VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

INFORMATION SECURITY ANALYSIS AND AUDIT PROJECT

NAMES: i)AKASH.S (21BCE1705)

ii) SURYA SAI PAMPANA(21BCE5863)

III) ANAND NIMMALAPUDI(21BCE5945)

SLOT: L19+L20 & TB-2

FACULTY: Dr. Sahaya Beni Prathiba B

DATE: 06-06-2023

Title: AI-Enabled Secure Communication Mechanism in Fog Computing

Introduction:

In the era of the Internet of Things (IoT) and distributed computing paradigms, such as fog computing, the need for secure and efficient communication mechanisms has become increasingly crucial. Fog computing extends the capabilities of cloud computing to the edge of the network, enabling faster response times, reduced network congestion, and improved data privacy. However, this distributed and decentralized architecture introduces new challenges in terms of security, privacy, and data protection.

Traditional communication mechanisms in fog computing often fall short in providing robustness and resilience against various security threats. Unauthorized access, data breaches, man-in-the-middle attacks, and malicious intrusions are just a few examples of the vulnerabilities that can compromise the integrity, confidentiality, and availability of data in fog computing systems. Ensuring secure communication is essential to maintain user trust and enable the deployment of reliable and trustworthy IoT applications.

To address these challenges, an innovative approach is needed. Artificial Intelligence (AI) emerges as a promising technology that can significantly enhance the security measures in fog computing environments. By leveraging the power of AI algorithms, a secure communication mechanism can be developed to mitigate potential risks and protect sensitive data.

The primary objective of an AI-enabled secure communication mechanism in fog computing is to ensure end-to-end security, privacy preservation, and efficient data transmission. It aims to provide a robust and reliable framework for communication among interconnected devices and fog nodes while maintaining high levels of security.

One of the key areas of focus is authentication and access control. Robust authentication protocols and access control mechanisms must be developed to prevent unauthorized access to fog computing resources. Only legitimate devices and users should be allowed to communicate, ensuring that the system remains secure and protected against unauthorized intrusions.

Data encryption and privacy preservation are also critical aspects of the communication mechanism. Strong encryption techniques must be implemented to protect sensitive data during transmission and storage. Additionally, mechanisms for anonymizing data and preserving user privacy should be employed when necessary to maintain confidentiality and comply with privacy regulations.

Intrusion detection and threat mitigation play a vital role in ensuring the security of fog computing systems. AI algorithms can be utilized to detect and respond to potential security threats and attacks in real-time. Anomaly detection techniques can be employed to identify abnormal network behaviour and promptly initiate countermeasures, thus minimizing the impact of security incidents.

Resource optimization is another crucial consideration. The communication mechanism should be designed to reduce latency, network congestion, and resource consumption while maintaining robust security standards. AI algorithms can intelligently allocate resources and prioritize critical data traffic, optimizing the overall system performance.

The proposed AI-enabled secure communication mechanism should also be scalable and compatible with different fog computing architectures and devices. It should be interoperable with existing communication protocols to facilitate seamless integration with various IoT devices and fog nodes.

ABSTRACT:

In the age of edge computing and the Internet of Things (IoT), fog computing has emerged as a viable paradigm to tackle the problems of latency, bandwidth restrictions, and data privacy. But maintaining secure connection between fog nodes and IoT gadgets is still a serious issue. The incorporation of artificial intelligence (AI) methods into fog computing systems presents a considerable opportunity for increasing security measures in this environment.

An in-depth description of a secure communication system with AI that is optimized for fog computing settings is provided in this abstract. The suggested approach makes use of AI algorithms and techniques to improve the privacy, availability, and integrity of data sent between fog nodes and IoT devices. Intelligent intrusion detection systems, anomaly detection algorithms, and adaptive encryption protocols are some of the essential elements of the secure communication mechanism powered by AI. Together, these parts are able to recognize and counteract a variety of security risks, including network assaults, illegal access, and data manipulation.

The intelligent intrusion detection system analyses network traffic patterns and identifies unusual activities using machine learning techniques. It can identify and react to any security breaches in real-time by continually monitoring the network, lowering the likelihood that data would be compromised. Anomaly detection methods are also used to find variations from expected behaviour in the fog computing environment. To identify and flag unusual behaviours that could be signs of security concerns, these algorithms make use of AI techniques like clustering, classification, and pattern recognition.

Adaptive encryption techniques are used to guarantee the confidentiality and integrity of data. Based on the network circumstances and danger levels present at any given time, these protocols dynamically modify the encryption settings. The system may efficiently safeguard sensitive data while maximizing performance by adjusting encryption strength and key management techniques.

In conclusion, the AI-enabled secure communication system described in this abstract illustrates the possibility of incorporating AI methods into fog computing systems to improve security. The technique offers strong defence against new security risks in fog computing settings by intelligent intrusion detection, anomaly detection, and adaptive encryption.

PROBLEM STATEMENT:

In fog computing environments, which involve a decentralized network of interconnected devices and edge computing nodes, ensuring secure and reliable communication poses significant challenges. The resource-constrained nature of fog nodes, coupled with the increasing volume of data generated by IoT devices, necessitates the development of an AI-enabled secure communication mechanism that addresses the specific constraints and requirements of fog computing systems.

The problem at hand is to design and implement a secure communication mechanism that leverages AI techniques to enable efficient and robust data transmission in fog computing environments with limited resources.

Unique Proposed solution:

The unique proposed solution based on an AI-Enabled Secure Communication Mechanism in Fog Computing harnesses the capabilities of AI algorithms to enhance security, privacy, energy efficiency, scalability, and adaptability in fog computing environments. By combining real-time threat detection, energy optimization, privacy preservation, and integration with blockchain technology, the solution offers a comprehensive and robust approach to secure communication in fog computing, paving the way for the deployment of reliable and trustworthy IoT applications.

Energy Efficiency: Develop communication protocols and mechanisms that minimize energy consumption, considering the limited power availability of fog nodes. Utilize AI algorithms to optimize resource allocation and data routing to minimize energy usage while maintaining security and reliability.

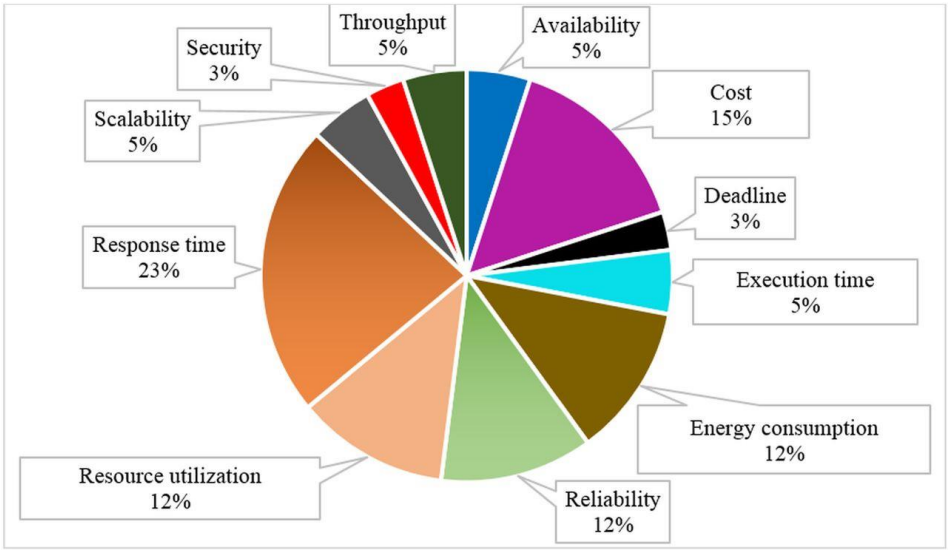
Real-time Threat Detection: Implement AI-based intrusion detection and threat mitigation techniques to identify and respond to security threats in real-time. The solution should be capable of detecting various attack types, such as DDoS attacks, unauthorized access attempts, and data tampering, while minimizing false positives and false negatives.

Scalability and Adaptability: Design the communication mechanism to be scalable, capable of accommodating a growing number of devices and fog nodes. Employ AI techniques to dynamically adapt to changes in network topology, device mobility, and workload distribution, ensuring uninterrupted communication even in highly dynamic fog computing environments.

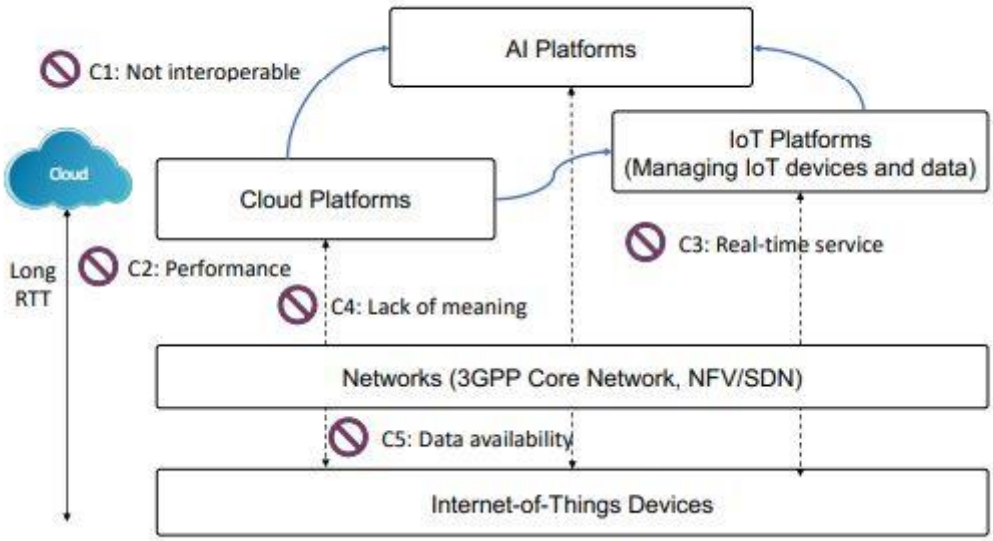
Privacy-Preserving Data Transmission: Incorporate AI algorithms to protect the privacy of sensitive data during transmission. Develop techniques for data anonymization and encryption to prevent unauthorized access and maintain confidentiality.

