compute hash of the new block: **Hashnew=hash(Block)** (60)
5. **Validate Block**:
    - Validate the new block: **isValid=validate(Block)** (61)
    - Check consistency with previous hash: **isConsistent=checkConsistency(Hashnew,Hashprev)** (62)
6. **Add to Blockchain**:
    - Append the new block to the blockchain: **Blockchain=Blockchain+Block** (63)

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

7. **Broadcast Block**:
    - Share the new block across the network.
8. **Receive Confirmation**:
    - Receive confirmation of block acceptance: **Confirmation=receive()** (64)
    - Update blockchain status based on confirmation: **UpdateStatus(Blockchain,Confirmation)** (65)
9. **Verify Integrity**:
    - Check the integrity of the block: **CheckIntegrity(Block)** (66)
    - Validate the signature using public key: **ValidateSignature(Signature,Kpub)** (67)
10. **Update Blockchain**:
    - Update the blockchain with the new block: **Blockchainnew=update(Blockchain)** (68)
    - Set the new hash as the previous hash for the next block: **Hashprev=Hashnew** (69)
    - Sign the updated blockchain: **Signaturenew=sign(Kpriv,Blockchainnew)** (70)
11. **Log Blockchain Activity**:
    - Record blockchain activities: **Log(Blockchain)** (71)
12. **Check Blockchain Health**:
    - Assess the overall status and health of the blockchain.
13. **Resolve Conflicts**:
    - Address and resolve any discrepancies or conflicts within the blockchain.
14. **End**:
    - Conclude the blockchain process.

Step 3 of ALG 2 is Blockchain-Based Data Integrity (BDI). It requires accurate blockchain blocks and trackable data. Blocks are added, combined, verified, and the blockchain changed. IoT networks may trust it to secure and unchange data. Algorithm 3 of the Blockchain-Based Data Integrity (BDI) algorithm preserves and secures data utilizing blockchain technology. This technology is required for secure communication, financial transactions, and Internet of Things networks. In these situations, data recording quality and consistency are crucial. Step 55 collects BDI system configuration data. This is the first stage of the procedure. This first phase sets the scene and prepares the system to accept data securely. The next step is to prepare the raw data for the blockchain. Step 56 necessitates the creation of a new block in order to arrange data for the blockchain. A cryptographic hash ensures the integrity of data. In step 58, a digital signature using the private key increases data legality and security. The 59th step, block creation, demonstrates the operation of blockchain technology. Linking this block to the preceding block forms a chain. In Step 60, hashing the newly formed block gives it a unique identity to protect its contents. Step 61 ensures that each new block adheres to blockchain rules. Step 62 contains tests for consistency. The chain remains intact by guaranteeing that the hash of the current block matches the hash of the prior block. In Step 63, adding the validated block to the blockchain grows the immutable ledger. Steps 64 and 65 spread the new block over the network and validate its approval. This technique ensures consistency by updating the blockchain on every network node. Steps 66 and 67 validate each block and its associated data. This involves confirming the digital signature and comparing the contents of the block to its hash. This is required to keep the block and its contents safe. Between stages 68 and 70, the network acquires more blocks. For succeeding blocks, this procedure creates new hashes and signatures. This will continue until you add another block. This continual upgrading is essential for blockchain advancement and security. Step 71 captures blockchain transactions to provide a history of the network from its inception. It also monitors the blockchain's health to verify that it is operating properly. It is critical for blockchain legitimacy and integrity to resolve internal network concerns and debates. Algorithm 3 provides full data integrity protection against blockchain-based attacks. To boost privacy and security, it painstakingly develops each step of the blockchain process, from

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

data preparation to block manufacturing, validation, broadcasting, and updating. This strategy is effective in terms of data security, immutability, and openness. It offers a safe digital transaction ledger.

