FOREST FIRE IMAGE CLASSIFICATION THROUGH DECENTRALIZED DNN INFERENCE

Dimitrios Papaioannou[1], *Vasileios Mygdalis*[2] *and Ioannis Pitas*[1]

[1]Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
[2]Faculty of Business and Economics, University of Antwerp, Antwerpen, Belgium
[1]{dpapaion, pitas}@csd.auth.gr, [2]Vasileios.Mygdalis@uantwerpen.be

ABSTRACT

In the realm of Natural Disaster Management (NDM), timely communication with local authorities is paramount for an effective response. To achieve this, multi-agent systems play a pivotal role by proficiently identifying and categorizing various disasters. In the field of Distributed Deep Neural Network (D-DNN) inference, such approaches often require DNN nodes to transmit their results to the cloud for inference, or they necessitate the establishment of a fixed topology network to enable inference directly on the edge, a practice prone to security risks. In this work, we propose a decentralized inference strategy tailored for fire classification tasks. In this approach, individual DNN nodes communicate within a network and enhance their predictions by considering other DNN node inference outputs that contribute to improving their individual performance. The overall coordination of the system on a specific decision is achieved through a consensus protocol, which acts as a universally accepted inference rule adopted by all DNN nodes operating within the system. We present a comprehensive experimental analysis, of the forest-fire classification task, focusing on enhancing both individual DNN node performance and the stability of the consensus protocol.

***Index Terms*—** Natural Disaster Management (NDM), Fire classification, Multi-Agent Systems, Decentralized DNN Inference, BFT Consensus Protocol

1. INTRODUCTION

Natural disasters, such as fires in forested areas, floods, and earthquakes, are a major issue plaguing modern society. In such circumstances, the destruction caused to human lives, properties, and national infrastructure is devastating and often associated with significant economic repercussions [1]. The prompt communication of essential information to local authorities regarding these phenomena is crucial to minimize exposure to dangers and mitigate the potentially destructive consequences. Multi-agent systems (MAS) can therefore be used for effective Natural Disaster Management (NDM) offering the capability to swiftly identify and categorize various disasters [2]. Typically, a MAS comprises multiple autonomous agents gathering data from various sources such as satellites, UAVs (Unmanned Aerial Vehicles), social media, and smart cameras. These agents communicate their findings with one another, within a network, collectively assess the situation, and prioritize response actions or coordinate emergency resources effectively [3]. The overall coordination of the system is commonly organized by centralized authorities, such as a master node or a robust cloud server, which can potentially give rise to significant security vulnerabilities or privacy implications.

In deep learning problems, many current approaches revolve around architectures operating within an edge-to-cloud DNN inference framework [4]. In this scenario, edge devices act as autonomous agents, collecting and processing data with limited capacity, while the results are transmitted to the cloud for the decision-making process. Subsequently, the outcome is communicated and distributed back to the edge devices. In such settings, various approaches involve techniques for compressing and pruning large-scale models to enable partial inference on the device while the remaining processing is executed in the cloud [5]. Similarly, other approaches entail model selection techniques wherein multiple DNNs are trained and stored in the cloud. Depending on the input data, a dynamic function is utilized to search for and execute the best model for the specific task [6]. Conversely, on-device inference primarily focuses on approaches such as computation offloading, wherein a multilayered DNN is partitioned among multiple nodes to form a larger network during inference [7]. Regardless of the approach used, none of the strategies mentioned above facilitate a fully decentralized inference process without the need for any centralized coordination, ensuring a properly operating protocol in which none of the participating devices can be considered reliable.

In simple terms, the strategies outlined above predominantly depend on mutual trust among participating DNN nodes, presuming their consistent and reliable operation under all circumstances. However, this reliance poses a considerable risk of failure due to potential factors such as system crashes, computational errors, or malicious attacks. In such scenarios, certain DNN nodes may present a façade of normalcy while simultaneously engaging in activities aimed at subverting and compromising the integrity of the system.

These concerns are particularly significant for systems requiring utmost reliability, especially in the context of NDM processes.