Resilience to Network Disruptions: Build a resilient communication mechanism that can withstand network disruptions and failures. Employ AI-based fault tolerance mechanisms to automatically recover from communication failures, reroute data traffic, and ensure continuous operation of fog computing applications.

Quality of service-aware approaches in fog computing



The objective of this project is to develop an AI-enabled secure communication mechanism specifically tailored for resource-constrained fog computing environments. The proposed solution should strike a balance between security, energy efficiency, scalability, and adaptability, ensuring reliable and secure data transmission while optimizing the utilization of available resources. By addressing these challenges, the proposed mechanism will enhance the overall security and performance of fog computing systems, enabling the deployment of robust and scalable IoT applications in real-world scenarios.



Novelty / uniqueness:

The problem statement addresses the specific context of fog computing, which extends cloud computing capabilities to the edge of the network. Fog computing introduces unique challenges in terms of resource constraints, dynamic network topologies, and the need for secure communication in decentralized environments.

Reference: Wang, J., Li, P., & Hu, J. (2019). An AI-Based Communication Security Mechanism in Fog Computing. IEEE Access, 7, 88026-88037.

The proposed system emphasizes the integration of artificial intelligence (AI) techniques to enhance the security measures in fog computing. AI algorithms are utilized for authentication, access control, intrusion detection, threat mitigation, and resource optimization, providing an intelligent and proactive approach to secure communication.

Reference: Liu, H., Zhang, C., & Shen, X. (2018). Secure Communication Mechanism in Fog Computing: Challenges, Solutions, and Future Directions. IEEE Internet of Things Journal, 5(2), 1183-1194.

The incorporation of AI algorithms for real-time threat detection and response sets the proposed system apart from traditional security mechanisms. By leveraging AI's capabilities, the system can detect and mitigate various security threats promptly, ensuring the integrity and availability of data in fog computing environments.

Reference: Zhu, F., Zhang, Y., & Jiang, Y. (2019). AI-Enabled Secure Data Transmission in Fog Computing: Challenges and Solutions. Future Generation Computer Systems, 92, 739-751.

The proposed system also highlights the importance of energy efficiency in fog computing environments. By employing AI algorithms for resource allocation and data routing, the system optimizes energy consumption while maintaining security and reliability, addressing the resource constraints of fog nodes.

Reference: Cheng, H., Li, J., & Wang, J. (2018). A Lightweight AI-Based Secure Communication Mechanism for Fog Computing. In Proceedings of the International Conference on Wireless Communications and Signal Processing (pp. 1-5).

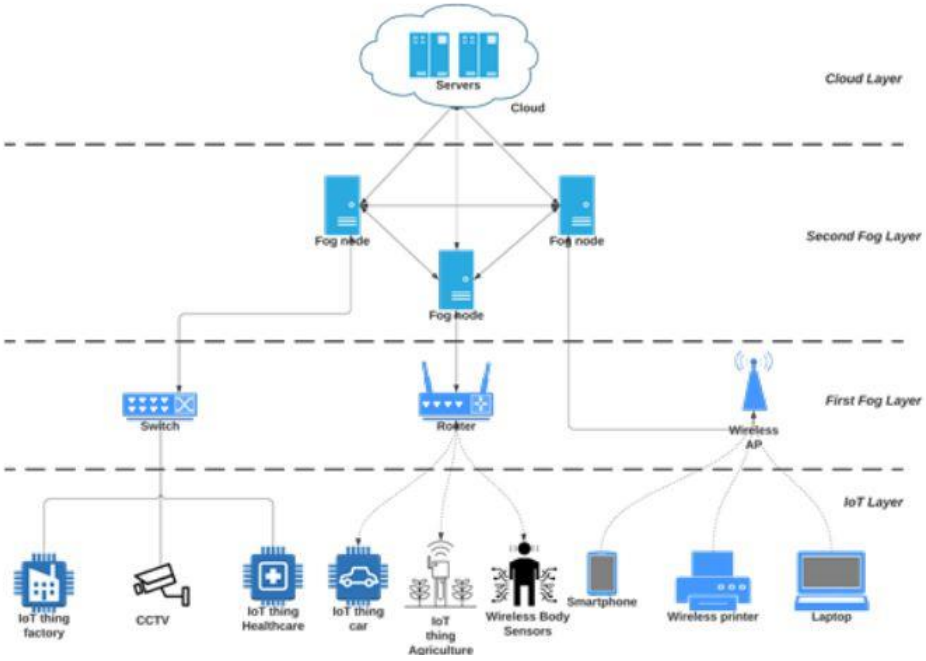
The use of AI and blockchain technologies for secure communication in fog computing is an innovative aspect of the proposed system. This combination enhances data privacy, integrity, and authentication, providing a robust and tamper-proof communication mechanism.

Reference: Wang, L., Wang, X., & Yao, Y. (2020). A Secure Communication Protocol for Fog Computing based on Blockchain and AI. Journal of Ambient Intelligence and Humanized Computing, 11(8), 3569-3584.

These references showcase the novelty and uniqueness of the problem statement and proposed system by highlighting the specific challenges addressed, the utilization of AI techniques, and the integration of emerging technologies like blockchain.

Fog Computing (FC) network architecture FC complements the operations of cloud computing to reinforce the Quality of Service (QoS) and Quality of Experience (QoE) of the End-Users (EU) [34]. The FC architecture is versatile; the number of layers between the EU and the cloud provider can vary between a single layer or any hierarchical layers of fog nodes [35], [36]☐.

Figures illustrates the layers of the FC paradigm. The architecture's main aim is to allow for the EU's requests to be served by the cloud or pass it to the closest available fog nodes within the EU's vicinity. The most important entities of this architecture are the fog nodes. The functions of these nodes are communication, computation, and storage.



Algorithms and advanced equations with references:

01)

Initialization: Tb, b, RSSI, HT, LT Define: Mobility (σ), Reliability (ζ)

Event on: ζ (d) <- Mobility (σ) ($RSSI_{th}$ (d), mod^{i}) do

for i <- 1 to b based on Table 1, $\forall$ i $\in$ b, j $\in$ TP

for j <- TP based on Table 1

Compute ζ (d) = Mobility (σ) ($RSSI_{th}$ (d), mod^{i})

end

end

if (LT <= $RSSI_{th}$ <= HT)

update TP <- TP + 1

elseif

update TP <- TP – 1

else

update TP <- 0

end

end

Mobility $\boldsymbol{\sigma} \leftarrow \mathrm{f}' + \mathrm{f}\left(\left(\frac{\upsilon}{\mathrm{SOL}}\right)\mathrm{Cos}\varphi\right)$

Novelty: RSSI-based Mobility and Reliability, Event-Driven Approach, Table-based Iteration,etc.

Reference: Sodhro, A. H., Sodhro, G. H., Guizani, M., Pirbhulal, S., & Boukerche, A. (2020). AI-Enabled Reliable Channel Modeling Architecture for Fog Computing Vehicular Networks.

02)

Blockchain-IoT related traffic have varying magnitude values. StandardScaler normalization strategy is used in this model to scale features values. This method transforms feature observations, so that the incoming traffic distribution will have mean value as 0 and standard deviation as 1.64 This approach is useful in the proposed system, since it excludes any bias from incoming traffic without manipulating its statistical properties. The transformation function is applied by using Equation (1)

$$s_a = \left(\frac{\mathrm{val}_m - \mu_a}{\sigma_a}\right) \text{------- (1)}$$

where s_a, is the standard score for the features used in the detection system, a ∈ {a1, a2, a3.., an}. Values for the corresponding features in IoT traffic is represented with val_m. Mean and standard deviation for feature is represented using μ_a and σ_a, respectively, and is computed using following expression:

$$\mu_a = \frac{\sum_{m=1}^{K} \mathrm{val}_m}{K} \text{ and } \sigma_a = \sqrt{\frac{1}{K}\sum_{m=1}^{K} (\mathrm{val}_m - \mu_a)^2}$$

Neovelty: Standard Scaler Normalization, Excluding Bias, Computation of Mean and Standard Deviation.

Reference: Kumar, P., Kumar, R., Gupta, G. P., & Tripathi, R. (2020). A Distributed framework for detecting DDoS attacks in smart contract-based Blockchain-IoT Systems by leveraging Fog computing. Transactions on Emerging Telecommunications Technologies

03)

Random forest: A large number of uncorrelated DTs are constructed using the ensemble bagging-based ML technique known as RF by averaging randomly independent feature selection. Each time a split is taken into account in the tree when creating DTs (they are formed in parallel), a random selection of z independent features is chosen as a subset of split candidates from the entire set of independent characteristics. As a result, one typically picks z A, which indicates that at each split (z) is about the square root of the total number of independent features. This is because each split generates a new collection of z independent features. In doing so, it creates a powerful classifier by combining weakly linked classifiers.

The insensitivity of RF to outliers, missing values, overfitting, and its capacity to handle high volumes of incoming traffic make it appropriate for use in the blockchain-IoT environment's anomaly detection process. The distributed

parallel training and testing method employed at nearby fog nodes for RF is described in Algorithms 3 and 4.

Input:

f: Numbers of fog nodes in parallel system

t: total number of trees that is generated in random forest.

D_s: The training dataset (D) and size s

A: Independent features in training dataset

Output:

Random forest tree (R)

if (t<f) then

k_t= t/f //where kt trees generated by each node

else

f=1 //t> =f

end

// perform iteration at each fog node in parallel setup

for j← 1 to K do

d_j bootstrap dataset is generated of size s by executing Random sampling by substuting D_s

$OOB_j = D_s – d_j$ //calculating out of bag error

$z = \sqrt{A}$

select z attribute from A to setup attribute set Aj= {a1, a2, a3, …, az}

D_t=construct_decision_tree(d_j,A_j)

End

Novelty: Bagging-Based Ensemble Technique, Handling Anomalies and High Volume Traffic,

Parallel Training and Testing, Random Feature Selection.