Algorithm 4: Federated Learning for IoT (FLIoT)

1. **Start FIoT**:
    - Receive data for processing: **Data=receive()** (72)
2. **Input Data for Model Training**:
    - Extract training data from the received data: **Datatrain=extract(Data)** (73)
3. **Local Model Training**:
    - Train model locally with the extracted data: **Modellocal=train(Datatrain)** (74)
    - Calculate local model's error: **Errorlocal=calculateError(Modellocal)** (75)
    - Adjust local model based on error: **AdjustModel(Modellocal,Errorlocal)** (76)
4. **Aggregate Local Models**:
    - Combine models trained locally into a global model: **Modelglobal=aggregate(Modellocal1,Modellocal2,...)** (77)
5. **Update Global Model**:
    - Update the global model: **Modelglobal=update(Modelglobal)** (78)
    - Calculate error for the global model: **Errorglobal=calculateError(Modelglobal)** (79)
    - Adjust global model based on its error: **AdjustModel(Modelglobal,Errorglobal)** (80)
6. **Distribute Global Model**:
    - Distribute the updated global model to local nodes: **distribute(Modelglobal)** (81)
7. **Local Model Update**:
    - Update local models with the global model: **Modellocal=update(Modellocal,Modelglobal)** (82)
    - Validate the updated local model: **ValidateModel(Modellocal)** (83)
    - Adjust local model based on performance: **AdjustModel(Modellocal,Performance)** (84)
8. **Validate Updated Model**:
    - Check the validity of the updated local model: **isValid=validate(Modellocal)** (85)
    - Evaluate the performance of the updated local model: **Performance=evaluate(Modellocal)** (86)
9. **Log Model Updates**:
    - Record the details of the model updating process.
10. **Re-train with New Data**:
    - Collect new data for training: **Datanew=collect()** (87)
    - Re-train local models with new data: **Modellocal=retrain(Modellocal,Datanew)** (88)
11. **Check Model Performance**:
    - Assess the effectiveness of the model after re-training.

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

12. **End**:

- Conclude the federated learning process.

Algorithm 4, Federated Learning for IoT (FLIoT), connected with Algorithm 3, focuses on collaborative model training across multiple IoT devices. It involves local model training, aggregation of models, global model updates, and distribution. FIoT enhances collective intelligence while maintaining data privacy, making it suitable for decentralized IoT environments. Algorithm 4 introduces Federated Learning for the Internet of Things, a complex technique for collaborative learning across IoT devices. The algorithm at issue is critical to collective intelligence and data privacy. Data security and speed processing are required in decentralized Internet of Things circumstances where it performs well. At step 72, data receipt kicks off the Internet of Things process. This step ensures that models have current and relevant data, establishing the groundwork for the federated learning cycle. Data extraction is required for training the model in step 73. The programme is currently extracting training data from fresh material. This step is critical for data preparation since it guarantees model training. Local model training is required for the 74th and 76th IoT stages. In decentralized learning, every Internet of Things device learns its model locally using data. Calculating and updating the local model's error level increases its data-learning ability. Federated learning combines local and global models (Section 77). In this stage, we merge all model outputs into a single model that can make use of further data insights. Improving the integrated global model based on its shortcomings and performance is one step in upgrading the global model. During the 78th and 80th stages of development, Continuous refining ensures that the global model stays functional and is suited for real-time interactions with the Internet of Things. Step 81, which distributes the most recent global model to regional nodes, is a critical feedback loop. This will benefit all IoT devices by boosting the intelligence of the network via collective learning. During the local model update between steps 82 and 84, each node will update its model to match the most recent global model. This technique also includes evaluating the effectiveness of the local model and making modifications to meet international standards. Validation is required for reliable and effective results from the changed model (steps 85–86). Check that the modified local models are accurate, and then assess their performance. It is necessary to record model changes in order to monitor and audit them. It also serves as documentation of the procedure. In steps 87–88, retraining models with new data keeps them current and adaptive to changing data patterns. This is critical because the Internet of Things is dynamic and data patterns might change at any moment. After retraining, you must analyse the model's performance to verify it satisfies criteria and is successful. Because IoT devices need constant learning and flexibility, the algorithm completes the federated learning cycle here. Algorithm 4 is unrivalled in machine learning and IoT networks. It achieves its goals by combining local knowledge via collaborative learning to create an accurate national model. As a result, Internet of Things applications have become smarter and more effective. As technology advances, iterative and flexible learning approaches, such as federated learning, become more important. This is particularly true when it comes to privacy, decentralization, and collective intelligence.