Motivated by these challenges, our study introduces a decentralized fire classification pipeline operating in two stages. Firstly, we employ a node-to-node model selection technique, enabling each autonomous DNN node to enhance its performance through consultation with neighboring DNN nodes. Secondly, we propose a novel consensus protocol designed to serve as a universally accepted inference rule for all participating DNN nodes within the system. In our experiments focused on fire classification, we observed a notable enhancement in the performance of individual DNN nodes. The proposed consensus protocol successfully functions as a universally accepted inference rule among participating DNN nodes, resulting in an overall accuracy score surpassing those obtained through typical centralized ensemble aggregation techniques such as majority voting and weight averaging [8].

The rest of the paper is structured as follows: In Section 2, a comprehensive review of distributed and decentralized inference architectures is provided. Section 3 describes the proposed Decentralized Fire Classification Pipeline. Experimental results are discussed in Section 4. Finally, a conclusion is provided in Section 5.

2. DISTRIBUTED AND DECENTRALIZED INFERENCE

In recent years, several methodologies have emerged for distributed DNN inference strategies, with a focus on architectures tailored for mobile and IoT services. These architectures typically employ edge-to-cloud pipelines, utilizing edge devices like smartphones and smart cameras for data collection [9]. In some cases, these devices also perform partial data processing to preserve privacy [10, 11, 12]. However, this approach introduces data transferring delays due to the frequent transition of high-dimensional intermediate representations between edge devices and the cloud. Additionally, there's a reliance on a dedicated server to oversee the entire process, posing a risk of compromise as it becomes a single point of failure, threatening the integrity of the system.

In decentralized settings, edge devices, exchange information directly with each other leveraging peer-to-peer (P2P) or gossiping communication links, to collaborate and produce results without requiring any form of centralized coordination. The Edge Ensembles [13] is a collaborative inference approach wherein DNN nodes within the system utilize a trainable shared encoder to encode and quantify input data. These quantified features are then transmitted and shared among the other DNN nodes in the system. Upon encountering features sent by another DNN node, a DNN node utilizes its local decoder model to communicate the mapping and aggregates it with the feature representation it has produced locally. While we acknowledge that such approaches seems to be heading in the right direction, they are not strictly inference-based methods. Also, they do not conduct model selection to filter and streamline the inference process, nor are they entirely fault-tolerant, as they lack the capability to effectively identify and exclude malicious users seeking to compromise the entire process.

Architectures like the Danku protocol [14] function as layer 2 solutions on the Ethereum blockchain, utilizing smart contracts in an endeavor to offer an efficient highly-secured decentralized exchange environment of trainable models, on which the best performing one is selected by the committe. In the Danku protocol, each node functions as a DNN node bound to a designated baseline model, responsible for training and sharing its knowledge with other DNN nodes. Ultimately, the protocol selects the best-performing model, and the node responsible receives a reward for its honest contributions. The Ethereum blockchain's inherent trust serves as collateral, ensuring integrity and fostering honest participation within the system. Building on top of layer 1 blockchain protocols indeed holds promise, but it comes with inherent risks, such as high gas fees and limitations on the amount of data that can be processed and written directly to the blockchain.

Distributed consensus protocols, originating from the Byzantine Generals' Problem introduced by Lamport et al. [15], have long been employed to ensure agreement among nodes in systems operating over unreliable communication links. Protocols like Practical Byzantine Fault Tolerance (PBFT) [16] have been extensively studied and form the basis for modern consensus mechanisms such as BFT-Smart [17], DR-BFT [18], and Raft [19]. While widely used in private blockchain systems like Hyperledger [20], these protocols have seen limited integration into distributed deep learning systems. This is largely due to their inefficiency during training, where state machine replication increases computational complexity by requiring the saving and verification of system states at each training epoch. However, in distributed DNN inference systems, this complexity is mitigated as the focus is on inference rather than training, reducing the computational overhead associated with state verification.