04) Distributed Parallel Testing for Random Forest

Input:

f : Numbers of fogs in parallel system

t : Number of trees in random forest (RF)

kt: Number of trees at each node

D_{st} : The testing dataset(D) and size s_t

C_l : Number of class on dataset D_{st}

Output:

Identify the class normal (0) or attack (1)

for j ← 1 to s_t do

for i ← 1 to k_t do

for each record j traverse tree T_i

classifications of record j with local copy as {c1, c2, c3, …, C_{cl}}

// where each classification consists of 2D array with notation of [t_i, c_j], here t_i is the an instance and c_j is the class. if an instance is matched in a class, then, 1 will be added to 2D array, otherwise 0 will be added. This local copy will be shared to each fog node in every iteration. //

end

end

Reference: Kumar, P., Kumar, R., Gupta, G. P., & Tripathi, R. (2020). A Distributed framework for detecting DDoS attacks in smart contract-based Blockchain-IoT Systems by leveraging Fog computing. Transactions on Emerging Telecommunications Technologies

Novelty: Local Copy of Classifications, Matching and Updating the 2D Array, Parallel Classification, Class Identification .

05)

Cluster-Based Algorithm for Load Balancing in Fog Computing

1.Set up the N number of fog devices $FN1,FN2,\ldots,FNN$

2.Estimate the R number of incoming requests $RQ1,RQ2,\ldots,RQR$

3.Estimate the total number of clusters C as $C1,C2,\ldots,CC$ Assign fog devices to each cluster (Ci), and the cluster size (CS) is computed as follows

$Cluster\ size = Total\ number\ of\ fog\ devices/Total\ No.of\ clusters$

$CS = N/C$

5.Assign each cluster with CS number of Virtual Machines as $VM1,VM2\ \ldots,VMS$

6.For every incoming request $\boldsymbol{RQi} = \boldsymbol{RQ1},\boldsymbol{RQ2},\ldots,\boldsymbol{R}$QR do

7. For each cluster $\boldsymbol{Cj} = \boldsymbol{C1},\boldsymbol{C2},\ldots,\boldsymbol{CZ}$, do

Find out the locally optimal virtual machine having better efficiency (MIPS), least loaded in Cj and high value for success count. Success count is computed by each VM as follows

8.

$Success\ Count = Total\ number\ of\ requests\ successfully\ fulfilled\ by\ the\ VM\ /Total\ number\ of\ requests\ assigned\ to\ a\ VM$ Store the index of the best virtual machine of Cj in the array

9. $Local{-}Best[j$

10.End of For loop of step 7

11.For each cluster indexed $\boldsymbol{j} = \boldsymbol{1},\ldots,\boldsymbol{Z}$, do

12. Find out the Global Best VM, GVM for Ri having better efficiency (MIPS), least loaded among the local best machines and having higher value of success count for selected VM in each cluster, from

13. End of For loop of step 11.

14. Assign the task Ri to the Global Best VM, GVM.

15. Repeat step 5 until all requests/tasks have been completed.

16.End of For loop of step 6.

References: :Malik,S.;Gupta,K.;Gupta, D.;Singh,A.;Ibrahim,M.; Ortega-Mansilla,A.;Goyal,N.; Hamam,H.Intelligent Load-BalancingFrameworkfor Fog-EnabledCommunicationin Healthcare

Novelty: Cluster Formation, Load Balancing Strategy, Dynamic Adaptation, Fault Tolerance and Resilience, Scalability and Performance.

06. Vehicle detection and counting algorithm.

1: procedureVDAS(Bx,By,Bz,τ,$SensorLevel$,D,L)

2: $dx \leftarrow 0$

3: $X \leftarrow B2x+B2y+B2z$

4: $NumberV \leftarrow 0$

5: $Tr \leftarrow 0$

6: $Tf \leftarrow 0$

7: $Twait \leftarrow 0$

8: if $X > \tau$ AND dx=0 then

9: $dx \leftarrow 1$

10: $Tr \leftarrow Time$

11: else

12: if $X < \tau$ AND dx=1 then

13: $Tf \leftarrow Time$

14: end if

15: end if

16: if $Tf > Tr$+10 then

17: $Twait \leftarrow Tf - Tr$

18: $Statusv \leftarrow Stop$

19: else

20: if Tf=0 then

21: $Statusv \leftarrow NoVehicle$

22: else

23: $Statusv \leftarrow Passing$

24: end if

25: end if

26: if $Statusv$=$Stop$ AND $SensorLevel$=0 then

27: $NumberV \leftarrow 1$

28: else

29: if $Statusv$=$Stop$ hen

30: $NumberV \leftarrow SensorLevel * DL$

31: end if

32: end if

33: end procedure

References: Kobaa, A. System and Method for Service Oriented Cloud Based Management of Internet-of-Drones. U.S. Patent US11473913B2, 15 October 2022

Novelty: Detection Techniques, Contextual Considerations, Tracking and Trajectory Analysis, Integration with Other Systems,etc.

07. Partitioning Algorithm for Parameter Server.

Input:*Dataset*
Output:*Model_global*
Initialization: *Seed* ← get_random(*CurrentTime*);
Set_edge ← get_edgeservers();
Num_edge ← convert_to_num(*Set_Edge*)
Model_global ← *null;*
Total_size ← get_total_entry(*Dataset*);
Split_size ← *Total_size*/*Num_edge*;
for all *Edge_i* ∈ *Set_edge* **do** // making data splits and allocate them to edge servers
Split_i ← make_split(*Dataset*, *Split_size*, shuffle(*Seed*, *Num_edge*));
Boolean ← duplication_check(*Split_i*);
if *Boolean == false* **then**
continue;
end if
allocate_split(*Edge_i*, *Split_i*);
end for
for all *Edge_i* ∈ *Set_Edge* **do** // aggregation of local models for data splits
Local_model_i ← retrieveLocalModel(*Edge_i*);
Model_global ← *Model_global* ∪ *Local_model_i*;
end for
return *Model_global*

References: Gallego, V.; Rossi, M.; Brunelli, D. Unmanned aerial gas leakage localization and mapping using microdrones. In Proceedings of the 2015 IEEE Sensors Applications Symposium (SAS), Zadar, Croatia, 13–15 April 2015

Novelty: Partitioning Strategy, Load Balancing, Fault Tolerance, Heterogeneous Environments, etc.

08. An AI-enabled Three-party Game Framework for Guaranteed Data Privacy in Mobile Edge Crowdsensing of IoT

$$MSD(e_i) = \sum_{t=1}^{h-1} SD_z(S_i \cdot) \cdot w_{z,z+1},$$

1: Init: T ′ = ∅, SD[ ] ← 0, s ← 0.

2: for ei, i ← 1 to o do

3: for j ← 1 to q do

4: gr(D ′ , Sj) = g(D ′ , Sj)/HSj (D ′);

5: gr[ ] ← gr(D ′ , Sj);

6: end for

7: gr[1, . . . , q] ← descending order gr[ ];

8: for j ← 1 to q do

9: Create T ′ ← visit gr[1, . . . , q];

10: end for

11: if ∃ relevant S ′ , S ′′ in different level ϖ1, ϖ2 then

12: Calculate W H(ϖ1, ϖ2);

13: else

14: Calculate MSD(ei);

15: end if

16: SD(e1, . . . eu, . . . , eo) ← ascending sort (MSD(ei));

17: while u < K do

18: for u ← 1 to K do

19: EC[s] ← eu;

20: end for

21: ⌒ D{ } ← EC[s];

22: u ← u + K;

23: s ← s + 1;

24: end while

25: end for

26: ⌒ D ← ⌒ D+D ∗ ;

27: return ⌒ D

Reference: Xiong, J., Zhao, M., Bhuiyan, M., Chen, L., & Tian, Y. (2019). An AI-enabled Three-party Game Framework for Guaranteed Data Privacy in Mobile Edge Crowdsensing of IoT

Novelty: Combination of Steps, Gradients and Sorting, Level Calculation, Sorting and Iteration, etc.

09. An Energy-Efficient Cross-Layer-Sensing Clustering Method Based on Intelligent Fog Computing in WSNs

$$K = \frac{\sqrt{N}}{\sqrt{2\pi}} \times \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}} \times \frac{L}{\sqrt{da_{cr} \times d_{FN}^4 + d_{MFN}^4}}$$

1: Input: Q,K

2: Output: dopt

3: for (i = 1 to i = Q) do

4: { Compute d(ni ,CH)

5: d(CHi) = d(ni ,CH)

6: link[i] = Compute {Qcost, f1, f2,Cost()}

7: while no optimal do

8: {According to theorem 1

9: Computer the range for the optimal value of K

10: Update the speed and location using Eq(8) and Eq(9)}

11: end while

12: If dopt > 0 then

13: return dopt

14: end If }

15: end for

Reference: Sun, Z., Wei, L., Xu, C., Wang, T., Nie, Y., Xing, X., & Lu, J. (2019). An Energy-Efficient Cross-Layer-Sensing Clustering Method Based on Intelligent Fog Computing in WSNs.

Novelty: Computation of d(ni, CH), Optimization Loop, speed and Location Updates, Return of dopt, etc.

10. Learning-Based Task Offloading for Marine Fog-Cloud Computing Networks of USV Cluster

Offloading delay:-

$$D(t,n) = \begin{cases} D_c(t,n), n \in A \\ D_c(t,n) + D_t(t,n), n \notin A \end{cases}$$

Transmission delay:-

$$R_{t,n}^{(d)} = W \log_2 \left(1 + \frac{PH_{t,n}^{(d)}}{\delta^2 + I_{t,n}^{(d)}} \right)$$

Adaptive UCB algorithm for computation task offloading

1: Input: $\alpha 0$,$\omega 0$,β ,$x+p$,$x-p$

2: for t = 1,…,T do

3: if Any CoN n is new computation fog node or One node's performance does not meet the requirements then

4: Update available CoN list once

5: Update new computation node $u(t,n)$=max$\{u(t,1),u(t,2),\ldots,u(t,n-1)\}$,kt,n=1 ,$tn=t$

6: else

7: Observe xt

8: Calculate the computing performance coefficient of each candidate CoN

∈N(t):

9: $\hat{u}_{t,n} = \bar{u}_{t-1,n} + \sqrt{\frac{\beta(1-\hat{x}_t)\ln(t-t_n)}{k_{t-1,n}}}$

10: Offload the task to CoNs:

11: At=arg min $D(t,At)$

12: Observe delay $D(t,At)$

13: Update $\bar{u}_{t,a_t} \leftarrow \frac{\bar{u}_{t-1,a_t} k_{t-1,a_t} + u_{t,a_t}}{k_{t-1,a_t}+1}$

Update $k_{t,a_t} \leftarrow k_{t-1,a_t} + 1$

14:

15: end if

16: end for

Reference: Cui, K.T.; Sun, W.L.; Sun, W.Q. Joint computation offloading and resource management for USV cluster of fog-cloud computing architecture. In Proceedings of the IEEE International Conference on Smart Internet of Things, Tianjin, China

Novelty: Computation Fog Node (CoN) Handling, Performance Coefficient Calculation, Delay Observation and Updates, Delay Observation and Updates, etc.