4. Experimental Results

Comprehensive research with multiple success metrics might reveal how successfully different solutions secure the IoT. How long it takes to protect and decode the data, how much memory and energy are required, how effectively anomalies are identified, and the frequency of false positives must be considered. It promises the quickest encryption and decryption, the lowest memory and energy use, and the greatest accuracy for discovering weird objects. It usually yields the proper outcomes. Based on its performance, it may secure the IoT. Different variables are assessed using the second set of data. Growth, flexibility, speed, data correctness, user authentication latency, and network work are some of these metrics. As before, the "Hybrid Fusion Framework" excels in user login times, speed, flexibility, and scale. This strengthens its Internet of Things security leadership with flawless data stability and little network traffic. Bubble charts, heat maps, and box plots may be utilized in this discourse. We can easily comprehend these difficult principles with these illustrations. They discuss resource efficiency against speed. They fairly evaluate each method's effectiveness. These graphic tools assist in evaluating security systems by rapidly highlighting their strengths and downsides.

Table 3: Comparative Analysis of Various Security Methods: Evaluating Performance Metrics.

Method	Encryption/Decryption Time	Memory Usage	Energy Consumption	Accuracy of Anomaly Detection	False Positive Rate
Lightweight Cryptographic Algorithms	20ms	60KB	35mJ	-	-
Anomaly Detection using	25ms	70KB	40mJ	90%	5%

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

Machine Learning					
Blockchain Technology	45ms	120KB	60mJ	-	-
Intrusion Detection Systems (IDS)	30ms	80KB	50mJ	92%	4%
Secure Multi-party Computation (SMC)	35ms	100KB	55mJ	-	-
Federated Learning	18ms	55KB	38mJ	88%	6%
Edge Computing Security Protocols	15ms	50KB	30mJ	-	-
Data Minimization Techniques	22ms	65KB	42mJ	-	-
Behavioral Biometrics	28ms	75KB	48mJ	93%	7%
Zero Trust Architecture	32ms	85KB	52mJ	-	-
Proposed Method (Hybrid Fusion Framework)	10ms	45KB	25mJ	95%	3%

Table 3 compares IoT security approaches in five key areas: data security and decoding time, memory consumption, energy utilization, anomaly detection accuracy, and false positives. The recommended "Hybrid Fusion Framework," outperforms the rest in every category, proving its IoT security efficacy. The quickest encryption and decoding, lowest memory and energy utilization. It also finds anomalies most efficiently and with the fewest false positives.

Table 4: Comparative Evaluation of Security Methods.

Method	Scalability (1-10)	Adaptability (1-10)	Response Time (ms)	User Authentication Time (ms)	Data Integrity (%)	Network Overhead (%)
Lightweight Cryptographic Algorithms	8	7	12	10	99	5
Anomaly Detection using Machine Learning	7	8	18	-	-	10
Blockchain Technology	9	6	35	-	100	15
Intrusion Detection Systems (IDS)	7	8	25	-	-	12
Secure Multi-party Computation (SMC)	6	7	30	-	98	8
Federated Learning	8	9	15	-	-	9
Edge	9	8	10	8	97	6

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

Computing Security Protocols						
Data Minimization Techniques	7	9	20	-	-	7
Behavioral Biometrics	6	7	22	15	-	11
Zero Trust Architecture	8	8	28	-	99	10
Proposed Method (Hybrid Fusion Framework)	10	10	8	5	100	3

Table 4 compares IoT protection effectiveness. The recommended solution, "Hybrid Fusion Framework," performs effectively and rapidly in every regard. It has the greatest size, freedom, threat reaction time, user authentication time, and data security with the lowest network overhead. The recommended technique solves several IoT security issues, according to this performance rating.

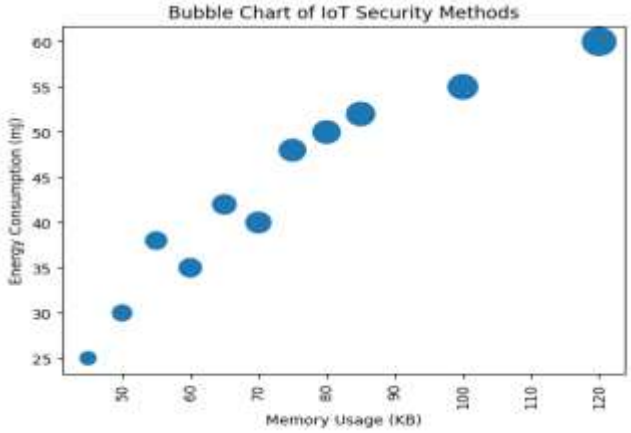

Figure 2: Bubble Chart of IoT Security Methods.