3. METHODOLOGY

In this section, we provide a detailed description of the proposed decentralized DNN fire classification pipeline. Let $\mathcal{G} = \{\mathcal{A}, \mathcal{E}\}$ be a graph comprised of N collaborating DNN nodes defined in a set $\mathcal{A} = \{\alpha_1, \alpha_2, ..., \alpha_N\}$, that are equipped with a specific Convolutional Neural Network (CNN) model tailored to perform a fire inference classification task of the form $\hat{\mathbf{y}}_i = \mathbf{f}_i(\mathbf{x}; \boldsymbol{\theta}_i)$, where $\mathbf{x}$ is the data sample and $\boldsymbol{\theta}_i$ is the learning parameters of the i^{th}'s DNN node model. Assuming that $\mathcal{C} = \{c_1, \ldots, c_c\}$ is a set of classification labels, then each node's inference output are softened using a softmax activation function so that, $\sum_{j=1}^{C} \hat{\mathbf{y}}_{ij} = 1$, where $i \epsilon [0, N]$ and $j \epsilon [0, |\mathcal{C}|]$ to represent probability distributions. $\mathcal{E}$ is defined

as a set of fixed communication links allowing them to communicate with each other. It is assumed that all DNN nodes have obtained access to the same test sample $\mathbf{x}$, while their goal is to produce a single prediction $\hat{y} = \underset{j \epsilon \mathcal{C}}{\text{argmax}}\{\hat{\mathbf{y}}_j\}$. This methodology encompasses two integral components: a) *Individualized Model Selection Process (IMSP)* method in which each DNN node is requested to detect and eventually select the neighboring nodes that will indeed help him to improve his performance and b) *Quality of Inference (QoI) Consensus Protocol* in which all DNN nodes are requested to collaborate with each other in order to reach a common decision agreement about the content of a given sample, and thus serve as a single inference rule.

3.1. Individualized Model Selection Process (IMSP)

In this section, we introduce the Individualized Model Selection Process (IMSP), a crucial aspect of our decentralized fire classification framework. During the DNN inference stage, each node independently generates predictions based on the observed sample $\mathbf{x}$ and communicates them across the network. The communication protocol operates within predefined time intervals, denoted as t. Once a sufficient number of transmissions have occurred and adequate time has elapsed, each DNN node engages in an individual decision-making process. This process incorporates both the node's local observations and the information received from other nodes, leading to personalized model selections tailored to each node's perspective.

After the time interval t has expired, each DNN node α_i receives prediction vectors from the remaining $N-1$ nodes. Based on a predefined criterion, node α_i determines whether to integrate these DNN inference outputs into its final decision or discard them. Any node that fails to transmit its DNN inference output within the specified time delay t is automatically recognized as null. Additionally, if the prediction from node α_j deviates from the predefined format, it is considered null and disregarded by all other nodes. Nodes select their neighbors based on the following aggregation schemes:

- **Majority Voting-based Condition:** For a given sample $\mathbf{x}$, the DNN inference output of node a_j only takes into account nodes that are more confident than itself:

$$\hat{y}'_j = \text{mode}(b_i \hat{y}_i) \mid \hat{y}_i = \text{argmax}(\mathbf{f_i}(\mathbf{x}; \boldsymbol{\theta_i})), \forall \mathbf{i} \epsilon [\mathbf{0}, \mathbf{N}] \tag{1}$$

  where $b_i = 1$ if $\max((\mathbf{f_j}(\mathbf{x}; \boldsymbol{\theta_j})) < \max((\mathbf{f_i}(\mathbf{x}; \boldsymbol{\theta_i}))$ and $b_i = 0$, otherwise, and $\text{mode}(\cdot)$ is a function denote the number that is most commonly viewed in the given set.

- **Average-based Condition:** For a given sample $\mathbf{x}$, the DNN inference output of node a_j considers only the nodes that are more confident than itself for each class separately, i.e:

$$\hat{y}'_j = \frac{1}{N'} \sum_{i=1}^{N} b_i \hat{y}_i, \tag{2}$$

  where $b_i = 1$ if $\hat{y}_j < \hat{y}_i$ and $b_i = 0$, otherwise, and $N' \leq N$ is the number of DNN nodes outputs taken into account, based on the above condition. Additionally, by introducing a $\text{med}(\cdot)$ function, we can apply a median rule for the aggregation step formally defined as:

$$\hat{y}'_j = \underset{i \epsilon N}{\text{med}}(b_i \hat{y}_i), \;\; \forall b_i \neq 0. \tag{3}$$