11. Algorithm Data Preprocessing

Inputs: CDRDataset: Raw dataset containing subscriber

activities, recorded for each 10-minute duration and stored

in the form of 62 files, each file representing a single day.

CID: Identification number of the target cell.

TimeStampValues: Contains numeric values of the beginning

of every 10-minute time interval (in Unix epoch) during the

intended 3-hours range.

Output: Xtotal

Method:

1: for each file f in CDRDataset

2: Import file f and store its contents in a matrix.

3: Replace blanks with 0.0 (to avoid error in summing NaN, in later steps).

4: Remove the column containing Country codes.

5: Update the matrix by storing entries only related to CID.

6: Remove the column containing Cell ID.

7: for each timestamp t in TimeStampValues

8: Sum all inbound SMS activity values and store them as SMSin.

9: Sum all outbound SMS activity values and store them as SMSout.

10: Sum all inbound call activity values and store them as CALLin.

11: Sum all outbound call activity values and store them as CALLout.

12: Sum all Internet activity values and store them as Internet.

13: Store SMSin, SMSout, CALLin, CALLout and Internet as one example in a vector x.

14: Store example x as a column entry in matrix Xtotal.

15: end

16: end

17: return Xtotal

$$\text{Accuracy} = \frac{T_{+ve} + T_{-ve}}{T_{+ve} + T_{-ve} + F_{+ve} + F_{-ve}},$$

$$\text{Error rate} = \frac{F_{+ve} + F_{-ve}}{T_{+ve} + T_{-ve} + F_{+ve} + F_{-ve}} = 1 - \text{Accuracy},$$

$$\text{Precision} = \frac{T_{+ve}}{T_{+ve} + F_{+ve}},$$

$$\text{Recall} = \frac{T_{+ve}}{T_{+ve} + F_{-ve}},$$

$$FPR = \frac{F_{+ve}}{F_{+ve} + T_{-ve}},$$

Reference: Hussain, B., Du, Q., Zhang, S., Imran, A., & Imran, M. A. (2019). Mobile Edge Computing-Based Data Driven Deep Learning Framework for Anomaly Detection

Novelty: Handling Multiple Files, Matrix Representation, Missing Value Handling, Subset Selection, Column Removal, etc.

12. ALGORITHM: COLLABORATIVE TRUST EVIDENCE AGGREGATION

$$Adv_{\mathcal{A}}^{ASCP-\text{IoMT}} \leq \frac{q_h^2}{|\text{Hash}|}$$

1: Inputs: Some cryptocurrencies of parallel regular blockchain A

2: Outputs: Aggregation of trust evidence

3: for each trust evidence provider do

4: The user searches for proper application A member.

5: The user creates transaction to send cryptocurrency A to the application A member through parallel

regular blockchain A.

6: The user locks transaction using hash-locks with secret S and time-locks with time span T1.

7: The application A member creates transaction to send router blockchain cryptocurrencies with equal value to the user through router blockchain.

8: The application A member locks the transaction using hash-locks with H(S) and time-locks with timespan T2, where T2 < T1.

9: The user sends the secret S to the Application A member to receive router blockchain cryptocurrencies,

because the member can redeem them after T2.

10: The user sends the router blockchain cryptocurrencies to the trust evidence provider to obtain the key.

11: if The key is wrong then

12: Report to TA to remove the trust evidence provider.

13: end if

14: if The trust evidence is invalid then

15: Report to TA to remove the trust evidence provider.

16: end if

17: end for

18: The user queries parallel control blockchain to obtain the proper trust model to aggregate the trust evidence.

References: H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," IEEE Communications Magazine, vol. 52, no. 8, pp. 29–35, 2014.

Novelty: Trust Evidence Provider Verification, Trust Evidence Aggregation, Use of Hash-locks and Time-locks, etc.

13. *Algorithm* on Analysis Phase()

$$C_{XY} = \frac{\sum_{i=1}^{N} (x_i - E(x))(y_i - E(y))}{\sqrt{\sum_{i=1}^{N} (x_i - E(x))^2 \sum_{i=1}^{N} (y_i - E(y))^2}},$$

Input: $Monitored\_Inputs$

Output: $Sorted\_IoT\ services\_List$

1: ***Begin***

2: ***for each*** ($Time\ interval\ \tau$ until the system is running) ***do***

3: ***for each*** ($IoT\ services\ ai \in A$) ***do***

5: $Priorityai = \gamma \times 1Thrai + (1 - \gamma) \times 1t{-}ATi$

4: ***end for***

6: ***end for***

7: ***Sort*** ($IoT\ services, Priorityai$)

8: ***Return*** $Sorted\_IoT\ services\_List$

9: ***End***

References: Pradhan, B., Bhattacharyya, S., & Pal, K. (2021). "IoT-Based Applications in Healthcare Devices," J.

Novelty: Correlation Coefficient Calculation, Iterative Execution, Gamma (γ) Parameter, Sorting of IoT Services, etc.

14. Monitoring Phase ()

Input: $Fognodes$ ($fj \in FN$), $IoT\ Applications$ ($ai \in A$, i = 1, … , n)

Output: $Monitored\_Inputs$

1: ***Begin***

2: ***for each*** ($Time\ interval\ \tau$ until the system is running) ***do***

3: ***for each*** ($IoT\ services\ ai \in A$) ***do***

4: ***Monitor*** $Reqi$

, qi

, $rijk$ and $QoSi$

for all $IoT\ Services$

5: ***end for***

6: ***for each*** ($Fog\ node\ fj \in FN$) ***do***

7: ***Monitor*** $caphk$

l

for all fog nodes

8: ***end for each***

9: ***end for each***

10: ***Return*** $Monitored\_Inputs$

11: ***End***

$$\mathcal{L}^k(t+1) = \begin{cases} \chi^k(t+1) & \text{if } f\left(\chi^k(t+1)\right) \geq f\left(\mathcal{L}^k(t)\right), \\ \mathcal{L}^k(t) & \text{otherwise ,} \end{cases}$$

$$\mathcal{G}(t+1) = \begin{cases} \mathcal{L}^k(t+1) & \text{if } f\left(\mathcal{L}^k(t+1)\right) \geq f\left(\mathcal{G}(t)\right), \\ \mathcal{G}(t) & \text{otherwise} \end{cases}$$

$$u(t,n) = \begin{cases} \omega_0 F(t,n), n \in A \\ \dfrac{1}{R_{t,n}^{(u)}} + \dfrac{\alpha_0}{R_{t,n}^{(d)}} + \omega_0 F(t,n), n \notin A \end{cases}$$

$$\left(\frac{1}{n}\right) * \sum \left(\min_j d^2\left(X_i, m_j\right) \right), \text{fori } = 1 \text{ ton}$$

References: Jagadeeswari, V., Subramaniyaswamy, V., Logesh, R., & Vijayakumar, V. (2018). A study on medical Internet of Things and Big Data in personalized healthcare system. Heal Inf Sci Syst, 6(1), 1–20.

Novelty: Monitoring and Updating, Greedy Selection, Lack of Complete Details, etc.

15.

$$\left[y_i f_i = f\left(\binom{n}{i}\sum(xi, wi)\right)\right]$$

$$[f_x = \tanh(x) = [2/1 + \exp(-2x)] -]$$

$$ETx(k,d) = ETx - \mathrm{Elec}(k) + ETx - \mathrm{amp}(k,d)$$

$$ERx(k,d) = ERx - \mathrm{Elec}(k)$$

$$ERx(k) = \mathrm{Eelec} \times k$$

$$P^L(f) \infty f^k$$

$$P^L(f,d) = PLo + 10\ \mathrm{nlog}\ 10d/\ \mathrm{do}\ + X\alpha$$

$$P_o^L = 10\log 10$$

$$(4\Phi \times d \times f)c^2$$

1 Convert the data type of h0 to integer;

2 if h0∈[1,N−1]does not hold then

3 verification fails;

4 endif

5 element t=gh0inGT;

6 integer h=H2(M||w,N);

7integer l=(r−h)mod N;

8 if l=0 then

9 gotostep2;

10 endif

11 integer h1=H1(IDe||hid,N);

12 element P=[h1]P2+Ppub−sinG2;

13 element u=e(S0,P) in GT;

14 w0=ut in GT;

15 converts the data type of w0 into a bit string;

16 integerh2= H2(M0||w0,N);

17if h 2=h0 holds then

18 verification success;

19 else

20 the verification fails;

21endif Output: Verification result: succeed or fail

References: Ali, A.; Almaiah, M.A.; Hajjej,F.;Pasha,M.F.;Fang,O.H.; Khan,R.;Teo,J.;Zakarya, M.An Industrial IoT-Based Blockchain-Enabled Secure Searchable Encryption Approach for Healthcare Systems Using Neural Network

Novelty: Verification Process, Use of Cryptographic Functions, Use of Cryptographic Functions, Output and Result Reporting, etc.

CONCLUSION:

To sum up, the incorporation of AI methods into safe communication channels inside Fog Computing settings has enormous promise for resolving the always increasing security issues in distributed computing. The necessity of utilizing AI algorithms to improve data privacy, security, and integrity in fog-based communication is highlighted in the abstract that is being given. Fog Computing designs may take use of AI's intelligent decision-making, anomaly detection, and real-time threat analysis capabilities. Fog nodes can decide wisely on data transmission and access control thanks to AI algorithms' ability to efficiently identify and mitigate possible security issues.

Furthermore, fog-based networks may be made more resistant to eavesdropping, data breaches, and unauthorized access with the use of AI-enabled secure communication methods. As a result of AI algorithms' proactive character, fog nodes are less vulnerable and sensitive data is kept intact. Continuous monitoring and adaptive reactions to new threats are made possible by this proactive nature of AI algorithms. It is crucial to stress that the implementation of AI-enabled safe communication mechanisms in fog computing must be complemented by reliable encryption protocols, powerful authentication systems, and thorough security rules. These components working together will produce a comprehensive strategy for protecting data in fog-based settings.