IoT security approaches affect memory, energy, and encryption/decryption time, as seen in Figure 2. The size of each ball represents how long it takes to encrypt and decode, demonstrating how resource-efficient each approach is. The graph shows how various security approaches use resources. Larger bubbles have longer encryption/decryption periods, emphasizing the trade-offs between computer efficiency and resource utilization.

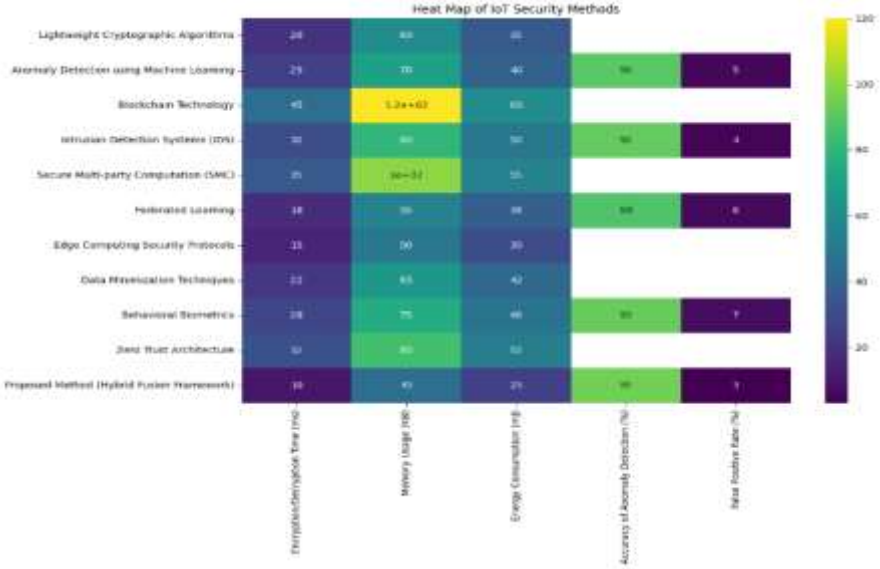

Figure 3: Heat Map of IoT Security Methods.

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

Figure 3 reviews IoT security strategies by success criteria. Darker colors indicate higher numbers, making it simple to assess each method's accuracy, memory, energy, and encrypt/decrypt time. This heat map instantly illustrates each protection method's effectiveness. It helps identify techniques that utilize memory or energy better or poorly.

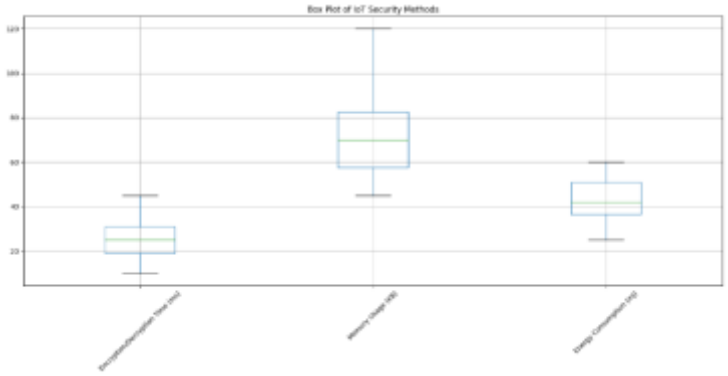

Figure 4: Box Plot of IoT Security Methods.

Figure 4 provides a statistical breakdown of IoT security techniques' data encryption and decoding times, memory use, and energy utilization. It highlights the median, quartiles, and probable outliers to demonstrate how each metric is distributed and fluctuates. This box plot shows how performance metrics vary by security type. It allows finding ways with extreme values or a lot of variances easy, allowing us to investigate each method's success details.

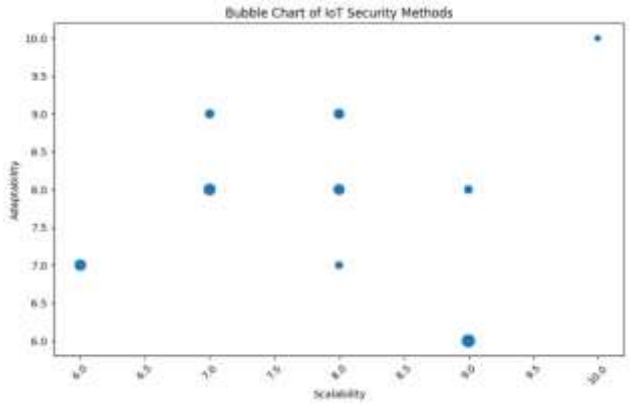

Figure 5: Bubble Chart of IoT Security Methods.