3.2. QoI-BFT protocol

In this section, we introduce *Quality of Inference (QoI)* a novel consensus protocol designed to operate as a single inference rule, enabling coordination among individual DNN node decisions within a fully decentralized framework, thus eliminating the need for centralized coordination. The QoI protocol adapts the traditional Byzantine Fault Tolerance State Machine Replication (BFT SMR) principles, where a minimum of $N \leq 2f + 1$ nodes are required to tolerate f potentially faulty DNN nodes. These nodes, termed *honest*, aim to collectively agree on the content and order of predictions despite potential adversarial behavior. The protocol operates under a synchronous assumption, ensuring timely delivery of DNN inference outputs while minimizing communication delays. Each DNN node broadcasts its inference output to all neighboring nodes, maintaining a complete history of DNN inference outputs in the same order. QoI guarantees validity, agreement, integrity, and total order of predictions, ensuring that honest DNN nodes receive and agree upon predictions for all samples. QoI's core processes encompass view change, normal operation, and conflict decision agreement operations, enabling coordinated decision-making and handling conflicts that may arise during the inference process.

DNN nodes engage in a series of actions referred to as *views*. In the context of the QoI protocol, operational activities are organized into rounds, where each consensus round represents a single execution of the normal operational process, regardless of its success. Views delineate the consensus rounds necessary for the network to achieve agreement on a given sample. Let $\mathcal{V} = \{v_1, \ldots, v_v\}$ be the view set, views are described as elements of an index $v \in \mathcal{V}$, comprising a sequence of test pairs whose predictions are scheduled within the time interval t. During each view, one node acts as the *primary* DNN node, while the remaining $N-1$ nodes function as *validators*. For the remainder of this study, we simplify our approach by considering each 'view' as encapsulating a single sample alongside its class label, denoted as $(\mathbf{x}, y)$. Our objective is that every honest DNN node in N maintains an identical DNN inference history set defined as $\hat{\mathcal{Y}} = \{\hat{\mathbf{y}}_{\mathbf{vj}}, \; \forall \mathbf{v} \in \mathcal{V} \text{ and } \mathbf{j} \in \mathcal{C}\}$.

3.2.1. View Change

Leader Election. At any given moment, synchronization among all DNN nodes is essential, with each commencing from the same view. Initially, a primary node is designated from the set of DNN nodes, $\mathcal{A}$, to initiate the consensus process for the first round. Subsequently, the remaining $N-1$ nodes operate as validators. The primary node is elected in a cyclical manner, ensuring equal opportunity for all nodes to claim the primary role as long as they adhere to the consensus rules The election formula for determining the primary node is defined as:

$$a_p = v \bmod |\mathcal{A}|, \tag{4}$$

where $|\mathcal{A}| = N$ and $v \epsilon \mathcal{V}$ represent the current view we are currently working on.

View Change. When a misbehavior is detected in the primary DNN node of the current view, a view change is triggered to facilitate its replacement. Specifically, in the v^{th} view, the primary agent is promoting a DNN inference output for the i^{th} sample of the form:

$$\hat{y}_p = \text{argmax}\left(\mathbf{f_p}\left(\mathbf{x_i}; \boldsymbol{\theta_p}\right)\right). \tag{5}$$

The primary DNN node a_p communicates its DNN inference output $\hat{y}_p$ to the validators by constructing and broadcasting a pre-prepared message in a specific format (e.g., $\{a_p, \hat{y}_p, v_p, r_p\}$) where a_p is the primary id, $\hat{y}_p \in \mathcal{C}$ is its predicted value for the current sample, v_p is the view index and $r_p \epsilon \mathcal{R}$ is the rewards he has collected so far. Set $\mathcal{R} = \{r_1, \ldots, r_N$ is generated and locally maintained in the system of each DNN node and represents the collected rewards for each node.