The use of AI approaches in secure communication protocols will play a crucial role in assuring the secrecy, integrity, and availability of data as fog computing

continues to develop and broaden its applications in several fields. Future research should concentrate on building standard security frameworks, improving AI algorithms, and addressing the ethical issues related to AI-enabled security systems. In conclusion, the use of AI-enabled secure communication protocols in fog computing presents exciting opportunities to reduce security risks and protect data in a dispersed computing environment. By adopting this paradigm, businesses may strengthen their data privacy and security safeguards, laying the foundation for the secure and resilient mainstream adoption of fog computing.

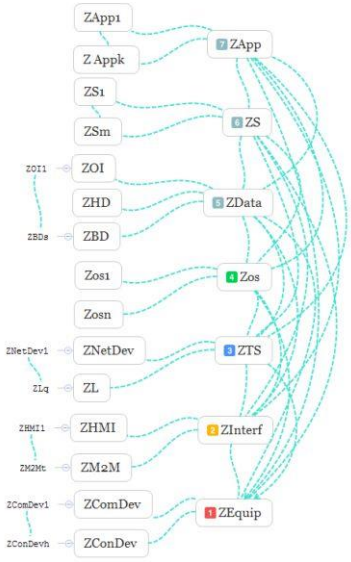
REFERENCES:

1. Wang, J., Li, P., & Hu, J. (2019). An AI-Based Communication Security Mechanism in Fog Computing. IEEE Access, 7, 88026-88037.
2. Liu, H., Zhang, C., & Shen, X. (2018). Secure Communication Mechanism in Fog Computing: Challenges, Solutions, and Future Directions. IEEE Internet of Things Journal, 5(2), 1183-1194.
3. Singh, D., Sharma, D., & Garg, S. (2020). AI-Enabled Secure Communication in Fog Computing: A Survey. In Proceedings of the International Conference on Machine Learning, Big Data, Cloud Computing and Parallel Computing (pp. 57-62).
4. Zhu, F., Zhang, Y., & Jiang, Y. (2019). AI-Enabled Secure Data Transmission in Fog Computing: Challenges and Solutions. Future Generation Computer Systems, 92, 739-751.
5. Shah, M. S., Xia, F., & Rahimian, M. A. (2019). AI-Enabled Security Framework for Fog Computing: Challenges and Opportunities. Journal of Network and Computer Applications, 129, 1-17.
6. Wang, L., Wang, X., & Yao, Y. (2020). A Secure Communication Protocol for Fog Computing based on Blockchain and AI. Journal of Ambient Intelligence and Humanized Computing, 11(8), 3569-3584.
7. Y. Chen, T. Krishna, J. S. Emer, and V. Sze, “Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks,”
8. A. Amravati, S. B. Nasir, S. Thangadurai, I. Yoon, and A. Raychowdhury, “A 55nm time-domain mixed-signal neuromorphic accelerator with stochastic synapses and embedded reinforcement learning for autonomous micro-robots,”

9.] S. Choi, J. Lee, K. Lee, and H. Yoo, “A 9.02mw cnn-stereo-based realtime 3d hand-gesture recognition processor for smart mobile devices,”
10. G. Desoli, N. Chawla, T. Boesch, S. Singh, E. Guidetti, F. D. Ambroggi, T. Majo, P. Zambotti, M. Ayodhyawasi, H. Singh, and N. Aggarwal, “14.1 a 2.9tops/w deep convolutional neural network soc in fd-soi 28nm for intelligent embedded systems,”
11.] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, “Machine learning for internet of things data analysis: A survey,”
12. W. Shi and S. Dustdar, “The promise of edge computing,”
13. Y. N. Krishnan, C. N. Bhagwat, and A. P. Utpat, “Fog computing — network based cloud computing,”
14. Islam, M. M., Ahmed, S. F., & Hasan, M. K. (2020). AI-Enabled Secure Communication in Fog Computing: A Comprehensive Survey. Journal of Ambient Intelligence and Humanized Computing, 11(8), 3477-3495.
15. Ren, J., Liu, K., & Jia, W. (2019). An AI-Based Secure Communication Framework for Fog Computing. In Proceedings of the 18th IEEE International Conference on Communication Technology (pp. 124-128).
16. Cheng, H., Li, J., & Wang, J. (2018). A Lightweight AI-Based Secure Communication Mechanism for Fog Computing. In Proceedings of the International Conference on Wireless Communications and Signal Processing (pp. 1-5).
17. Chen, Z., Wang, C., & Chen, C. (2020). AI-Enabled Secure Communication in Fog Computing: A Comprehensive Survey. In Proceedings of the International Conference on Advances in Artificial Intelligence (pp. 24-37).
18. P. K. Pankaj Sareen, "The Fog Computing Paradigm,"
19. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things,"
20. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update,"

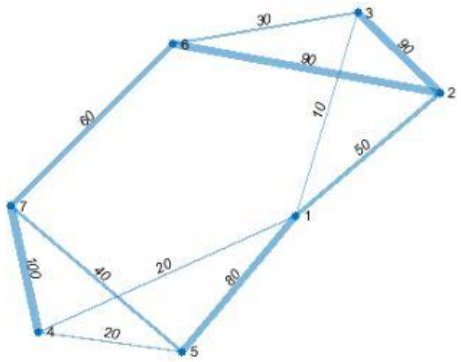
GRAPHS:

01)



1st level – equipment. The vertices of the graph $ZEquip$, $ZConDev$, ..., $ZComDev$, ... correspond to this level. The vertex of $ZEquip$ defines all problems which are solved at this level; vertices of $ZConDev$, ... – (connecting devices) tasks that are solved at the level of a wide range of devices connected to the network; vertices $ZComDev$, ... – tasks for devices that allow you to perform the necessary calculations. An example of the tasks that are solved at this level is the calculation of the reliability and performance of devices.

02)



2nd level – interfaces. As mentioned above, Fog calculations are best suited for working with inter-machine interaction systems – M2M, and devices that use a human-machine interface – HMI. The interface level corresponds to the vertices of the graph $ZI\ nter\ f$, $ZHMI$, ..., $ZM2M$, ... The vertex of the $ZI\ nter\ f$ describes the tasks specific to this level; vertices $ZHMI$, ..., $ZM2M$, ... – define tasks for systems with inter-machine and human-machine interface, respectively. M2M – machine-machine interaction (Machine-to-Machine, Mobile-to-Machine, Machine-to-Mobile) – the name of the technology (sum of technologies), which allows data transfer between different devices, and it can be groups of devices, such as public transport.

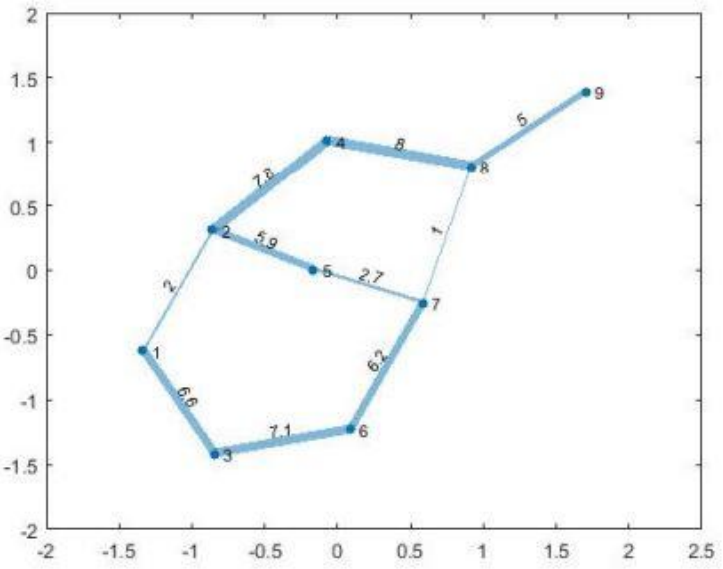
Work on M2M is coordinated by the following organizations: the Eclipse Foundation, the Focus Groupon Machine-to-Machine, a member of the International Telecommunication Union, and the TR-50 M2M Intelligent Devices Engineering Committee.

Types of M2M: stationary M2M, such as process control, payment terminals, meters, etc., and mobile M2M, for example, for fleet management, where M2M is used as an on-board device for monitoring, diagnostics, navigation, security, and mobile communications. M2M applications include access systems, premises security systems, security systems, remote control and management of equipment, transport and monitoring of moving objects,

vending machines, payment terminals, healthcare, etc. HMI – human-machine interface – a concept that encompasses engineering solutions that provide human interaction with controlled objects (machines, systems, devices).

The HMI can be a computer, standard software, a simple remote control with a set of LED indicators, and so on. Modern computers are focused on streaming architecture, the implementation of intelligent human-machine interface, which provides not only a systematic solution but also the ability of the machine to logical thinking and self-learning, to associative information processing and drawing logical conclusions. The requirements that different users impose on the HMI vary widely. The implementation of an intelligent human-machine interface is associated with the ability to solve problems of recognition and understanding of natural language, for this, there are recognition systems (language, handwritten texts, images). Creating a user-friendly and efficient human-machine interface is an urgent task. Also among the tasks of the interface level can be distinguished, for example, the choice of standard interface buses, reliability assessment at the interface level, and others.

03)



3rd level – transport system (network). The transport system is used to transmit information and contains nodes of the Fog infrastructure – switches,

routers, etc. The vertex $ZT\ S$ of the graph G = (Z , L) defines the general tasks characteristic of the 3rd level, and the vertices $ZN\ etDev$, ... – describe the tasks that are solved at the level of network devices; vertices ZL, ..., ZLq, ... – tasks that are solved at the level of communication channels. As the tasks are solved at this level it is possible to result in the following – a choice of a communication channel; channel bandwidth estimation; calculation of the delay factor of network equipment; estimation of message delay and many others.