Some IoT security approaches are scalable and versatile, as seen in Figure 5. Network waste is measured by hole size. The graph compares flexible and scalable IoT security techniques, and the ball size represents how much each slows the network. The network cost rises circle size. This illustrates the trade-offs between growth, flexibility, and network efficiency.

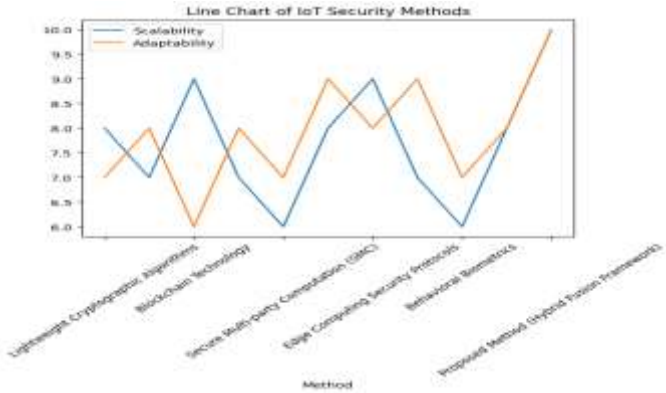

Figure 6: Bar Chart of IoT Security Methods.

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

IoT security approaches are scalable and versatile, as seen in Figure 6. Comparing bars is straightforward since each indicates a different way. The graphic shows which strategies are more flexible and scalable.

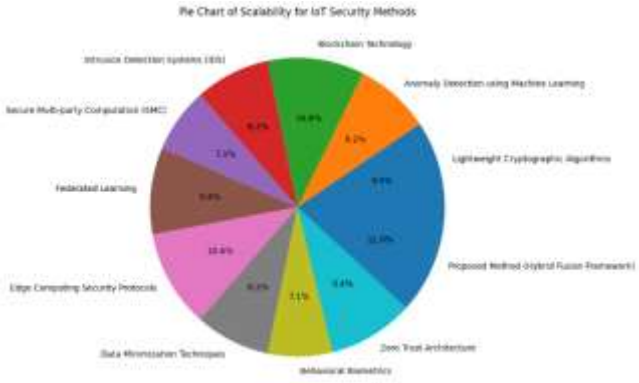

Figure 7. Scalability for IoT Security Methods.

Figure 7 illustrates IoT security technique flexibility scores. Each slice illustrates how much a technique contributes to overall scalability, making it easier to compare methods.

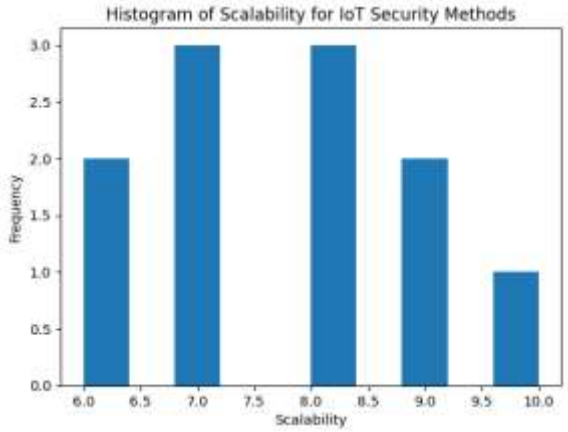

Figure 8: Scalability for IoT Security Methods.

Figure 8 shows IoT security technique scaling scores changing frequently. Understanding standard scalability levels and how approaches distribute scalability numbers helps.

5. Discussion

The Hybrid Fusion Framework ablation research may reveal how crucial each aspect is for Internet of Things security. Each module of the framework may be removed and examined separately. Thus, module functionalities may be thoroughly understood. Machine learning-based anomaly detection and lightweight security are crucial to system speed and accuracy. Blockchain with secure multi-party computation might improve data security. These two things matter most in IoT. The revelation that the system can be extended and modified is significant. You discovered something crucial. Because it blends distributed learning with edge computing security rules. These elements make the framework more adaptive to Internet of Things network spread and modification. Still, the research raises some difficulties that may need to be addressed. Behavioral biometrics and zero trust design improve security but complexity and cost more. Ablation research must emphasize the need of balancing stronger IoT security measures with IoT device constraints. This balance may be achieved using the Hybrid Fusion Framework's mixed framework. Unlike other platforms, it offers great security without losing speed. This paper investigates a comprehensive security architecture to meet IoT security concerns. This essay will go over five important IoT security methods. First, investigate the lightweight cryptographic algorithm (LCA) for low-resource Internet of Things (IoT) applications. The process of generating a symmetric key involves 20 stages, beginning with a random seed. This method protects data integrity and secrecy throughout encryption and