Let $a_j \epsilon \mathcal{A}$ represent a random validator that has just received the primary's message. He calculates its prediction value as:

$$\hat{y}_j = \text{argmax}\left(\mathbf{f_j}\left(\mathbf{x_i}; \boldsymbol{\theta_j}\right)\right),\ a_j \in \{\mathcal{A} | a_j \neq a_p\}. \tag{6}$$

If its DNN inference predicted value $\hat{y}_j \neq \hat{y}_p$ or $v_j \neq v_p$ then, from now onwards, the j^{th} DNN node recognizes the primary as faulty. When a DNN node identifies the primary node as faulty, it promptly multicasts a view-change message to the remaining validators, adhering to a specific format (e.g., $\{a_j, v_j + 1, vo_j, r_j\}$. The parameter $vo_j = 1$ if $\hat{y}_j \neq \hat{y}_p$ or $vo_j = 0$ otherwise. Once the validators receive that view change messages, they append them to a local log. If $\frac{\sum_{i=1}^{N-1} vo_i}{|\mathcal{A}|} \geq 0.5$ then the primary is globally recognized as faulty since it has lost the favor of the majority. As a result of this failure in achieving consensus, DNN nodes transition to the new view, commencing the consensus round anew to reflect the updated perspective.

Reward System. The reward system provides incentives for primary DNN nodes to sustain the approval of the majority of validators. It rewards honest behavior with quality points q while penalizing the loss of majority favor. For a specific primary node a_p, the reward and penalty are computed as follows:

$$r_p = \begin{cases} q, & \text{if } \frac{\sum_{i=1}^{N-1} vo_i}{|\mathcal{A}|} < 0.5 \\ 0.5 r_p, & \text{if } \frac{\sum_{i=1}^{N-1} vo_i}{|\mathcal{A}|} \geq 0.5 \end{cases}. \tag{7}$$

If the primary DNN node fails to secure majority support, then a significant amount of the accumulated primary points are forfeited. This mechanism reinforces the importance of honest performance for sustained rewards. Conversely, failure to maintain majority favor jeopardizes the node's high-quality status, aiding in the identification of potentially faulty or malicious DNN nodes for exclusion from the decision-making process.

3.2.2. Normal Operation

In this stage, the designated primary DNN node evaluates decisions from validators, selecting pertinent ones, and assembling a final decision. This decision is then multicast to honest validators. For a primary DNN node a_p, let $\hat{\mathcal{Y}}_p = \{\hat{\mathbf{y}}_\mathbf{p}\}$ denote the set of the valid collected predictions so far. Then the set $\hat{\mathcal{Y}}_p$ is dynamically updated according to his observations as:

$$\hat{\mathcal{Y}}_p = \begin{cases} \hat{\mathcal{Y}}_p \cup \{\hat{\mathbf{y}}_\mathbf{j}\}, & \text{if } \hat{y}_p = \hat{y}_j \\ \hat{\mathcal{Y}}_p, & \text{otherwise} \end{cases}, \tag{8}$$

and the final decision is produced by combining the selected validators decisions using average 2 or median rule 3.

After the primary DNN node has reached a final decision, it multicasts an encrypted *prepare* message to all nodes, including itself. The message is structured as $\{a_p, \hat{y}_p, v_p, r_p\}$ where a_p is the primary's id, $\hat{y}_p$ is its final DNN inference output for the current sample, v_p is the view index and r_p is the collected so far rewards. Upon receiving a prepare message from the primary, each validator verifies its validity by confirming whether v_p matches its own view number and whether $\hat{y}_p$ aligns with its locally produced prediction. If the prepare message is indeed valid, it is transmits the prepared message to the rest validators. Validators await receipt of $2f + 1$ identical messages from different nodes to proceed to the commit phase. Here, a *commit* message of the form $\{a_j, \hat{y}_j, v_j, r_j\}$ message is sent. Once a validator confirms the validity of the commit message, it forwards it back to the primary. If the primary receives $2f + 1$ identical commit messages from different validators, it concludes that consensus has been achieved for that specific sample.