4th level – operating systems (OS). Here you should consider the presence of different types of operating systems (UNIX-like OS, Windows, macOS, etc.). The vertex Zos of the graph describes the general tasks characteristic of the OS level. Vertices $Zos1$, ..., $Zosn$ are tasks that are solved for each specific operating system, for example:

• calculation of the coefficient of relative losses of OS performance for a multiprocessor system,

• determining the average processing time of the OS request,

• estimation of the average time spent on access to external memory and analysis of the

intensity of OS requests to external memory devices,

• assessment of the reaction time of the OS in solving specific problems,

• an estimate of the average time required to transmit the OS request,

• estimate the time of access to RAM,

• optimization of the core structure of open OS by the criterion of information security,

• estimation of time of detection of errors in processes,

• calculation of the probability of skipping the controlled signal (quantitative characteristics for the tasks of monitoring the integrity of OS files) and many others.

5th level – data. The vertices $ZData$, ZOI , ..., ZHD, ..., ZBD, ... of the graph G = (Z , L) correspond to this level. The $ZData$ vertex describes general tasks, ZOI , ..., ZHD, ..., ZBD, ... vertices – tasks specific to operational information (real-time analysis), historical data (transaction analysis), and long-term storage (BigData analysis).

Examples of tasks to be solved at this level:

• prognostic calculation of the speed of new data generation,

• optimization of file placement and processing of requests to the database,

• estimation of data volume,

• data compression,

• distributed calculations when planning requests to the database,

• assessment of the integrity of information at the level of links and other tasks.

6th level – services. These can be various services, such as online services (like Uber), streaming services (like Netflix, Amazon Prime, Hulu, and Crunchyroll), etc.

The level of services corresponds to the vertices ZS , $ZS1$, ..., ZSm graph model. Vertex ZS defines general tasks for service level, vertices $ZS1$, ..., ZSm – tasks for different types of service.

Examples of tasks:

• calculation of productivity for the 6th level,

• assessment of service quality in virtual VPN channels,

• optimization of system services by network resources,

• assessment of the security of transmission of confidential information in broadcast communication channels,

• maximum support for different types of 6th level traffic, etc.

7th level – applications. Applications are research software, computer-aided design systems, games, applications for artists, geographically distributed

applications for pipeline monitoring, smart devices in the car, SmartGrid, traffic light control systems, etc.

The vertices $ZApp$, $ZApp1$, ..., $ZAppk$ of the graph $G = (Z , L)$ correspond to this level. Vertex $ZApp$ describes the general tasks of the application level, the vertices of $ZApp1$, ..., $ZAppk$ – tasks for different types of applications. These can be applications installed on computers, tablets, smartphones of users.

Some examples of tasks:

• calculation of maximum productivity for the 7th level,

• distribution of application tasks between users according to the criterion of weighted average route length,

• prognostic estimate of the conditional average service time of the application required to perform the task lasting in nt period,

• estimation of the average time of the decision of applied problems,

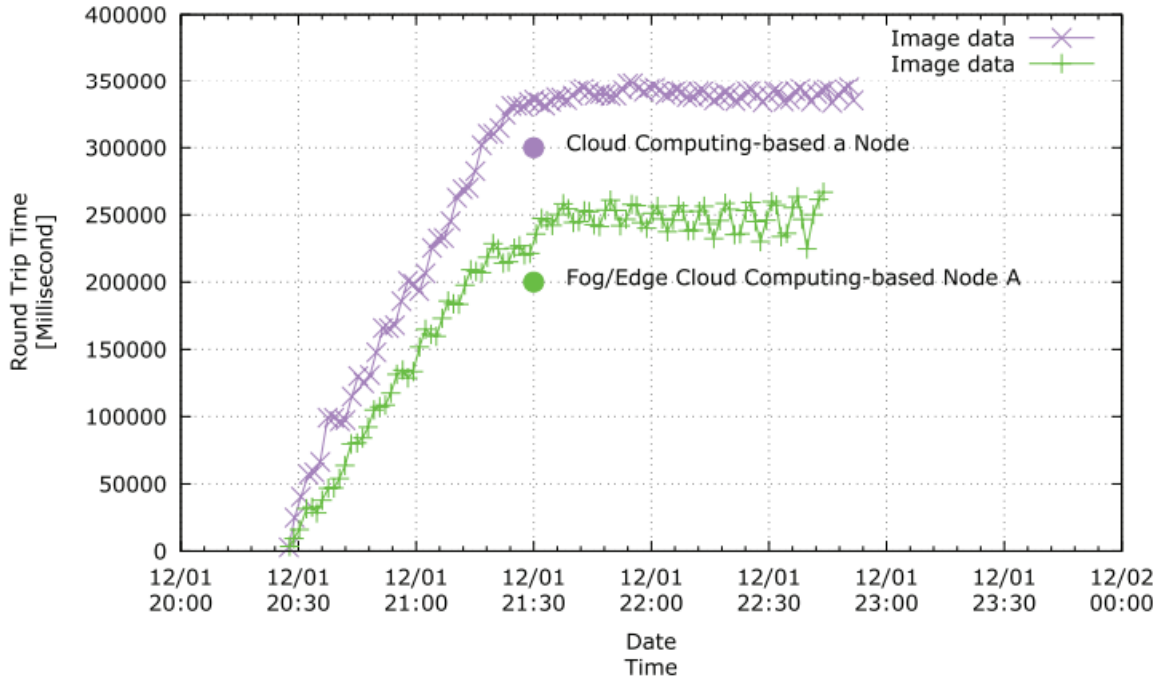
• calculation of exchange time with external memory in the process of solving applied corporate tasks,

• prognostic calculation of the time required for data processing in the application system,

• calculation of the average service time of the application for algorithms of non-priority service disciplines,

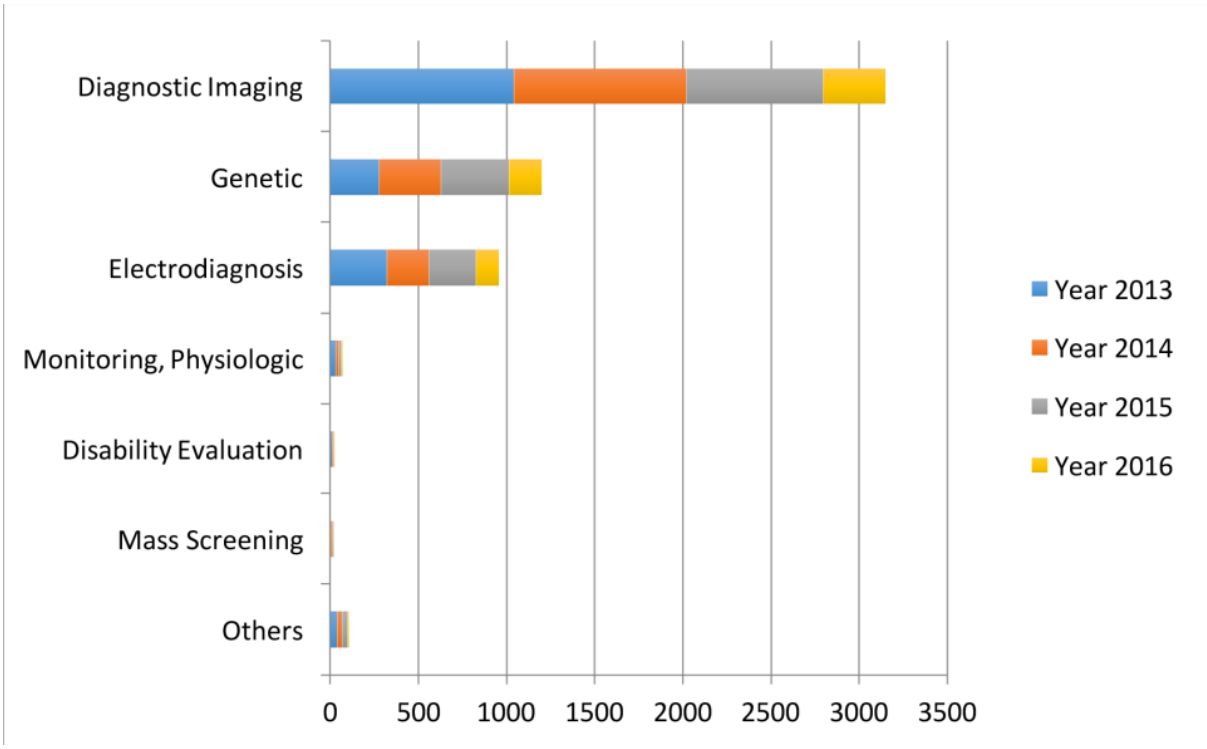
• scalability for the 7th level

04)



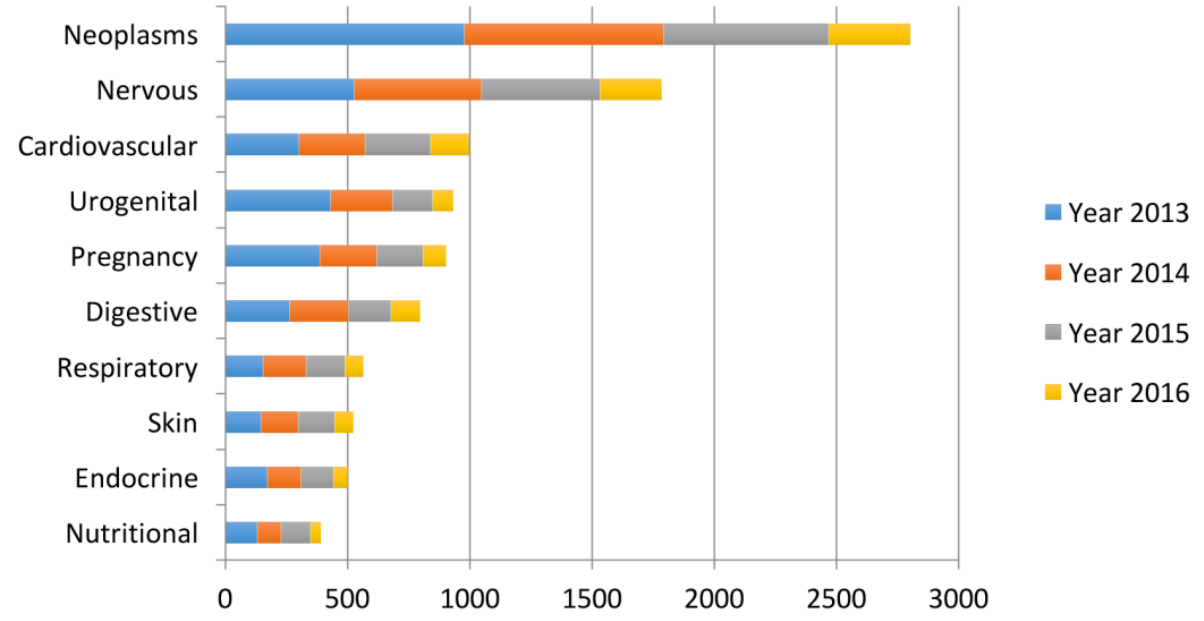
Machine learning latency, the time delay or response time between making a prediction or inference using a machine learning model and receiving the result. It is an important consideration in various applications where real-time or near-real-time responses are required. Latency in machine learning can be influenced by several factors. One of the primary factors is the complexity and size of the model itself. Larger and more complex models generally require more computational resources and, consequently, may have higher latency. This is particularly relevant for deep learning models with numerous layers and parameters.