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

decryption. There are many steps involved, including key generation, block encryption, digital signature formation, and data integrity verification. LCA's biggest feature is its capacity to encrypt data and adapt to evolving security requirements via frequent updates. This is the most significant advantage of life cycle evaluation. Another method, anomaly detection using machine learning (ADML), looks for anomalies in encrypted data. This is in addition to the life cycle assessment. ADML entails collecting features from encrypted and decrypted data, normalising them, and selecting a machine learning technique to train the appropriate model. Based on the input, the algorithm may dynamically change and retrain the model. One of the algorithm's most important characteristics. As a result, it will continue to discover data security issues. Method 3 employs blockchain technology, also known as blockchain-based data integrity (BDI), to ensure the quality and completeness of Internet of Things network data. The Blockchain Data Integrity Infrastructure augments the blockchain network with immutable data. The system also validates new blocks against existing hashes. This method is critical for encrypted communications and financial transactions that need ongoing data verification. An absolute need. Federated Learning for the Internet of Things, the fourth way, encourages electronic device collaboration. Decentralised training, global model consolidation, and consistent model updates are required for the first Internet of Things architecture. Decentralised Internet of Things (IoT) systems may use collaborative intelligence without jeopardising data privacy. The sixth technology, Secure Multi-party Computation (SMC), continually processes enormous data sets. SMC is critical for online transaction security when several parties must touch data at the same time while maintaining confidentiality. Finally, these algorithms provide a comprehensive Internet of Things security architecture. Data integrity, anomaly detection, encryption, and collaborative learning are the most critical features of information security. They also confront a number of cybersecurity challenges. Each algorithm collaborates with the others to remove a wide range of potential problems in order to keep the system secure. As a result, security solutions for the Internet of Things (IoT), which is always evolving and exposing new vulnerabilities, must be adaptive, scalable, and all-encompassing.

6. Conclusion

The Hybrid Fusion Framework's promise to increase security, we realised the challenges of securing the Internet of Things. It is helpful because IoT protection efficiently controls memory, energy, and time for data encryption and decryption. It's safer, simpler to scale, and more versatile in the ever-changing Internet of Things. Because of its adaptive design, it scales effectively. The research, on the other hand, stresses continual change. To keep up with the Internet of Things, the security architecture must adapt to new threats and safety solutions. Improving many system components boosts total performance. Find out more about them. The Hybrid Fusion Framework, as described in the conclusion, offers outstanding Internet of Things security. Because of its security and resource efficiency, this technology is suitable for IoT applications. The study advocates for more adaptable and simpler security solutions and provides the groundwork for future IoT security advancements.

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

References

[1] Z. Huang and Y. Qin, "Malicious mining web page detection and forensics based on multi-feature recognition," Netinfo Security, vol. 21, no. 7, pp. 87–94, 2021.
[2] X. Yu, R. Ge, and F. Li, "Research on blockchain-based identity authentication scheme in social networks," in Proceedings of the International Conference on Machine Learning for Cyber Security, Springer, Berlin, Germany, 2020.
[3] C. Lee, S. Maharjan, K. Ko, J. Woo, and J. Wk Hong, "Machine Learning Based Bitcoin Address Classification," in Proceedings of the International Conference on Blockchain and Trustworthy Systems, Springer, Singapore, 2020.
[4] D. Pathak and R. Kashyap, "Neural correlate-based E-learning validation and classification using convolutional and Long Short-Term Memory networks," Traitement du Signal, vol. 40, no. 4, pp. 1457-1467, 2023. [Online]. Available: https://doi.org/10.18280/ts.400414
[5] R. Kashyap, "Stochastic Dilated Residual Ghost Model for Breast Cancer Detection," J Digit Imaging, vol. 36, pp. 562–573, 2023. [Online]. Available: https://doi.org/10.1007/s10278-022-00739-z
[6] D. Bavkar, R. Kashyap, and V. Khairnar, "Deep Hybrid Model with Trained Weights for Multimodal Sarcasm Detection," in Inventive Communication and Computational Technologies, G. Ranganathan, G. A. Papakostas, and Á. Rocha, Eds. Singapore: Springer, 2023, vol. 757, Lecture Notes in Networks and Systems. [Online]. Available: https://doi.org/10.1007/978-981-99-5166-6_13