3.2.3. Conflict Decision Agreement

Conflict Decision Agreement emerges when the complexity of a sample surpasses the understanding of the majority of nodes. A sample for which the majority of nodes fail to reach

a consensus is termed a *conflict* sample. In such cases, DNN nodes are ordered based on the rewards $\mathcal{R}$ they have collected, in descending order, for the conflict sample i^{th} as $r_{j+1} < r_j$ where $r_{j+1}, r_j \epsilon \mathcal{R}$.

The final decision for each sample is determined using the *Group of Experts Rule*, where DNN nodes are grouped based on the correlation of their decisions, with the most qualified group making the decision. In cases where this method proves ineffective, the *Most Honest Rule* is invoked. Here, the DNN node with the highest accumulated rewards is entrusted with making the final decision.

Under the Group of Experts Rule, each DNN node in the set $\mathcal{A}$ operates sequentially, commencing with the node possessing the highest reward score and proceeding in descending order. The initial node serves as the primary node and generates a prediction $\hat{y}_p$ using Eq. (5). If any subsequent node agrees with this prediction, they form a group and collectively advocate for $\hat{y}_p$, combining their rewards. The node that has aligned with the primary node is then excluded from further participation. This process iterates until all nodes in $\mathcal{A}$ have formed groups, resulting in sets of agreed-upon DNN inference predictions and cumulative rewards for the specific conflict sample. The final decision for the i^{th} sample occurs when consensus for the next sample $i+1$ is reached. At this point, let g_i be one of the formed groups and r_{g_i} the combined rewards for that group, if the primary DNN node for the next sample $a_p \in g_i$ and:

$$\frac{r_{g_i}}{\sum_{i=1}^{N} r_i} \geq 0.51 \tag{9}$$

then the agent a_p, is responsible to decide for the i^{th} conflict sample as $\hat{y}_p = \text{argmax}\left(\mathbf{f_p}(\mathbf{x_i}; \boldsymbol{\theta_p})\right)$.

Alternatively, under the Most Honest Rule, if the primary's group fails to meet the quality threshold, the decision is made by the DNN node with the highest reward score (e.g., $a_p = \text{argmax}(r_j)$), and the decision he produce is the one applied to the i^{th} conflict sample.

4. DISCUSSION AND EXPERIMENTS

This section contains a discussion and in-depth examination of the suggested methods' experimental results. In our systematic design, we envision a decentralized network consisting of numerous autonomous DNN nodes, each representing either Unmanned Autonomous Vehicles (UAVs) or ground stations equipped with dedicated CNN models for inference tasks. Within this network, some nodes are equipped with robust, well-established pre-trained models capable of delivering state-of-the-art inference results. However, alongside these proficient nodes, there exist poorly performing nodes. These nodes may produce subpar results due to malicious intent or inadequate training.

We conduct experiments on Blaze dataset consisting of 1576 testing images of labeled areas before and during a fire event, uniformly distributed in 4 classes (e.g., burnt, fire, half burnt and non-burnt). We utilize a total of 7 pre-trained models, namely four versions of the EfficientNet [21] architecture (e.g., BO, B1, B2, B3), Inception v3 [22], ResNet-50 [23], and ResNet-101 [24]. All models were pre-trained on ImageNet [25] and further fine-tuned for an additional 200 epochs on the Blaze dataset. We evaluate the performance of each individual DNN node and discuss their results with the IMSP method's numerical outcomes. Additionally, we apply the QoI consensus protocol to both the base DNN nodes and their enhanced versions.

Fig. 1. Forest fire example from Evros, Greece. The baseline DNN nodes' inference outputs are: $\{fire, fire, fire, fire, burnt, burnt, non - burnt\}$. Under the average condition of the IMSP method, the final decisions for each agent are: $\{fire, fire, fire, fire, fire, fire, burnt\}$. The overall system's decision was determined to be $fire$.

Table 1. Results of IMSP method on Blaze Dataset.