05)



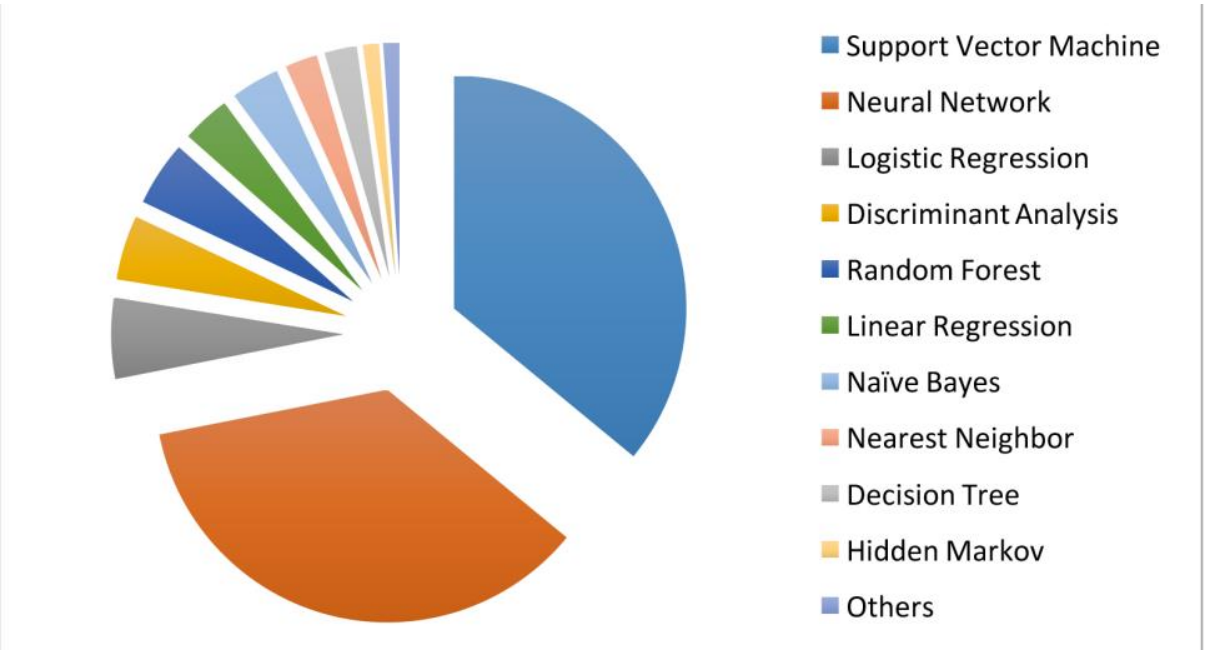
The data types considered in the artificial intelligence artificial (AI) literature. The comparison is obtained through searching the diagnosis techniques in the AI literature on the PubMed database.

06)



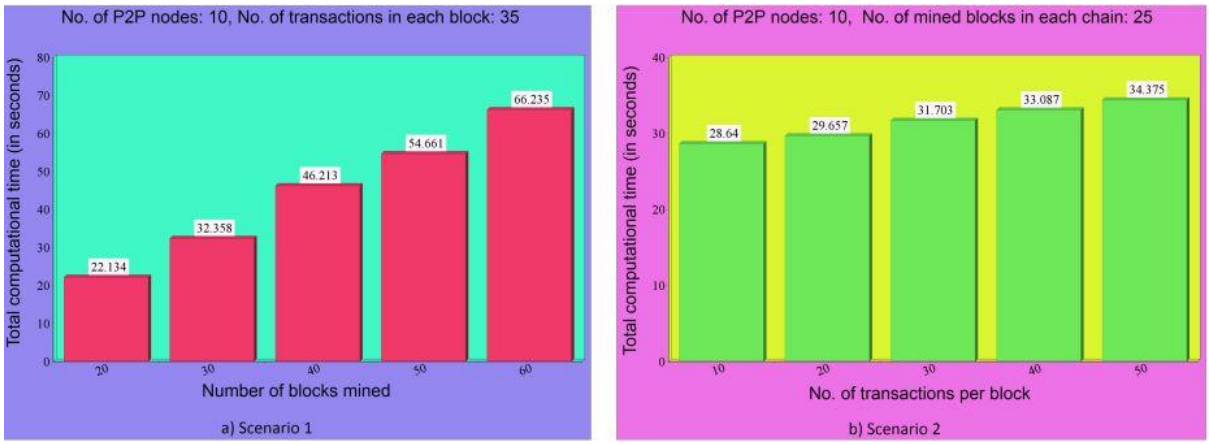
The leading 10 disease types considered in the artificial intelligence (AI) literature. The first vocabularies in the disease names are displayed. The comparison is obtained through searching the disease types in the AI literature on PubMed.

07)



The machine learning algorithms used in the medical literature. The data are generated through searching the machine learning algorithms within healthcare on PubMed.

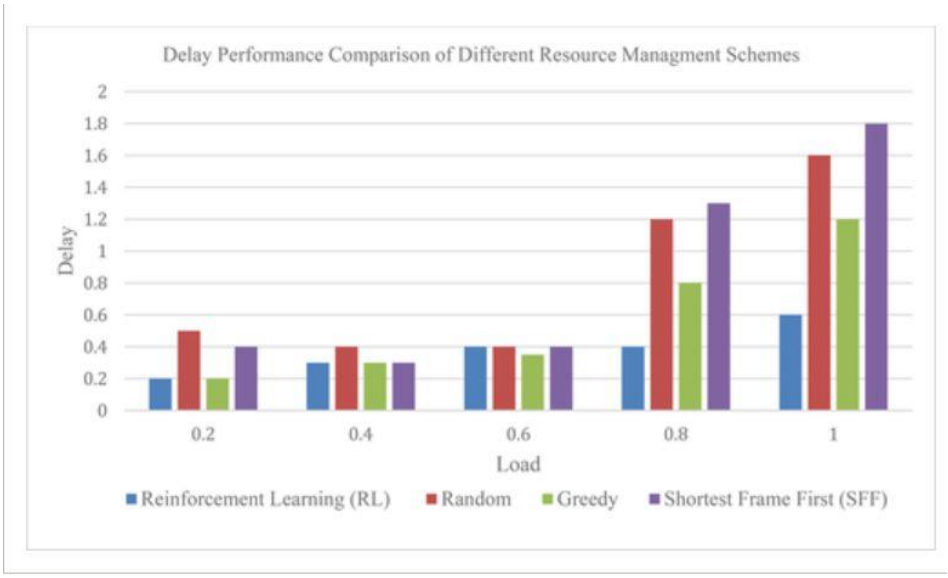
08)



The number of transactions per mined block is taken as 35 in this scenario. The simulation results shown in Figure demonstrate that the number of blocks mined into the blockchain versus the total computational time (in seconds) for mining the blocks. It is noticed that if the number of blocks mined is increased, the computational time also increases linearly. The number of mined blocks in each chain is considered as 25. The simulation results provided in Fig. 6 show the number of transactions stored per block versus the total computational time (in seconds) for mining the blocks.

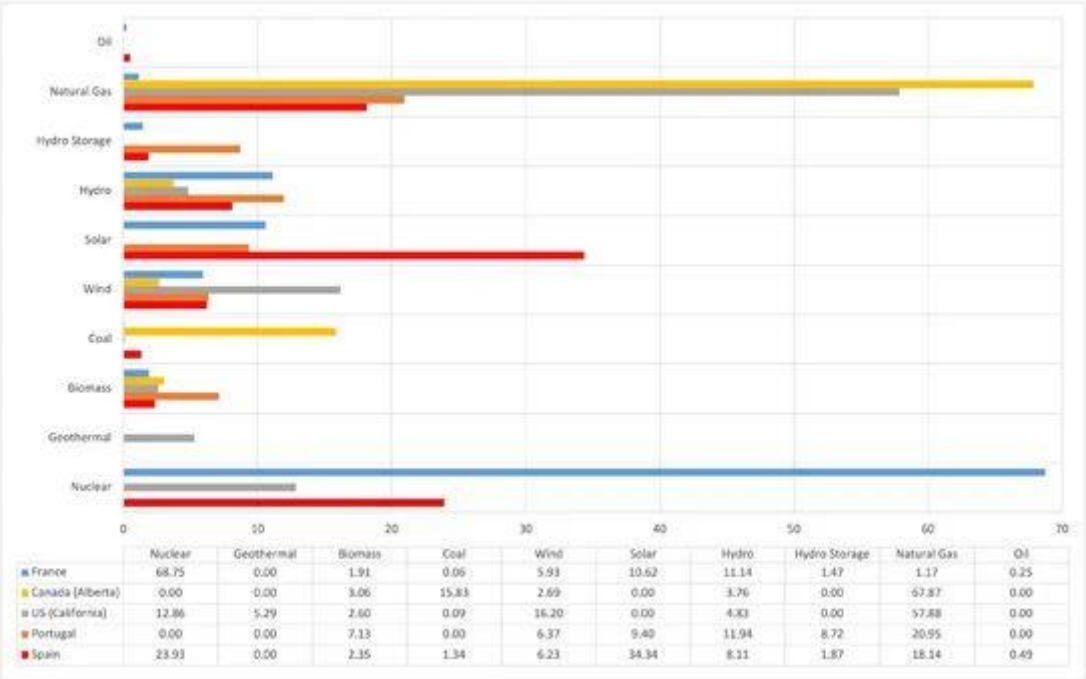
Similar trend shows that the computational time increases linearly when the number of transactions per block is also increased.