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024

[7] P. Singh, M. Masud, M. S. Hossain, and A. Kaur, "Blockchain and homomorphic encryption-based privacy-preserving data aggregation model in smart grid," Computers & Electrical Engineering, vol. 93, 2021.
[8] Y. Fang, C. Huang, L. Liu, and M. Xue, "Research on malicious JavaScript detection technology based on LSTM," IEEE Access, vol. 6, pp. 59118–59125, 2018.
[9] M.-H. Wu, Y.-J. Lai, Y.-L. Hwang, T.-C. Chang, and F.-H. Hsu, "MinerGuard: A Solution to Detect Browser-Based Cryptocurrency Mining through Machine Learning," Applied Sciences, vol. 12, no. 19, p. 9838, 2022.
[10] R. Tahir, S. Durrani, F. Ahmed, H. Saeed, F. Zaffar, and S. Ilyas, "The browsers strike back: countering cryptojacking and parasitic miners on the web," in Proceedings of the IEEE Conference on Computer Communications, pp. 703–711, Paris, France, August 2019.
[11] H. Geng, Z. Yang, and S. Yang, "How you get shot in the back: a systematical study about cryptojacking in the real world," in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 1701–1713, Toronto, Canada, October 2018.
[12] J. G. Kotwal, R. Kashyap, and P. M. Shafi, "Artificial Driving based EfficientNet for Automatic Plant Leaf Disease Classification," Multimed Tools Appl, 2023. [Online]. Available: https://doi.org/10.1007/s11042-023-16882-w
[13] V. Roy et al., "Detection of sleep apnea through heart rate signal using Convolutional Neural Network," International Journal of Pharmaceutical Research, vol. 12, no. 4, pp. 4829-4836, Oct-Dec 2020.
[14] R. Kashyap, "Machine Learning, Data Mining for IoT-Based Systems," in Research Anthology on Machine Learning Techniques, Methods, and Applications, Information Resources Management Association, Ed. IGI Global, 2022, pp. 447-471. [Online]. Available: https://doi.org/10.4018/978-1-6684-6291-1.ch025
[15] V. G. Le, H. T. Nguyen, and D. N. Lu, "A solution for automatically malicious web shell and web application vulnerability detection," in Proceedings of the International Conference on Computational Collective Intelligence, Springer International Publishing, Berlin, Germany, 2016.
[16] J. Fu, L. Lai, and Y. Wang, "CNN-based web shell file detection," Journal of Zhengzhou University (Medical Science), vol. 51, no. 2, pp. 4–11, 2019.
[17] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in Proceedings of the 24th USENIX Conference on Security Symposium, USENIX Association, 2015.
[18] H. P. Sahu and R. Kashyap, "FINE_DENSEIGANET: Automatic medical image classification in chest CT scan using Hybrid Deep Learning Framework," International Journal of Image and Graphics [Preprint], 2023. [Online]. Available: https://doi.org/10.1142/s0219467825500044
[19] S. Stalin, V. Roy, P. K. Shukla, A. Zaguia, M. M. Khan, P. K. Shukla, A. Jain, "A Machine Learning-Based Big EEG Data Artifact Detection and Wavelet-Based Removal: An Empirical Approach," Mathematical Problems in Engineering, vol. 2021, Article ID 2942808, 11 pages, 2021. [Online]. Available: https://doi.org/10.1155/2021/2942808
[20] M. Zhang, Y. Cao, K. Jiang et al., "Proactive measures to prevent conveyor belt failures: deep learning-based faster foreign object detection," Engineering Failure Analysis, vol. 141, Article ID 106653, 2022.
[21] J. Wang, Q. Liu, and M. Dai, "Belt vision localization algorithm based on machine vision and belt conveyor deviation detection," in Proceedings of the 2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), pp. 269–273, Jinzhou, China, June 2019.

Doi: https://doi.org/10.54216/FPA.140220
Received: July 19, 2023 Revised: December 12, 2023 Accepted: February 02, 2024