Experiment	Dataset	Accuracy (%)						
		A1	A2	A3	A4	A5	A6	A7
Base Nodes	Blaze	82.32	**85.32**	84.49	83.85	61.90	76.90	67.20
DA -MV		49.84	44.22	58.77	45.63	74.66	**83.41**	**81.49**
DA - Avg		**83.54**	85.26	**85.07**	**85.07**	**78.69**	82.03	79.90

To elucidate certain notations herein, we designate the DNN nodes as *Base Nodes* and showcase their respective accuracy metrics across the Blaze dataset. Furthermore, the term DA - Avg denotes the Decentralized DNN nodes subsequent to the implementation of the proposed averaging criterion within the IMSP inference method, delineated in Subsection 3.1. Similarly, DA - MV represents the decentralized DNN nodes following the integration of the majority voting mechanism. To ensure manageability and discernibility of individual node performance, an effort is made to maintain a limited count of nodes. In the base nodes scenario, accuracy scores range from 61.90% to 85.32%, indicating varying performance levels among the nodes. As depicted in Table 1, the DA-MV condition struggles to effectively maintain and en-

hance the baseline accuracy of efficient models resulting in a drop in their accuracy from $82.32\% - 85.32\%$ to $44.22\% - 58.77\%$. This issue arises because underperforming nodes have the potential to mislead the process, as they tend to exhibit high confidence in their incorrect predictions. However, in DA-Avg, this is not a concern. By focusing directly on each class probability score, it can effectively detect and exclude poorly performing nodes. With accuracy scores ranging from 78.69% to 85.54%, DA-Avg method outperforms both the base nodes and the DA - MV condition in almost all of the cases, indicating that DA-Avg approach effectively mitigates the negative impact of poor behaving node's accuracy and may offer a more robust solution.

Table 2. Comparison of Aggregation Methods in Blaze Dataset

Experiments	Dataset	Centralized Voting Rules		QoI Consensus Protocol
		Weight Average	Majority Voting	
Base Nodes		84.65	83.66	**85.64**
DA - Avg	Blaze	-	-	85.51

Moving forward, a thorough comparison between established centralized decision-making aggregation methods, such as majority voting and weighted average, and the outcomes derived from the Quality of Inference (QoI) consensus protocol, is depicted in Table 2. As evidenced, the QoI not only yields comparable results but also effectively outperforms both majority and weighted average rules by 0.99%. This performance consistency persists post-application of the IMSP method, which initially enhances the baseline nodes' performance by employing the average condition rule. The DA-MV results are omitted due to their lower performance compared to DA-Avg. Through the consensus process among already enhanced nodes, a final decision is reached within a fully decentralized structure. This architecture diverges from a straightforward decision-making process seen in centralized architectures, favoring a collaborative approach that enhances the final process through the QoI protocol. This protocol effectively mitigates occasional ties in majority voting through a reward system and conflict sample management, resolving ties based on the integrity of nodes during the consensus process rather than a predetermined rule.

5. CONCLUSION

In this study, a decentralized deep neural network (DNN) inference framework tailored for fire classification tasks, with the aim of enhancing the decision-making process within a multi-agent system for effective natural disaster management is introduced. An individualized model selection process facilitates the exchange and aggregation of information among DNN nodes to enhance their individual performance. To ensure consistent and reliable inference, a Quality of Inference (QoI) protocol is proposed enabling DNN nodes to maintain a comprehensive record of DNN inference history to inform system-wide knowledge and decision-making. Furthermore, the adoption of a fault-tolerant inference architecture enabled the effective management of misbehaving DNN nodes, reducing their ability to influence the decisions of well-behaved DNN nodes. Through the empirical classification task on the Blaze dataset, the proposed methodologies had proven that they could be established as a secure decentralized inference framework, had shown resilience to malicious attacks, and had been capable of delivering performance on par with centralized decision techniques. In the context of future work, there is potential to integrate and refine the QoI protocol to function as a consensus mechanism within an AI-Blockchain framework designed for diverse deep learning tasks. Furthermore, exploring the integration of these methods into other computer vision applications such as segmentation and object detection represents a promising avenue for investigation.

6. ACKNOWLEDGMENT

This work has received funding from the European Commission—European Union under HORIZON EUROPE (HORIZON Research and Innovation Actions) under grant agreement No 101093003 (TEMA HORIZON-CL4-2022-DATA-01-01). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union—European Commission. Neither the European Commission nor the European Union can be held responsible for them.