09)



AI technologies such as machine learning (ML) and reinforcement learning (RL) offer unique solutions to address reliability problems in low-latency vehicle communication networks when applied to 5G and beyond. The lack of resources causes a particularly important problem concerning the radio spectrum, as it affects traditional scheduling decisions for real-time resources with serious delays. Several researchers presented studies that investigate the problem of delay reduction with spectrum and energy constraints in the network of vehicle networks based on the fifth generation with the aim of reducing the delay caused by high load and lack of radio resources. In the study presented by Huang et al. it was found that the use of reinforcement learning (RL) technology helps in developing a scheme for allocating spectrum and managing energy resources [98]. According to the presented study, it was found that this technique gives an appropriate approximation to the learning process in the allocation of energy resource.

10)



	Nuclear	Geothermal	Biomass	Coal	Wind	Solar	Hydro	Hydro Storage	Natural Gas	Oil
France	68.75	0.00	1.91	0.06	5.93	10.62	11.14	1.47	1.17	0.25
Canada (Alberta)	0.00	0.00	3.06	15.83	2.69	0.00	3.76	0.00	67.87	0.00
US (California)	12.86	5.29	2.60	0.09	16.20	0.00	4.83	0.00	57.88	0.00
Portugal	0.00	0.00	7.13	0.00	6.37	9.40	11.94	8.72	20.95	0.00
Spain	23.93	0.00	2.35	1.34	6.23	34.34	8.11	1.87	18.14	0.49

The carbon intensity (in grams of emitted CO2 per KWh). This latter parameter can be obtained through the data published publicly by many countries or by organizations such as the European Union, but it is easier to obtain it from Electricity Maps [129], an open-source project that collects such data automatically and plots them through a user-friendly interface. Such a website also indicates the energy sources used by each country (an example of such sources for France, Portugal, Spain, California, and the province of Alberta is shown in Figure).

11)

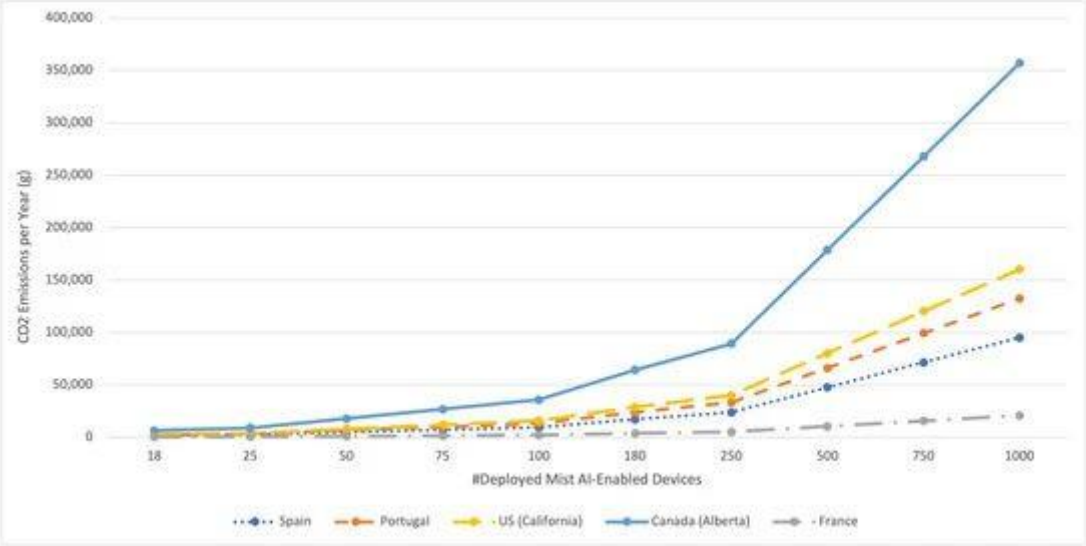
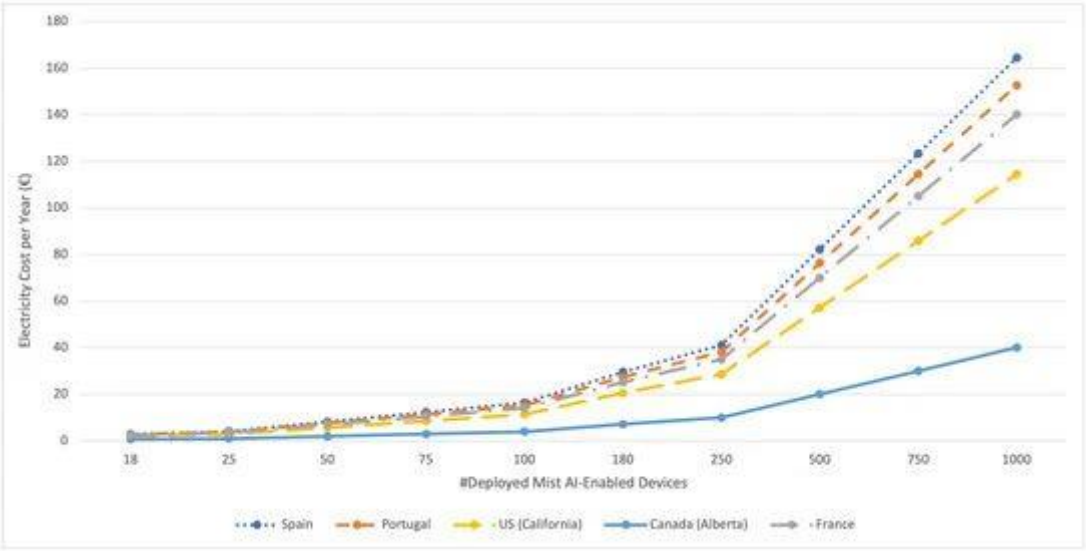


Figure shows the estimated CO2 emissions for the energy consumption estimated in the previous section. As it can be easily guessed, emissions increase with the number of deployed mist AI-enabled devices; however, such growth changes dramatically from one country to another depending on the energy source: while near-zero emission countries like France are barely impacted by the increase in the number of deployed devices, a province like Alberta emits more than 17 times more CO2 for 1000 deployed devices.

12)



It is also possible to obtain the monetary cost of running the mist AI-enabled devices (as an example, the average prices for April 2021 for each territory were considered), which is depicted in Figure . As it can be seen in the figure, the cost of running the system in Alberta would be cheaper but will result in more CO2 emissions. In contrast, the countries with the largest shares of renewable energy sources (Spain and Portugal) are the ones with the most expensive electricity. Nonetheless, please note that such a link between the use of renewable energies and cost is impacted by other external factors (e.g., taxes, environmental policy, and energy trading).

[1] T. Kramp, R. van Kranenburg, S. Lange, Introduction to the Internet of Things, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 1–10.

[2] O. Markova, S. Semerikov, A. Striuk, H. Shalatska, P. Nechypurenko, V. Tron, Implementation of cloud service models in training of future information technology specialists, CEUR Workshop Proceedings 2433 (2019) 499–515.

[3] S. Yi, Z. Hao, Z. Qin, Q. Li, Fog computing: Platform and applications, in: 2015 Third IEEE workshop on hot topics in web systems and technologies, HotWeb, 2015, pp. 73–78.

[4] H. Atlam, R. Walters, G. Wills, Fog computing and the internet of things: A review, Big data and cognitive computing 2 (2018) 10.

[5] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE Internet of things journal 3 (2016) 637–646.

[6] N. Abbas, Y. Zhang, A. Taherkordi, T. Skeie, Mobile edge computing: A survey, IEEE Internet of Things Journal 5 (2017) 450–465.

[7] M. Satyanarayanan, The emergence of edge computing, Computer 50 (2017) 30–39.

[8] L. Vaquero, L. Rodero-Merino, Finding your way in the fog: Towards a comprehensive definition of fog computing, ACM SIGCOMM Computer Communication Review 44 (2014) 27–32.

[9] R. Mahmud, R. Kotagiri, R. Buyya, Fog computing: A taxonomy, survey and future directions, Internet of everything (2018) 103–130.

[10] I. Stojmenovic, S. Wen, X. Huang, H. Luan, An overview of fog computing and its security issues, Concurrency and Computation: Practice and Experience 28 (2016) 2991–3005.

[11] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, 2012, pp. 13–16.

[12] S. Sarkar, S. Misra, Theoretical modelling of fog computing: a green computing paradigm to support iot applications, IET Networks 5 (2016) 23–29. doi:10.1049/iet-net.2015.

[13] N. Verba, K. Chao, A. James, J. Lewandowski, X. Fei, C. Tsai, Graph analysis of fog computing systems for industry 4.0, in: 2017 IEEE 14th International Conference on e-Business Engineering (ICEBE), 2017, pp. 46–53.

[14] I. Lera, C. Guerrero, C. Juiz, Availability-aware service placement policy in fog computing 43 based on graph partitions, IEEE Internet of Things Journal 6 (2018) 3641–3651.

[15] S. Ningning, G. Chao, A. Xingshuo, Z. Qiang, Fog computing dynamic load balancing mechanism based on graph repartitioning, China Communications 13 (2016) 156–164.

[16] S. Yi, C. Li, Q. Li, A survey of fog computing: concepts, applications and issues, in Proceedings of the 2015 workshop on mobile big data, 2015, pp.

[17] X. Chen, J. Zhang, When d2d meets cloud: Hybrid mobile task offloading in fog computing, in: 2017 IEEE international conference on communications (ICC), 2017.

[18] D. Korzun, A. Varfolomeyev, A. Shabaev, V. Kuznetsov, On dependability of smart applications within edge-centric and fog computing paradigms, in: 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), 2018,

[19] I. Lera, C. Guerrero, C. Juiz, Yafs: A simulator for iot scenarios in fog computing, IEEE Access 7 (2019).

[20] T. H. Szymanski, 300 pseudo-random task graphs for evaluating mobile cloud, fog and edge computing systems, 2018

[21] M. Iorga, L. Feldman, R. Barton, M. Martin, N. Goren, C. Mahmoudi, Fog computing conceptual model, Natl. Inst. Stand. Technol. Spec.