7. REFERENCES

[1] WJ Wouter Botzen, Olivier Deschenes, and Mark Sanders, "The economic impacts of natural disasters: A review of models and empirical studies," *Review of Environmental Economics and Policy*, 2019.

[2] Elton Domnori, Giacomo Cabri, and Letizia Leonardi, "Multi-agent approach for disaster management," in *2011 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 2011, pp. 311–316.

[3] NM Adams, M Field, E Gelenbe, DJ Hand, NR Jennings, DS Leslie, D Nicholson, S Ramchurn, SJ Roberts, and A Rogers, "The aladdin project: intelligent agents for disaster management," in *IARP/EURON Workshop on Robotics for Risky Interventions and Environmental Surveillance (RISE)*, 2008.

[4] Zhi Zhou, Xu Chen, En Li, Liekang Zeng, Ke Luo, and Junshan Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[5] Song Han, Huizi Mao, and William J Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[6] Ben Taylor, Vicent Sanz Marco, Willy Wolff, Yehia Elkhatib, and Zheng Wang, "Adaptive deep learning model selection on embedded systems," *ACM SIGPLAN Notices*, vol. 53, no. 6, pp. 31–43, 2018.

[7] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang, "A survey of model compression and acceleration for deep neural networks," *arXiv preprint arXiv:1710.09282*, 2017.

[8] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[9] Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, and Lingjia Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," *ACM SIGARCH Computer Architecture News*, vol. 45, no. 1, pp. 615–629, 2017.

[10] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung, "Branchynet: Fast inference via early exiting from deep neural networks," in *2016 23rd international conference on pattern recognition (ICPR)*. IEEE, 2016, pp. 2464–2469.

[11] Shigeng Zhang, Yinggang Li, Xuan Liu, Song Guo, Weiping Wang, Jianxin Wang, Bo Ding, and Di Wu, "Towards real-time cooperative deep inference over the cloud and edge end devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 2, pp. 1–24, 2020.

[12] Tinghao Zhang, Zhijun Li, Yongrui Chen, Kwok-Yan Lam, and Jun Zhao, "Edge-cloud cooperation for dnn inference via reinforcement learning and supervised learning," in *2022 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*. IEEE, 2022, pp. 77–84.

[13] May Malka, Erez Farhan, Hai Morgenstern, and Nir Shlezinger, "Decentralized low-latency collaborative inference via ensembles on the edge," *arXiv preprint arXiv:2206.03165*, 2022.

[14] A Besir Kurtulmus and Kenny Daniel, "Trustless machine learning contracts; evaluating and exchanging machine learning models on the ethereum blockchain," *arXiv preprint arXiv:1802.10185*, 2018.

[15] Leslie Lamport, Robert Shostak, and Marshall Pease, "The byzantine generals problem," in *Concurrency: the works of leslie lamport*, pp. 203–226. 2019.

[16] Miguel Castro and Barbara Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems (TOCS)*, vol. 20, no. 4, pp. 398–461, 2002.

[17] Alysson Bessani, João Sousa, and Eduardo EP Alchieri, "State machine replication for the masses with bft-smart," in *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, 2014, pp. 355–362.

[18] Yuqi Fan, Huanyu Wu, and Hye-Young Paik, "Dr-bft: A consensus algorithm for blockchain-based multi-layer data integrity framework in dynamic edge computing system," *Future Generation Computer Systems*, vol. 124, pp. 33–48, 2021.

[19] Diego Ongaro and John Ousterhout, "In search of an understandable consensus algorithm," in *2014 USENIX annual technical conference (USENIX ATC 14)*, 2014, pp. 305–319.

[20] Christian Cachin et al., "Architecture of the hyperledger blockchain fabric," in *Workshop on distributed cryptocurrencies and consensus ledgers*. Chicago, IL, 2016, vol. 310, pp. 1–4.

[21] Mingxing Tan and Quoc Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[22] Cheng Wang, Delei Chen, Lin Hao, Xuebo Liu, Yu Zeng, Jianwei Chen, and Guokai Zhang, "Pulmonary image classification based on inception-v3 transfer learning model," *IEEE Access*, vol. 7, pp. 146533–146541, 2019.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[24] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[25] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.